UNIVERSITÉ DE LORRAINE

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Modèles de langue exploitant la similarité structurelle entre séquences pour la reconnaissance de la parole

THÈSE

présentée et soutenue publiquement le 12 décembre 2012

pour l'obtention du

Doctorat de l'université de Lorraine

spécialité informatique

par

Christian Gillot

Composition du jury

Rapporteurs :	Laurent Besacier - Professeur - Université J. Fourier, LIG - Grenoble
	Patrice Bellot - Professeur - Université d'Aix-Marseille, LSIS - Marseille
Examinateurs :	Guillaume Gravier - Chargé de recherche CNRS HDR, IRISA - Rennes
	Guy Perrier - Professeur - Université de Lorraine, LORIA - Nancy
	Jean-Paul Haton - Professeur - Université de Lorraine, LORIA - Nancy
Directeur :	Christophe Cerisara - Chargé de recherche CNRS HDR, LORIA - Nancy

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Mis en page avec la classe thloria.

Remerciements

Remerciements tout d'abord à ma famille, mes amis, tous ceux qui ont été proches, une thèse c'est fait de hauts et de bas et ils ont toujours été là. Remerciements à mon directeur de thèse Christophe, toujours de bonne volonté et disponible. Remerciements à Jean-Paul Haton qui m'a encouragé à aller là où je le voulais dans mes travaux de recherche. Remerciements à tous les membres de l'équipe Parole, les précaires comme les permanents, avec qui j'ai eu de nombreuses discussions enrichissantes, d'un point de vue personnel comme professionnel.

À mon grand-père Maurice, ouvrier à la Sollac, parti un peu trop vite pour que nous puissions nous connaître.

Table des matières

Introduction

Reconnaissance de la parole

La parole est le langage articulé de l'être humain qui rend possible la communication sonore. La « reconnaissance de la parole » est le processus de transcription d'un enregistrement sonore, c'est-à-dire qu'à l'issue de ce processus on obtient la suite de mots prononcée. C'est une des choses que l'on fait tout naturellement lorsqu'on écoute quelqu'un parler. La « synthèse de la parole » est le processus inverse, qui étant donné une suite de mots produit un enregistrement sonore lui correspondant. C'est ce que l'on fait lorsqu'on parle à quelqu'un.

1 Motivation

La reconnaissance de la parole est une faculté de l'homme qu'il utilise le plus souvent sans même y penser. On y a recours très couramment puisque l'homme est un être social.

De nombreuses personnes sont employées à des tâches de transcription de parole. Par exemple, en 2010, la direction générale de la traduction chargée de traduire des documents pour la Commission Européenne compte 2 500 employés permanents. Le coût de la réalisation de ces tâches est immense et constitue un frein à l'accessibilité de l'information aux personnes qui ne peuvent pas les réaliser elles-mêmes.

Par exemple, d'après le ministère de la santé 6,6 % de la population française en 2005 soit plus de 4 millions de personnes ont des problèmes auditifs, dont 500 000 des déficiences auditives sévères ou profonde.

Pour rendre accessible la télévision à ces citoyens, la loi française numéro 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées impose aux grandes chaînes de télévision de sous-titrer l'ensemble de leurs programmes. Mais les chaînes dont l'audience est inférieure à 2,5 % en sont exemptées en raison du coût budgétaire trop important pour ces petites structures.

Ne serait-ce que par cet exemple, on voit l'enjeu que constitue l'amélioration de l'efficacité économique de ces tâches par l'informatisation et les utilisations nouvelles qui deviendraient alors possibles.

La science est une autre motivation tout aussi importante. Étant donné qu'à l'heure actuelle on ne comprend que partiellement les mécanismes cognitifs de la parole, on ne sait pas les modéliser fidèlement. La communauté scientifique travaille donc à proposer

des modèles efficaces modélisant les différents processus de la parole.

Il serait également fort intéressant en soi de pouvoir interagir avec les systèmes d'informations avec la parole. Non seulement pour faciliter l'utilisation de l'informatique mais également la rendre accessible aux personnes qui ne peuvent le faire, par exemple dans les cas d'illettrisme ou d'handicaps.

La « reconnaissance automatique de la parole » vise donc à créer un programme informatique pour réaliser de façon automatique la transcription d'enregistrements sonores.

2 Modèle de langue

Dans la reconnaissance automatique de la parole, on a coutume de décomposer le problème avec deux modèles indépendants : le modèle acoustique et le modèle de langue. Le premier vise à modéliser les différents sons de la langue et le second les suites de mots propres à cette langue.

Ainsi les phrases « On est souvent injuste en s'abstenant d'agir et non seulement en agissant. » et « On n'est sous vent un juste en s'habillant d'haïr et non seul ment en agissant. » sont acoustiquement très proches et toutes deux constituées de mots de la langue française. Mais si vous entendez quelqu'un prononcer la citation de Marc-Aurèle, il serait fort surprenant que vous compreniez la seconde phrase qui semble toute droite issue d'un cadavre exquis. En effet, vous avez en tant qu'être humain une exceptionnelle capacité de compréhension de la langue, et ce même lors de conditions adverses : bruit ambiant, plusieurs personnes parlant en même temps, variantes de la langue, variantes d'accent, fautes de grammaire, fautes de vocabulaire, répétitions de mots, hésitations, etc.

Un modèle de langue n'est pas seulement utile pour la reconnaissance de la parole, en effet il sert à de nombreuses autres applications. On peut citer la reconnaissance optique de caractères qui est le processus d'extraction du texte contenu dans des images, notamment utilisé pour numériser des livres, le traitement automatique du courrier ou des chèques, etc. Un modèle de langue est également utilisé pour la recherche d'information, par exemple dans les moteurs de recherche d'Internet. Citons encore les systèmes de recommandation qu'on utilise par exemple pour déterminer dans un site Web les autres pages qui sont suceptibles de vous intéresser.

Cette thèse a pour objectif l'amélioration du modèle de langue dans son contexte d'utilisation de la reconnaissance de la parole en français. Toutefois, on peut tout à fait appliquer ces travaux à d'autres langues et à d'autres applications. Nous commencerons par passer en revue les travaux existants et liés à la problématique du modèle de langue dans la partie I. Puis nous exposerons dans la partie II les différentes hypothèses explorées au cours de la thèse pour mieux modéliser la langue française.

Les principales hypothèses de travail ainsi décrites, au nombre de trois, dérivent du même principe sous-jacent qui a guidé la réflexion menée tout au long de cette thèse. Ce principe vise à calculer la probabilité du modèle de langage de manière plus précise, c'est-à-dire en prenant en compte une information contextuelle plus riche que celle classiquement utilisée, afin d'identifier quelles parties du corpus d'apprentisage représentent le mieux les énoncés spécifiques du test que nous souhaitons transcrire. Dans les deux premières hypothèses, le contexte est simplement modélisé par l'historique des mots précédant le mot cible, mais les approches proposées diffèrent de l'état de l'art par le fait qu'elles manipulent un historique de longueur bien plus grande qu'il n'est d'usage, et une nouvelle mesure de similarité pour prendre en compte cet historique est utilisée. Dans la troisième hypothèse de travail, le contexte est étendu à la notion de structures syntaxiques spécifiques, identifiées comme telles par des connaissances expertes. L'objectif ainsi défini s'appuie donc sur des notions développées dans le paradigme des modèles à base de mémoire, encore connus sous le nom d'apprentissage à base d'exemples (*Instance-based or memory-based learning*). Mais, à la différence d'autres approches développées relativement récemment, comme dans le projet *Sound2Sense* par exemple, un aspect important de ce travail consiste à intégrer de telles connaissances et données spécifiques au sein même des modèles probabilistes de langue, en modifiant le moins possible la délicate architecture d'un système de transcription automatique de la parole état de l'art. Ceci est une tâche d'autant plus difficile que les paradigmes impliqués, celui des calculs de probabilités d'une part, et des modèles à base d'exemple d'autre part, diffèrent fondamentalement l'un de l'autre. Un certain nombre de compromis et d'hypothèses de travail sont donc décrits dans la deuxième partie afin de permettre cette difficile mais potentiellement intéressante intégration.

Première partie

Modèle de langue : un état de l'art

1

Modèle de langue probabiliste

1.1 Introduction

Le but du modèle de langue est d'estimer la qualité linguistique relative des différentes hypothèses de mots du système qui lui sont proposées. Supposons que lors d'un cours un professeur de comptabilité prononce à voix haute les mots « vingt centimes » et que l'on dispose de l'enregistrement, le modèle acoutisque de reconnaissance automatique proposera pour ces mots « vingt centimes » de très nombreuses hypothèses, des centaines, voire des milliers dont par exemple :

- Vincent Times
- vingt cent temps
- Vincent Timbre
- vingt centimes
- etc

Alors pourquoi choisir l'hypothèse « vingt centimes » plutôt qu'une autre hypothèse ? On peut utiliser une base de connaissance contenant beaucoup de textes écrits en français, la consulter pour calculer la fréquence de chaque hypothèse et enfin choisir l'hypothèse la plus fréquente. Pour s'en convaincre, on peut chercher le nombre de pages Web contenant ces expressions sur un moteur de recherche. Voici les résultats du 17 mai 2011 sur Google :

Suite de mots	Nombre de pages Web
Vincent Times	7 470
vingt cent temps	0
Vincent Timbre	378
vingt centimes	36 400

D'après ces résultats, la meilleure hypothèse a tout l'air d'être « vingt centimes ». Ce petit travail que nous avons fait de manière intuitive est l'idée centrale des modèles de langue probabilistes que nous allons maintenant formaliser.

1.2 Formalisation

Soit une langue et son vocabulaire noté $\mathcal{V}$, c'est-à-dire l'ensemble des mots utilisés dans cette langue. Soit une suite de mots quelconque w de cette langue composée des mots $w_1 w_2 ... w_n$, $n \in \mathbb{N}^*$. On note w_i^j la sous-séquence de w composée des mots $w_i w_{i+1} ... w_j$. Un modèle de langue probabiliste donne une estimation de la probabilité $P(w)$ de la suite de mots w dans la langue modélisée. On décompose cette probabilité de la façon suivante :

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\dots P(w_n|w_1^{n-1}) \tag{1.1}$$

Le problème du modèle de langue probabiliste revient donc à estimer la probabilité conditionnelle d'un mot connaissant son historique, c'est-à-dire les mots le précédant. Par exemple, la probabilité de la phrase « l'erreur est humaine » se calcule de la façon suivante :

$$P(\text{l'erreur est humaine}) = P(\text{l'})P(\text{erreur}|\text{l'})P(\text{est}|\text{l'erreur})P(\text{humaine}|\text{l'erreur est}) \tag{1.2}$$

Pour estimer les probabilités conditionnelles de l'équation 1.1, conformément à l'intuition de départ, on utilise un corpus, c'est-à-dire une grande base de connaissance constituée de textes de la langue modélisée. On le décompose en trois parties :

1. le corpus d'apprentissage $\mathcal{C}$, le plus volumineux qui constitue la mémoire du modèle.
2. le corpus de développement $\mathcal{D}$, plus petit qui permet de régler les paramètres du modèle.
3. le corpus d'évaluation $\mathcal{T}$, lui aussi plus petit qui permet d'évaluer la performance du modèle.

À l'heure actuelle, un corpus d'apprentissage est de l'ordre du milliard de mots. À titre de comparaison, environ 10 000 mots sont prononcés en une heure donc le corpus d'apprentissage est l'équivalent de plus de 11 ans de parole. Quelques dizaines de milliers de mots sont suffisants pour les corpus de développement et d'évaluation.

1.3 Évaluation

Avant de discuter des différents modèles de langue existants, on se pose la question suivante : comment évaluer la qualité d'un modèle de langue ? Dans de nombreux domaines, une évaluation objective peut être difficile ou coûteuse à mettre en œuvre. Par exemple, dans le domaine de la synthèse vocale, on a souvent recours à un jury humain qui évalue de façon subjective la qualité des modèles, ce qui est très coûteux. Fort heureusement pour les modèles de langues, les mesures standards d'évaluation sont plus simples et objectives.

1.3.1 Évaluation de la reconnaissance de la parole

Pour évaluer la qualité d'un système de reconnaissance de la parole, la mesure la plus courante est le taux d'erreur en mots (TEM, en anglais *word error rate* abrégé en WER). Le processus de mesure commence par une reconnaissance automatique avec le modèle de langue candidat sur le corpus d'évaluation audio. Puis on compare l'hypothèse de reconnaissance avec la transcription humaine du corpus d'évaluation en cherchant la transformation la moins coûteuse possible de l'hypothèse d'évaluation en la transcription de référence à l'aide de quatre opérations de transformation :

1. l'appariement (A), lorsque les mots de l'hypothèse et la référence sont identiques
2. la modification (M), lorsqu'on change un mot de l'hypothèse par un mot de la référence
3. la suppression (S), lorsqu'on supprime un mot de l'hypothèse
4. l'insertion (I), lorsqu'on insère un mot de la référence dans l'hypothèse

L'opération d'appariement a un coût nul tandis que les autres opérations ont un coût de 1. Lorsqu'on a trouvé la meilleure transformation possible à l'aide d'un algorithme de programmation dynamique, on fait le total de chaque opération et le taux d'erreur en mot est donné par :

$$TEM = \frac{M + S + I}{A + M + I} = \frac{M + S + I}{H} \tag{1.3}$$

où H est le nombre de mots de la transcription de référence.

Par exemple, le système de référence de l'équipe PAROLE a reconnu le morceau de phrase suivant enregistré sur France Inter : « bien d'autres moments très concrètement ». La transcription de référence de ce morceau de phrase est : « hé bien nous demandons très concrètement ». La transformation de coût minimal, qu'on appelle aussi alignement, est donnée dans la table suivante :

Référence	hé	bien	**	nous	demandons	très	concrètement
Hypothèse	**	bien	d'	autres	moments	très	concrètement
Opération	S	A	I	M	M	A	A

Sur cet exemple, on a 1 opération de suppression, 1 opération d'insertion, 2 opérations de modifications pour 6 mots dans la transcription de référence ce qui donne un taux d'erreur en mots de $\frac{1+1+2}{6} = 0.66$, c'est-à-dire un taux de 66 %.

De façon générale, les performances du système de reconnaissance dépendent de beaucoup de paramètres, bien sûr du modèle de langue et du modèle acoustique, mais également des conditions d'enregistrement : bruit ambiant, multiples locuteurs, qualité du

microphone, accent du locuteur, proportions de mots inconnus, etc. C'est pourquoi on évalue la qualité d'un modèle de langue en faisant une comparaison du taux d'erreur en mots du système de reconnaissance de la parole que l'on obtient avec le modèle de langue étudié et un modèle de langue état de l'art.

1.3.2 Évaluation sur du texte

Comme nous l'avions vu en introduction, la reconnaissance de la parole n'est pas la seule application du modèle de langue. De plus le processus de reconnaissance de la parole est très gourmand en ressources informatiques et peut prendre beaucoup de temps, ce qui ralentit le processus de recherche. C'est pourquoi on utilise bien souvent une autre mesure d'évaluation qui se calcule sur du texte et beaucoup plus rapidement : la perplexité.

La perplexité se calcule donc sur un texte d'évaluation $\mathcal{T}$, elle est directement relié à la probabilité que le modèle de langue donne à ce texte. Si le texte $\mathcal{T}$ est composé des mots $w_1 w_2 \dots w_n$ alors la probabilité du texte $\mathcal{T}$ par le modèle de langue P est donné par :

$$P(T) = \prod_{i=1}^{n} P(w_i | w_1^{i-1}) \tag{1.4}$$

L'entropie croisée $H_P(\mathcal{T})$ du modèle P sur le texte $\mathcal{T}$ est définie ainsi :

$$H_P(\mathcal{T}) = -\frac{1}{n} log_2 P(T) \tag{1.5}$$

L'entropie croisée mesure le nombre de bits moyen qu'il faut pour encoder les données d'évaluation $\mathcal{T}$ avec un algorithme de compression associé au modèle de langue P. Par abus de langage, lorsqu'on parlera d'entropie, on se référera à l'entropie croisée de P sur le corpus d'évolution $\mathcal{T}$.

La perplexité $PPL_P(\mathcal{T})$ d'un modèle P est la réciproque de la moyenne géométrique de la probabilité assignée par le modèle de langue à chacun des mots du corpus d'évaluation $\mathcal{T}$ et qu'on peut écrire en fonction de l'entropie croisée :

$$\underset{P}{PPL}(\mathcal{T}) = 2^{H_P(\mathcal{T})} \tag{1.6}$$

Plus l'entropie croisée est petite et donc plus la perplexité est petite, meilleur est le modèle de langue.

1.4 Méthodes statistiques

Un modèle statistique va donc utiliser des statistiques issues du corpus d'apprentissage $\mathcal{C}$ pour estimer la distribution de probabilité $P(w_i | w_1^{i-1})$ conditionnelle d'apparition d'un mot w_i étant donné son historique w_1^{i-1}.

1.4.1 Modèle de langue n-gramme

Présentons maintenant le modèle de langue de l'état de l'art le plus couramment utilisé, le modèle n-gramme. Pour un historique w_1^{i-1}, l'idée est de calculer la distribution de probabilité pour chaque mot du vocabulaire $w_i \in \mathcal{V}$ en se basant sur la statistique du nombre d'occurrences de l'historique suivi de ce mot, c'est-à-dire du nombre d'occurrences de la séquence $w_1^{i-1}w_i$, puis d'attribuer une masse de probabilité pour chaque mot en fonction de cette statistique. On note $C(w_i^j)$ le nombre d'occurrences de la suite de mots w_i^j dans $\mathcal{C}$. La probabilité conditionnelle d'un mot étant donné son historique est donnée par le rapport d'occurrences suivant :

$$P(w_i|w_1^{i-1}) = \frac{C(w_1^i)}{\sum_{x\in\mathcal{V}} C(w_1^{i-1}x)} \tag{1.7}$$

Par exemple si la probabilité à estimer est $P(\text{humaine}|\text{l'erreur est})$, on a donc :

$$P(\text{humaine}|\text{l'erreur est}) = \frac{C(\text{l'erreur est humaine})}{\sum_{w_i\in\mathcal{V}} C(\text{l'erreur est } w_i)} \tag{1.8}$$

Supposons maintenant que le système ait besoin d'estimer la probabilité $P(\text{pernicieuse}|\text{l'erreur est})$ mais que la suite de mots « l'erreur est pernicieuse » n'apparaisse pas dans le corpus. Par exemple, au 17 mai 2011, une requête de recherche sur Internet sur cette chaîne ne renvoit aucune page. On suppose donc que $C(\text{l'erreur est pernicieuse}) = 0$, d'après la formule 1.7, on a :

$$P(\text{pernicieuse}|\text{l'erreur est}) = \frac{0}{\sum_{w_i\in\mathcal{V}} C(\text{l'erreur est } w_i)} = 0 \tag{1.9}$$

Ce modèle donne une probabilité nulle à une suite de mots et la considère donc impossible. Pourtant celle-ci apparait tout à fait plausible en français, comment faire pour contourner ce problème ? D'autant plus que plus l'historique est long, moins on trouve d'exemples de mot suivant cet historique dans la base d'apprentissage...

En effet, si on a un vocabulaire de $Card(\mathcal{V})$ mots, il existe $Card(\mathcal{V})^n$ n-grammes possibles. Or si l'on a par exemple un vocabulaire de 63690 mots comme c'est le cas dans le système de reconnaissance automatique de la parole de l'équipe PAROLE, on a donc plus de $1,64.10^{11}$ milliards de 4-grammes possibles. Sachant que le corpus d'apprentissage contient moins d'un milliard de mots, on comprend bien qu'il n'y ait pas assez de données pour pouvoir estimer le modèle 4-gramme.

Une solution simple de résoudre ce problème pour le modèle n-grammes d'ordre n est d'ignorer les premiers mots de l'historique et donc de considérer la suite de mots comme un processus de Markov d'ordre $n-1$ et donc :

$$P(w_i|w_1^{i-1}) = P(w_i|w_{i-n+1}^{i-1}) = \frac{C(w_{i-n+1}^{i})}{\sum_{x \in \mathcal{V}} C(w_{i-n+1}^{i-1}x)} \tag{1.10}$$

À l'heure actuelle les modèles n-grammes utilisés sont d'ordre 4 ou 5. Or comme on l'a déjà constaté, même une suite de mots « l'erreur est pernicieuse » qui est de longueur 4 peut être mal estimée. Pour résoudre ce problème, on va combiner l'estimation de probabilité des modèles n-grammes d'ordre 1 à n au moyen d'une technique de lissage.

1.4.2 Techniques de lissage

La façon la plus simple de combiner plusieurs modèles probabilistes est d'utiliser une combinaison linéaire. Jelinek et Mercer [34] décrivent de façon générale cette classe de modèles interpolés.

Formellement, si on note $P^j(w_i|w_1^{i-1})$ l'estimation de probabilité donnée par le modèle n-gramme d'ordre j et λ_j le poids du modèle d'ordre j, avec $\sum_{j=1}^{n} \lambda_j = 1$, on a alors :

$$P(w_i|w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j P^j(w_i|w_1^{i-1}) \tag{1.11}$$

On apprend les coefficients λ_j par optimisation de la perplexité sur le corpus $\mathcal{D}$. Cela nous donne un premier modèle qui n'est pas le plus performant, mais qui règle bien le problème d'estimation des n-grammes inexistants dans le corpus d'apprentissage.

On peut écrire plus élégamment cette interpolation à la manière de Brown *et al.* [7] en définissant le modèle d'ordre n de façon récursive :

$$P^j_{interp}(w_i|w_1^{i-1}) = \lambda_j P^j(w_i|w_1^{i-1}) + (1 - \lambda_j) P^{j-1}(w_i|w_1^{i-1}) \tag{1.12}$$

Nous allons maintenant passer en revue quelques unes des techniques de lissage les plus courantes de l'état de l'art. Les lecteurs intéressés par une présentation plus exhaustive des techniques de lissages peuvent se référer à l'état de l'art [18] de Chen et Goodman.

Lissage de Jelinek-Mercer

Une première amélioration proposée par Jelinek et Mercer [34] consiste à faire varier les coefficients d'interpolation λ_j en fonction du contexte actuel. En effet, mieux un modèle est estimé pour un contexte donné, moins on a besoin de recourir aux modèles d'ordre inférieur, qui sont par nature moins précis.

Le facteur d'interpolation dépend donc du contexte actuel, ce qu'on note : $\lambda_j(w_{i-j+1}^{i})$. Il faut alors déterminer les valeurs optimales des λ_j qui donnent le meilleur modèle de

langage possible, c'est-à-dire qui minimise la perplexité sur le corpus de validation $\mathcal{T}$.

On peut apprendre ces valeurs par l'algorithme de Baum-Welch [5] en minimisant la perplexité du corpus de développement $\mathcal{D}$. Jelinek et Mercer décrivent la technique d'interpolation par suppression dans laquelle le corpus d'apprentissage est découpé en plusieurs parties. À chaque passe de l'algorithme, chaque partie sert soit à estimer la probabilité n-gramme soit à estimer les valeurs λ_j. On parle d'interpolation par suppression car les parties qui servent à estimer les λ_j sont « supprimées » du corpus d'apprentissage de la probabilité n-gramme. Les valeurs finales des λ_j sont les moyennes des valeurs obtenues à chaque passe. Bien que plus complexe à mettre en œuvre, cette technique est intéressante lorsqu'on a peu de données car elle permet de toutes les utiliser pour estimer la probabilité n-gramme.

Le nombre des λ_j à estimer est bien trop important pour qu'on puisse en réaliser une estimation fiable. C'est pourquoi Bahl, Jelinek et Mercer [4] proposent de regrouper les λ_j en fonction du compte de l'historique $C(w_{i-j+1}^{i-1})$. L'hypothèse est que plus un historique a d'occurences, mieux le modèle d'ordre j sera estimé. Malgré tout, cela fait encore trop de paramètres, c'est pourquoi on regroupe en classes (*bucket*) les comptes qui sont proches de façon à ce que chaque classe ait assez de données pour être correctement estimée. Dans sa thèse [17], Chen a montré qu'il est plus efficace de regrouper les λ_j par le nombre d'occurrences moyen des éléments de la distribution d'ordre j, c'est-à-dire $\frac{\sum_{w_i} C(w_{i-j+1}^{i})}{|w_i : C(w_{i-j+1}^{i}) > 0)|}$.

Quelque soit la manière de procéder, les paramètres à estimer restent très nombreux ce qui constitue une difficulté. C'est pourquoi les techniques de lissage plus robustes et les plus utilisées à l'heure actuelle ont pour point commun de n'avoir qu'un petit nombre de paramètres à estimer.

Estimation de Good-Turing

L'estimation de Good-Turing [30] est le point de départ des techniques de lissage les plus performantes. Elle stipule que pour tout n-gramme qui apparaît r fois dans le corpus d'apprentissage $\mathcal{C}$, on ne devrait prétendre l'avoir vu que r^* fois :

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{1.13}$$

où n_r est le nombre de n-grammes qui apparaissent exactement r fois dans $\mathcal{C}$. On normalise ensuite pour obtenir une probabilité avec ces comptes modifiés, ainsi pour un n-gramme w_{i-n+1}^{i} qui a pour compte r, on a :

$$P_{GT}(w_{i-n+1}^{i}) = \frac{r^*}{N} \tag{1.14}$$

où $N = \sum_{r=0}^{\infty} n_r r^*$. On peut aussi remarquer que :

$$N = \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1} = \sum_{r=1}^{\infty} r n_r \tag{1.15}$$

c'est-à-dire que N est égal au compte de départ (non modifié) de la distribution.

Du fait de la définition de r^* 1.13, on ne peut pas utiliser l'estimation de Good-Turing lorsque n_r est nul. Dans ce cas, il faut lisser les valeurs n_r de façon à ce qu'elles soient toutes strictement positives. Gale et Sampson [26] proposent un algorithme simple et efficace pour ce faire.

On ne peut pas utiliser l'estimation de Good-Turing par elle-même car elle ne permet pas la combinaison d'un modèle n-gramme avec ses modèles d'ordre inférieur mais elle est à la base des techniques de Katz, de décompte absolu et de Kneser-Ney que nous allons maintenant décrire.

Lissage de Katz

L'idée du lissage de Katz [37] est d'utiliser l'estimation de Good-Turing pour les n-grammes de compte non nul ce qui donne une masse de probabilité pour les n-grammes de compte nul qui est allouée en suivant le modèle d'ordre inférieur. Le compte modifié de Katz d'ordre n du n-gramme w_{i-n+1}^i est défini comme suit :

$$C_{katz}(w_{i-n+1}^i) = \begin{cases} d_r r & \text{si } r > 0 \\ \alpha^n(w_{i-n+1}^{i-1}) P_{katz}^{n-1}(w_{i-n+2}^i) & \text{si } r = 0 \end{cases} \tag{1.16}$$

Tous les n-grammes dont le nombre d'occurrences est supérieure à zéro sont décomptés par le facteur de décompte d_r, qui vaut $\frac{r^*}{r}$, c'est-à-dire le décompte donné par l'estimation de Good-Turing. Les comptes déduits des n-grammes de compte non nul sont redistribués aux n-grammes de compte nul suivant la distribution de probabilité d'ordre inférieur. La valeur $\alpha^n(w_{i-n+1}^{i-1})$ est choisie de façon à laisser le nombre total de comptes de la distribution $\sum_{w_i} C_{katz}(w_{i-n+1}^i)$ inchangé, c'est-à-dire $\sum_{w_i} C_{katz}(w_{i-n+1}^i) = \sum_{w_i} C(w_{i-n+1}^i)$. La valeur correcte du paramètre $\alpha^n(w_{i-n+1}^{i-1})$ est donnée par :

$$\begin{aligned} \alpha^n(w_{i-n+1}^{i-1}) &= \frac{1 - \sum_{w_i : C(w_{i-n+1}^i) > 0} P_{katz}^n(w_i | w_{i-n+1}^{i-1})}{\sum_{w_i : C(w_{i-n+1}^i) = 0} P_{katz}^{n-1}(w_i | w_{i-n+2}^{i-1})} \\ &= \frac{1 - \sum_{w_i : C(w_{i-n+1}^i) > 0} P_{katz}^n(w_i | w_{i-n+1}^{i-1})}{1 - \sum_{w_i : C(w_{i-n+1}^i) > 0} P_{katz}^{n-1}(w_i | w_{i-n+2}^{i-1})} \end{aligned} \tag{1.17}$$

Puis on normalise pour calculer $P^n_{katz}(w_i|w^{i-1}_{i-n+1})$:

$$P^n_{katz}(w_i|w^{i-1}_{i-n+1}) = \frac{C_{katz}(w^i_{i-n+1})}{\sum_{x\in\mathcal{V}} C_{katz}(w^{i-1}_{i-n+1}x)} \tag{1.18}$$

La récursion se termine par le modèle unigramme estimé par maximum de vraisemblance.

Examinons maintenant la définition du facteur de décompte d_r. Lorsque le nombre d'occurrences est suffisamment grand, supérieur à k, on le considère fiable et on ne le décompte pas : dans ce cas $d_r = 1$. Katz conseille de fixer ce seuil à $k = 5$. Pour les nombres d'occurrences inférieurs $r \leq k$, le facteur de décompte est calculé à partir de l'estimation de Good-Turing par rapport à la distribution globale d'ordre n ; ainsi n_r de la formule 1.13 est le nombre total de n-grammes qui apparaissent exactement r fois dans le corpus d'apprentissage. Les d_r sont choisis comme étant proportionnels à l'estimation de Good-Turing et de façon à ce que le nombre global d'occurrences décomptés dans la distribution globale d'ordre n soit égal au nombre total d'occurrences qui devraient être assignées aux n-grammes d'occurrence nulle selon l'estimation de Good-Turing. Ces contraintes aboutissent aux équations suivantes :

$$1 - d_r = \mu(1 - \frac{r^*}{r}) \tag{1.19}$$

pour $r \in \{1, 2, \ldots, k\}$ avec une constante μ. Puisque l'estimation de Good-Turing prédit que le nombre total de comptes qui devraient être assignés aux n-grammes de compte nul est $n_0 0^* = n_0 \frac{n_1}{n_0} = n_1$, alors la deuxième contrainte correspond à l'équation :

$$\sum_{r=1}^{k} n_r(1 - d_r)r = n_1 \tag{1.20}$$

La solution unique de ces équations est donnée par :

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{1.21}$$

Nous avions indiqué dans la section 1.4.2 que l'estimation de Good-Turing devait elle-même être lissée pour gérer les cas où n_r est nul. Pour le lissage de Katz, puisqu'on n'a besoin que des n_r pour r petit, il y aura suffisamment d'exemples dans le corpus d'apprentissage pour correctement les estimer et il n'y a donc pas besoin de lissage des estimations de Good-Turing.

Le lissage de Katz est une technique de lissage dite « par repli » car elle ne fait appel au modèle d'ordre inférieur que si le modèle actuel n'a pas d'estimation.

Lissage par décompte absolu

Plutôt que de multiplier les rapport d'occurrences des n-grammes comme le fait l'estimation de Good-Turing, Ney, Essen et Kneser [47] proposent de soustraire un décompte fixe au nombre d'occurrences de chaque n-gramme présent dans le corpus d'entraînement. Church et Gale [19] ont montré empiriquement qu'en moyenne le décompte de Good-Turing (c'est-à-dire $r - r^*$) pour $r \geq 3$ est constant. Donc la technique du décompte absolu est très proche du décompte de Good-Turing.

Formellement, on définit le décompte absolu $0 \leq D \leq 1$ qu'on soustrait à chaque n-gramme dont le compte est non nul. La probabilité est définie récursivement par la formule :

$$P^n_{abs}(w_i|w^{i-1}_{i-n+1}) = \frac{\max(C(w^i_{i-n+1}) - D, 0)}{\sum_{x \in \mathcal{V}} C(w^{i-1}_{i-n+1}x)} + (1 - \lambda_n) P^{n-1}_{abs}(w_i|w^{i-1}_{i-n+2}) \tag{1.22}$$

La valeur du coefficient λ_n est définie de façon à ce que la distribution de probabilité somme bien à 1 et donc :

$$1 - \lambda_n = \frac{D}{\sum_{w_i} C(w^i_{i-n+1})} N_{1+}(w^{i-1}_{i-n+1}\cdot) \tag{1.23}$$

Où N_{1+} est le nombre de mots qui suivent un historique, c'est-à-dire :

$$N_{1+}(w^{i-1}_{i-n+1}\cdot) = |\{w_i : C(w_{i-n+1}w_i) \geq 1\}| \tag{1.24}$$

Dans leur article [47], Ney, Essen et Kneser suggèrent d'apprendre la valeur de D par estimation par suppression sur le corpus d'apprentissage $\mathcal{C}$. Ils parviennent à l'estimation suivante :

$$D = \frac{n_1}{n_1 + 2n_2} \tag{1.25}$$

où n_1 et n_2 sont le nombre total de n-grammes avec respectivement un compte de un et deux dans $\mathcal{C}$.

Même si originellement, Ney, Essen et Kneser ont introduit cette technique de lissage par repli, la version présentée ici est une technique de lissage par interpolation. En effet, même pour les mots pour lesquels le modèle de plus haut ordre a une estimation de probabilité non nulle, on interpole cette estimation avec l'estimation des modèles d'ordre inférieur. Chen et Goodman [18] ont montré sur des évaluations comparant différentes techniques de lissage que pour une même technique, la technique par interpolation est plus performante que la technique de lissage par repli.

Lissage de Kneser-Ney

Kneser et Ney [39] ont proposé une variante du décompte absolu dans laquelle les modèles d'ordre inférieur sont modifiés pour apporter une information complémentaire au modèle d'ordre le plus élevé.

En effet dans tous les algorithmes décrits jusque ici les modèles d'ordre inférieur utilisent la même information, à savoir l'estimation par maximum de vraisemblance du modèle n-gramme. Mais étant donné que les modèles d'ordre inférieur sont surtout importants dans l'interpolation lorsque le modèle d'ordre supérieur a un compte faible ou nul, ils devraient par conséquent être optimisés pour ces situations. C'est ce que fait la technique de lissage de Kneser et Ney, en modifiant les modèles d'ordre inférieur.

Supposons que l'on crée un modèle bigramme avec un corpus d'apprentissage où il existe un mot très courant, disons « Mézières », mais qui n'apparait qu'après un seul mot disons « Charleville ». Puisque C(Mézières) est grand, la probabilité unigramme P(Mézières) est également grande et donc un modèle de langue lissé par décompte absolu donnera une probabilité relativement grande au mot « Mézières » pour des bigrammes dont l'historique est inconnu à l'apprentissage. Mais intuitivement cette probabilité ne devrait pas être grande puisqu'il n'existe qu'un seul historique qui est suivi par le mot « Mézières ». On devrait donc assigner une probabilité faible au mot « Mézières » puisque ce mot n'apparait dans le corpus que précédé du mot « Charleville », ce qui est déjà bien modélisé par le modèle bigramme.

L'intuition dans ce cas de figure d'un modèle bigramme est d'attribuer la probabilité unigramme non pas proportionnellement au nombre d'occurrences du mot mais au contraire au nombre de mots différents qu'il suit.

De façon générale, la probabilité est définie de la façon suivante :

$$P^n_{abs}(w_i|w^{i-1}_{i-n+1}) = \frac{\max(C(w^i_{i-n+1} - D), 0)}{\sum_x C(w^{i-1}_{i-n+1}x)} + (1 - \lambda_n)P^{n-1}_{abs}(w_i|w^{i-1}_{i-n+1}) \tag{1.26}$$

C'est-à-dire de la même manière que pour la technique du décompte absolu. Mais les comptes des distributions d'ordre inférieur sont modifiés de la façon suivante :

$$C(w^i_{i-n+2}) = N_{1+}(\cdot w^i_{i-n+2}) \tag{1.27}$$

De toutes les méthodes présentées, c'est la technique de lissage de Kneser-Ney qui est la plus performante aussi bien en terme de perplexité sur du texte qu'en taux d'erreur en mots en reconnaissance de la parole comme démontré par de nombreuses expériences dans [18]. Chen et Goodman y présentent également une variation qui consiste à utiliser

non pas un facteur mais trois facteurs de décomptes D_1, D_2 et D_{3+} qui s'appliquent respectivement aux n-grammes qui apparaissent une fois, deux fois et plus de trois fois dans le corpus $\mathcal{C}$. Cette variation qu'ils nomment technique de Kneser-Ney modifié améliore encore légèrement les performances.

1.4.3 Conclusion

Le modèle n-gramme lissé par la technique de Kneser-Ney interpolé modifié est la technique statistique de modélisation de langue qui est la plus utilisée à l'heure actuelle et qui est donc l'état de l'art. Elle est plus performante que les autres techniques de lissage présentées jusqu'ici.

De plus, les gains de performance apportés par les techniques de modélisation plus complexes de l'état de l'art que l'on présentera dans la prochaine section ne justifient pas le remplacement du modèle n-gramme lissé par la technique de Kneser-Ney dans les systèmes de reconnaissance de la parole. En effet, les techniques plus avancées actuelles n'apportent qu'un gain faible au prix d'une complexité de développement, de temps de calcul et de mémoire trop importante. C'est pourquoi, par son bon compromis entre performance et complexité, la technique de Kneser-Ney est l'état de l'art et celle qu'on utilise dans les systèmes de reconnaissance de la parole.

Toutefois on ne peut pas se contenter des performances du modèle de langue n-gramme lissé par la technique de Kneser-Ney. À l'heure actuelle, suivant le volume de données du corpus d'apprentissage disponible, on apprend un modèle de langue d'ordre 3, 4 ou 5. Les dépendances de mots qui sont plus éloignés que l'ordre du modèle ne peuvent donc pas être capturées et pour certains historiques les données disponibles sont si faibles qu'on ne peut qu'apprendre un modèle d'ordre 2 ce qui accentue ce problème. Cela aboutit à des phrases insensées ou qui ne respectent pas des règles de français simples, voici quelques exemples tirés d'une reconnaissance du système de référence de l'équipe PAROLE avec à gauche l'hypothèse du système et à droite la transcription de référence :

Hypothèse de reconnaissance	Transcription de référence
à ce moment là je me suis **dit** que	à ce moment là je me suis **dis** que
de façon originale des **verts** de **grand compositeur** de la musique classique	de façon originale des **airs** de **grands compositeurs** de la musique classique
pour une économie plus souple et plus **insistantes**	pour une économie plus souple et plus **efficiente**
outré des pavillons de **surgit** générale	**outre** des pavillons de **chirurgie** générale
le nouveau centre **dispense trahi** formation annuelle	le nouveau centre **dispensera une** formation annuelle
le nouveau projet **le** centre de qualification professionnelle maritime dont les travaux de construction ont été **annoncées**	le nouveau projet **de** centre de qualification professionnelle maritime dont les travaux de construction ont été **lancés**
on estime à quatre **cent** mille **noms** de caméras	on estime à quatre **cents** mille **le nombre** de caméras
bonjour à toutes et à tous **ceux** les caméras vous **regarde**	bonjour à toutes et à tous les caméras vous **regardent**
lorsque on les a interrogés qui n' **osais** pas me répondre	lorsque on les a interrogés qui n' **osaient** pas me répondre
a le droit d' avoir notre **âme inutile**	on a le droit d' avoir notre **intimité**
la peine capitale se transformant alors en **peines** de prison	la peine capitale se transformant alors en **peine** de prison
c' est plutôt rare pour ne pas dire **exceptionnelle à** la commission européenne a été **condamné** aujourd'hui **condamné** à indemniser	c' est plutôt rare pour ne pas dire **exceptionnel** la commission européenne a été **condamnée** aujourd'hui **condamnée** à indemniser

On peut voir sur ces exemples que le modèle de langue n-gramme produit de nombreuses hypothèses qui sont proches de la transcription de référence mais qui sont erronées pour des raisons qui semblent pourtant évidentes aux yeux de tout locuteur français. On voit par exemple des erreurs de type sémantique « âme inutile » au lieu d'« intimité » ou encore « verts de grand compositeur » au lieu de « airs de grands compositeurs ». On voit aussi des erreurs de type syntaxiques, par exemple des fautes d'accords de dépendances lointaines comme dans le cas d' « osais » au lieu d'« osaient » mais aussi de dépendances locales comme dans le cas de « les caméras vous regarde » au lieu de « les caméras vous regardent ».

L'état de l'art présenté ci-dessus pour les modèles de langue n-gramme traditionnels est complété au paragraphe 2.2 avec un focus sur les modèles de langue syntaxiques, qui nous intéressent plus particulièrement dans ce travail.

2

Utilisation de la linguistique

Jusqu'à présent les différentes techniques de modélisation de la langue présentées sont complètement automatiques : on commence par fournir à une boîte noire un corpus constitué de textes de la langue à modéliser puis une fois le processus d'apprentissage terminé, on obtient un modèle de langue. Ces modèles considèrent les mots comme des symboles, sans modéliser explicitement le sens de chaque mot.

La linguistique, l'étude du langage humain, propose de nombreuses théories de formalisation du langage naturel. C'est un vaste domaine qui tente d'apporter des réponses à des questions telles que :

- Qu'est-ce qu'un mot ?
- Quelles sont les relations entre les mots d'une phrase ?
- Est-ce qu'une suite de mots appartient à une langue ?
- Quel est le sens d'un énoncé ?
- Résolution d'anaphore, par exemple qui est « le » dans la phrase « Je le lui ai dit. » ?

La linguistique est d'autant plus complexe que les langues ne sont pas figées mais en perpétuelle évolution : chaque année apparaissent de nouveaux mots quand d'autres tombent en désuétude. D'autre part, même si deux locuteurs du français ont le sentiment de parler la même langue, ils ne parlent pas tout à fait la même langue. En effet, chaque locuteur possède son propre vocabulaire et respecte plus ou moins la grammaire et l'orthographe officiels.

Prenons en exemple la phrase suivante :

Mon chien a mangé mon manuscrit de thèse.

Au cours de l'apprentissage scolaire du français, on apprend un modèle linguistique formel qui nous permet de dire qu'on a ici une phrase dont le sujet est « mon chien », le verbe est « mangé » et est au passé composé et que le complément d'objet direct est « mon manuscrit de thèse ». On a également appris que « mon » est un pronom, « chien

» un nom, « le » est ici sans aucun doute un article défini (mais qu'il peut être pronom dans d'autres cas). On peut résumer ces informations sous forme d'un arbre syntaxique :

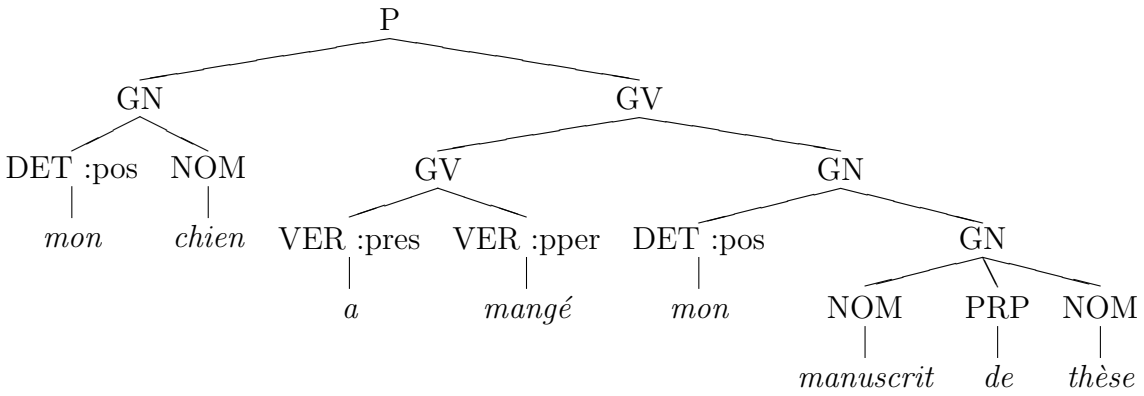


FIGURE 2.1 – Exemple d'arbre syntaxique

En linguistique, on modélise couramment la langue comme une suite de processus de plus en plus complexes :

1. la morphologie étudiant la structure des mots.
2. la syntaxe étudiant la structure des phrases.
3. la sémantique étudiant le sens des énoncés.
4. la pragmatique étudiant la relation d'un énoncé dans son contexte.

Le domaine du traitement automatique des langues (TAL) est constitué de l'ensemble des algorithmes informatiques qui ont trait au langage naturel. Le TAL est un domaine de recherche très actif, en constante évolution.

Très tôt, les chercheurs travaillant sur les modèles de langue se sont tournés vers les travaux de leurs collègues de TAL pour améliorer leurs modèles. Toutefois, le résultat ne fut pas à la hauteur de leurs espérances et les modèles les plus performants et les plus utilisés à l'heure actuelle sont purement statistiques, tels que décrit dans la section 1. Tant et si bien que Frederick Jelinek, un des grands pionniers de la reconnaissance de la parole alors à la tête du département de reconnaissance de la parole chez IBM est célèbre pour avoir dit « à chaque fois que je vire un linguiste, les performances de notre système de reconnaissance de la parole s'améliorent. » (“*Every time I fire a linguist, the performance of our speech recognition system goes up.*”)[36].

En effet, à l'heure actuelle il n'y a pas de technique utilisant des connaissances linguistiques apportant une amélioration en performances suffisante pour justifier le coût humain et informatique associé. Les modèles statistiques modélisent en grande partie tous les phénomènes linguistiques présent dans un corpus lorsqu'il y a assez de données. Pour qu'une

modélisation de connaissance linguistique apporte un gain de performance, il faut qu'elle soit suffisamment robuste. Or du fait de la grande variabilité de la langue, et encore plus de la langue orale, la modélisation d'une connaissance linguistique est en soit un problème complexe.

Dans ce chapitre, nous allons passer en revue les différentes hypothèses étudiées dans la littérature en partant des catégories grammaticales pour aller progressivement vers des connaissances de plus haut niveau. Pour des informations plus complètes sur l'utilisation de la linguistique pour la reconnaissance de la parole, vous pouvez lire l'état de l'art[32] de Huet, Sébillot et Gravier.

2.1 Morphologie

À chaque mot d'un énoncé, on peut associer une catégorie grammaticale qui décrit son rôle syntaxique. En effet, une même forme graphique d'un mot peut transcrire plusieurs mots de signification et de rôle syntaxique différents suivant le contexte. Par exemple le mot « la » peut être un déterminant comme dans la phrase « La cuisine est en bas. », ou un pronom comme dans la phrase « Je la verrai demain. » ou bien encore un nom comme dans la phrase « Donne-nous un la. ». On peut également observer que des formes graphiques différentes renvoient à une même idée ainsi les mots « espère », « espérai », « espérer » font tous référence à l'idée de l'action d'espérer respectivement à la forme verbale au présent, au passé simple et à l'infinitif.

On appelle l'étiquetage morphosyntaxique la tâche de TAL qui consiste en un algorithme qui assigne à chaque mot d'un énoncé sa catégorie grammaticale, ce qu'on appelle en informatique son étiquette morphosyntaxique. Remarquons qu'il n'existe pas un jeu d'étiquettes bien défini mais qu'il varie selon le degré de granularité désiré. Il existe même des algorithmes découvrant automatiquement des étiquettes dans un corpus. Dans la table 2.1, on trouve le jeu d'étiquettes de TreeTagger [57], un des logiciels les plus populaires pour l'étiquetage morphosyntaxique. Achim Stein en 2003 a conçu un jeu d'étiquettes pour le français.

Par exemple pour les trois phrases d'exemples citées plus haut, on obtient :

Phrase	La	cuisine	est	en	bas	.
Étiquettes	DET :ART	NOM	VER :pres	PRP	NOM	SENT
Phrase	Je	la	verrai	demain	.	
Étiquettes	PRO :PER	PRO :PER	VER :futu	ADV	SENT	
Phrase	Donne	nous	un	la	.	
Étiquettes	VER :impe	PRO :PER	DET :ART	NOM	SENT	

Code de l'étiquette	**Signification**
ABR	abbréviation
ADJ	adjectif
ADV	adverbe
DET :ART	article
DET :POS	pronom possesif (ma, ta, ...)
INT	interjection
KON	conjonction
NAM	nom propre
NOM	nom
NUM	numéral
PRO	pronom
PRO :DEM	pronom démonstratif
PRO :IND	pronom indéfini
PRO :PER	pronom personnel
PRO :POS	pronom possesif (mien, tien, ...)
PRO :REL	pronom relatif
PRP	préposition
PRP :det	mot à la fois préposition et article (au,du,aux,des)
PUN	ponctuation
PUN :cit	ponctuation de citation
SENT	étiquette de fin de phrase
SYM	symbol
VER :cond	verbe au conditionnel
VER :futu	verbe au futur
VER :impe	verbe à l'impératif
VER :impf	verbe à l'imparfait
VER :infi	verbe à l'infinitif
VER :pper	verbe au participe passé
VER :ppre	verbe au participe présent
VER :pres	verbe au présent
VER :simp	verbe au passé simple
VER :subi	verbe au subjonctif passé
VER :subp	verbe au subjonctif présent

TABLE 2.1 – Jeu d'étiquettes du TreeTagger

L'intérêt de l'étiquetage morphosyntaxique est de réduire la masse d'évènements à modéliser à l'ordre n de $|\mathcal{V}|^n$ pour le modèle n-gramme à $|\mathcal{G}|^n$, où $\mathcal{G}$ est l'ensemble des étiquettes possibles. Alors que le vocabulaire d'un système de reconnaissance contient de l'ordre de dizaines de milliers de mots, le nombre d'étiquettes est de l'ordre de quelques dizaines. Dans le système de reconnaissance de l'équipe PAROLE, le vocabulaire est constitué de 63682 mots, le jeu d'étiquettes du logiciel TreeTagger est de 33 éléments.

On retrouve la même idée de clustering développée dans les modèles de langue exploitant des classes de mots. Toutefois, l'information morphologique n'est pas souvent utilisée seule mais plutôt comme base pour extraire l'information syntaxique, plus riche.

2.2 Modèle de langue syntaxique

À partir de l'information lexicale et morphologique, extraire l'information syntaxique consiste à donner les dépendances syntaxiques entre les mots dans un formalisme syntaxique. Le programme qui réalise cette tâche s'appelle un analyseur syntaxique. Dans la communauté scientifique du TAL, les chercheurs ont beaucoup travaillé à la réalisation d'analyseurs syntaxiques et ont proposé différents formalismes [9]. Dans l'immense majorité des travaux, les analyseurs traitent une phrase de l'écrit respectant les règles grammaticales de la langue.

Il n'est donc pas chose facile d'intégrer ces analyseurs dans un modèle de langue pour la reconnaissance de la parole. En effet à l'oral, on ne connaît pas les frontières des phrases, ni les virgules, ni les majuscules (qui permettent de distinguer « pierre » de « Pierre » à l'écrit). Les énoncés contiennent souvent ce qu'on considère comme des fautes de langue mais que l'on trouve dans l'usage et qui doit donc être modélisé. Par exemple la phrase : « J'ai vu le Pierre hier. » est agrammaticale mais tout à fait possible à l'usage pour certains locuteurs. On doit également modéliser les phénomènes de disfluences, c'est-à-dire les hésitations, les répétitions, les corrections du discours, etc.

Prenons par exemple un extrait du corpus de validation :
« le le la la séparation palestinienne hamas fatah le l' ouverture de gaza vers l' égypte et et la séparation avec euh avec euh avec israël c'est ça vraiment une politique de dire que voilà les responsabilités maintenant c'est celles des palestiniens on peut pas arriver à une solution finale parce que les les palestiniens ne ne peuvent pas se parler ne peuvent pas travailler ensemble. »

Naïvement, on pourrait penser que dans un modèle de langue syntaxique du français, on peut intégrer la règle : « Deux déterminants ne peuvent se suivre et le nom qui suit un article doit être accordé en genre et en nombre avec celui-ci. ». Mais si cette règle est

valide pour un écrit en « bon » français, l'extrait du corpus de validation montre qu'à l'usage elle est fausse en raison des phénomènes de disfluences.

De nombreux chercheurs en linguistique et en TAL, à l'instar de Noam Chomsky, ont pour but de proposer des théories pour modéliser les principes fondamentaux du langage de façon explicite, comme par exemple les théories de Newton sur la gravité en physique. C'est dans cet état d'esprit que la plupart des analyseurs syntaxiques utilisent une théorie sensée couvrir la langue et considèrent qu'ils ont en entrée une phrase correcte selon la norme de la langue.

Pour le modèle de langue, qu'il soit statistique ou non, le but est tout autre puisqu'il s'agit d'obtenir un modèle le plus performant possible qui rende donc compte au mieux de l'usage de la langue, de tous les usages. L'essai de Peter Norvig [49] montre que pour les tâches du TAL les modèles statistiques sont nécessaires car l'usage lui-même est probabiliste : la langue ne suit pas de modèle fixe. Une étude approfondie de ce problème par Christopher D. Manning montre [42] en étudiant un corpus en anglais qu'à toute règle on peut trouver des usages qui ne la respectent pas, suivant une certaine probabilité. Ainsi la norme officielle de l'anglais dit que le verbe « *quiver* » (trembler) est dit intransitif mais à l'usage, bien que ce soit beaucoup moins probable, on trouve des usages transitifs.

C'est pourquoi il est difficile d'intégrer un analyseur syntaxique de l'écrit dans un modèle de langue pour la reconnaissance de la parole et d'en tirer des améliorations de performances intéressantes. En effet, à chaque fois que l'on se trouve en présence de phénomènes propres à l'oral ou à des usages ne respectant pas la norme suivie, l'analyseur syntaxique risque d'introduire des erreurs.

Néanmoins utiliser un analyseur syntaxique de l'écrit existant est le point de départ logique pour intégrer des informations syntaxiques dans un modèle de langue. Progressivement des chercheurs ont adapté les analyseurs de l'écrit en introduisant des probabilités. Nous parlerons également des travaux en cours sur la découverte automatique de grammaire.

2.2.1 Modèle de langue structuré

En 1997, Ciprian Chelba [15] a été un des premiers à intégrer de l'information syntaxique dans un modèle de langue statistique pour la reconnaissance de la parole. L'idée est d'utiliser l'analyseur syntaxique pour capturer des dépendances de longue distance intéressantes qui échappent au modèle n-gramme. Le modèle de langue ne calcule plus les probabilités $P(W)$ où W est la phrase actuelle mais les probabilités jointes $P(W, T)$ où T est l'ensemble des arbres syntaxiques. L'information syntaxique est apprise de façon automatique sur le corpus d'apprentissage brut, c'est-à-dire que le modèle apprend de

quels mots il doit faire dépendre le mot à prédire.

Dans son article [15], Chelba propose également un modèle de langue qui intègre l'analyseur syntaxique en dépendances de Collins [20] qui est lui-même appris de façon statistique. Comme l'analyseur fonctionne sur des phrases complète, le modèle n'est utilisé que pour ré-évaluer les probabilités des n meilleures phrases proposées par le moteur de reconnaissance. Malgré la complexité du modèle, il n'améliore que très faiblement les performances par rapport à un modèle bigramme, ce qui peut être dû à une modélisation syntaxique encore trop imprécise et au fait que le modèle n'est pas intégré au moteur de reconnaissance lui-même.

En 1998, Chelba et Jelinek pallient à ce problème en proposant un modèle fonctionnant de gauche à droite sur des phrases incomplètes et qui peut donc être intégré au cours de la reconnaissance. Le modèle calcule la probabilité jointe des mots, des catégories grammaticales et l'analyse syntaxique en même temps, ce qui est plus efficace que d'avoir trois processus indépendants. En raison du grand nombre de paramètres, le modèle est très lent mais ils obtiennent une baisse de la perplexité de 11% par rapport à un modèle trigramme lissé par la technique d'interpolation par suppression. Ce fut un résultat encourageant qui montre la pertinence et le potentiel de l'intégration d'information syntaxique dans un modèle de langue.

En 1999, Jun Wu et Sanjeev Khudanpur [63] étendent un modèle n-gramme en intégrant par une méthode d'entropie maximale des informations sémantiques et des informations syntaxiques. L'information sémantique modélisée est le sujet de la conversation ce qui est utile pour prédire les mots pleins, c'est-à-dire les mots les plus porteurs de sens. L'intégration de l'information syntaxique quant à elle est surtout utile dans le cas où le prédicteur syntaxique du prochain mot est hors de portée du modèle n-gramme classique. L'analyseur syntaxique utilisé est celui de Chelba et Jelinek de 1998. En combinant les trois sources d'informations pour ré-évaluer les probabilités des 100 meilleures hypothèses proposées par le modèle n-gramme, ils parviennent à diminuer la perplexité de 12%, ce qui est cohérent avec les résultats de Chelba et Jelinek, et à diminuer le taux d'erreur en mots (TEM) de 0,8 % en absolu. Il est intéressant d'observer dans les résultats que les gains de performance obtenus par la source d'information sémantique et la source d'information syntaxique sont presque additifs.

Lorsqu'on ré-évalue les probabilités des n meilleures hypothèses proposées par le modèle n-gramme par un modèle de langue plus complexe, on perd beaucoup du gain potentiel en performance du nouveau modèle de langue : le modèle n-gramme peut par exemple donner une mauvaise probabilité à une bonne hypothèse qui nécessite une modélisation des dépendances longue distance. C'est ce qui a été démontré par Jelinek et Chelba dans

[16], où, après un travail important de programmation, ils ont adapté leur moteur de reconnaissance pour qu'il donne en sortie un treillis d'hypothèses plutôt que les n meilleures hypothèses. Le modèle de langue structuré est ensuite utilisé pour explorer ce treillis à l'aide de l'algorithme A^*, la meilleure hypothèse obtenue devient le nouveau résultat de la reconnaissance. On obtient effectivement des résultats en TEM qui sont meilleurs, de l'ordre de 0,7% en absolu avec uniquement l'information syntaxique.

2.2.2 Grammaire hors-contexte probabiliste

Eugene Charniak [14] a exploré une autre manière d'intégrer de l'information syntaxique dans les modèles de langue. Il utilise son analyseur syntaxique statistique qui apprend une grammaire hors-contexte probabiliste (*probabilistic context-free grammar, PCFG*) par maximum d'entropie sur le Penn Tree-bank présenté dans [13]. L'analyseur a de bonnes performances sur l'écrit puisque sa précision et son rappel moyen sont d'environ 90%.

Il qualifie son approche d'analyse par tête immédiate (*immediate-head parsing*), c'est-à-dire que la probabilité d'un mot dépend de la tête du constituant syntaxique qui lui est immédiatement supérieur dans l'arbre syntaxique. Dans l'exemple de la figure 2.2, le mot *in* dépend du verbe *put* car le noeud vp/put est la tête immédiatement supérieure dans l'arbre syntaxique.

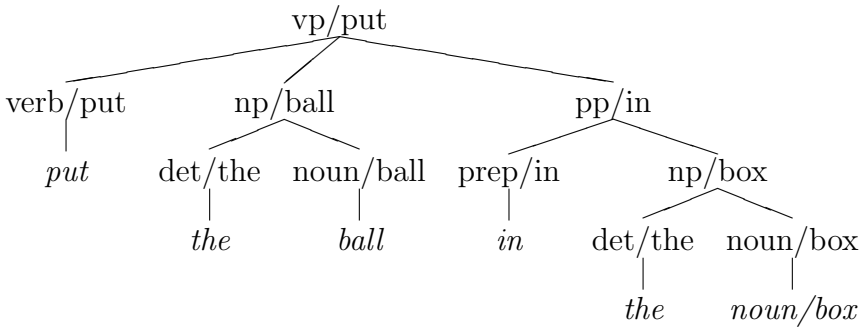


FIGURE 2.2 – Exemple d'arbre syntaxique selon [14]

En perplexité, ce modèle est plus efficace que les modèles de Chelba et Jelinek. Toutefois il possède un inconvénient de taille : il fonctionne sur des phrases entières. Il n'est donc pas possible de l'intégrer à un moteur de reconnaissance de la parole. On pourrait l'utiliser pour ré-évaluer les n meilleures hypothèses de reconnaissance mais comme l'ont montré Chelba et Jelinek, pour tirer les performances maximales d'un modèle de langage pour la reconnaissance de la parole, il faut l'intégrer dans le processus de reconnaissance lui-même.

Brian Roark a proposé [50] un autre modèle de langue intégrant un analyseur syntaxique hors contexte. Son approche d'analyse descendante probabiliste permet de réaliser des analyses de phrases incomplètes et donc d'utiliser le modèle de langue syntaxique directement dans la reconnaissance comme dans l'approche de Chelba. Cela résout le principal écueil de l'approche de Charniak qui ne fonctionne qu'avec des phrases complètes. Roark a ainsi pu réaliser des expériences de reconnaissance de la parole qui montrent qu'à volume de données d'apprentissage égal (mais très petit par rapport à un million de mots) le modèle syntaxique améliore légèrement les performances de reconnaissance.

2.2.3 Inférence grammaticale automatique

Il est coûteux de constituer des corpus annotés syntaxiquement. Pour de nombreuses langues, il n'existe donc pas de corpus annotés suffisamment volumineux pour apprendre de façon supervisée un analyseur syntaxique. De plus à l'ère d'Internet, on peut facilement constituer des corpus de centaines de millions de mots alors que le plus grand corpus annoté syntaxiquement, le Penn Tree Bank contient moins de trois millions de mots.

Une approche alternative est de construire automatiquement une grammaire à partir d'un grand volume de textes d'une langue, non annotés syntaxiquement. On peut citer la thèse de Klein [38] ainsi que les travaux de Solan *et al.* [59].

Alors qu'on est à des performances de plus de 90 % en précision et en rappel avec les approches supervisées, les approches automatiques n'en sont qu'à 60 %. Il reste donc beaucoup de travail à faire dans le domaine de l'inférence grammaticale automatique avant de parvenir à des résultats suffisamment intéressants pour pouvoir être intégrés dans un modèle de langue pour la reconnaissance de la parole.

2.2.4 Analyse et perspective

L'écriture d'une grammaire à la main ou l'annotation syntaxique d'un corpus requièrent beaucoup de temps humain et sont donc coûteuses en temps et en argent. La principale ressource en anglais, le Penn Tree Bank [43], contient 4,8 millions de mots annotés en étiquettes morphosyntaxiques et plus de 2,8 millions de mots annotés syntaxiquement. L'article indique que l'annotation syntaxique de 2,5 millions de mots peut être réalisée en un an par une équipe d'annotateurs expérimentés qui travaillent 3 heures par jour à cette tâche. Le corpus Penn Tree Bank constitue le plus grand projet d'annotation en anglais et c'est pourquoi il est toujours à la base de très nombreux travaux en TAL dont ceux de Jelinek, Chelba, Charniak et Roark que nous avons cités ci-dessus.

En français, il existe plusieurs efforts d'annotation morphosyntaxique et syntaxique de corpus. Le plus grand projet [1] est porté par Anne Abeillé, Lionel Clément, François Toussenel *et al.*. Appelé « French Tree Bank », il porte sur l'annotation d'un corpus d'un million de mots constitué de textes du journal « le Monde ». S'il est entièrement annoté morphosyntaxiquement seule une petite partie est annoté syntaxiquement, en arbres syntaxiques. On peut également citer le projet FreeBank [53] de Suzanne Salmon-Alt *et al.*, qui a pour objectif d'annoter un corpus libre d'accès d'un million de mot avec des informations multiniveaux (structurel, morphologique, syntaxique, référentiel). Un premier noyau de 100 000 mots est complètement annoté. Dans l'article [53], les auteurs indiquent qu'une annotation complète (bien plus fine que celle du Penn Tree Bank) du corpus d'un million de mots coûterait environ 180 mois-hommes, soit 7 ans et demi pour deux personnes. Plus récemment, Christophe Cerisara, Claire Garden *et al.* [11] ont lancé un projet d'annotation syntaxique en dépendances du corpus ESTER. La spécificité de ce projet est qu'il s'agit de transcriptions de radio et donc de la langue orale, qui ne respecte pas toujours les normes de l'écrit et avec ses phénomènes propres. À l'heure actuelle, 60 000 mots sont annotés.

Comme on l'a vu dans la section 2.2.3, les efforts d'inférence grammaticale automatique n'en sont qu'à leurs débuts. Les performances des analyseurs en résultant sont bien en deçà de celles des analyseurs syntaxiques classiques.

Remarquons également que les analyseurs syntaxiques supposent qu'on leur donne en entrée une phrase, ou un début de phrases pour ceux qui ont été adaptés pour réaliser des analyses partielles. Or en reconnaissance de la parole, les frontières des phrases ne sont pas connues a priori ce qui rend encore plus difficile l'intégration de l'analyse syntaxique. Il existe bien des algorithmes de segmentation en phrase mais ils ne sont pas suffisamment robustes à l'heure actuelle. Par conséquent leur utilisation conjointe avec un analyseur syntaxique risque d'introduire plus d'erreurs de reconnaissance que d'en corriger, c'est bien pour cela que les résultats de reconnaissance présentés ont été obtenus après une segmentation manuelle en phrases.

Pour le français, il faudra plusieurs années d'efforts d'annotations syntaxiques avant d'aboutir à des corpus aussi volumineux qu'en anglais. À l'heure actuelle, il n'est donc pas possible d'appliquer pour le français les approches syntaxiques développées pour l'anglais. Il faut donc des approches alternatives pour intégrer de l'information syntaxique en français.

2.3 Intégration de la sémantique et pragmatique

En linguistique, les travaux en sémantique présupposent bien souvent une analyse syntaxique des données en entrée. Or nous avons vu qu'il reste encore du chemin à faire

avant de parvenir à des analyses syntaxiques suffisamment robustes en reconnaissance de la parole grand vocabulaire. C'est pourquoi nous présenterons ici les différents problèmes qui se posent aux niveaux sémantiques et pragmatiques ainsi qu'un bref état de l'art sur ces questions du TAL.

Notons qu'il existe toutefois des travaux intégrant de l'information sémantique aux modèles de langue pour la reconnaissance de la parole. Ainsi, les modèles de langue exploitant un cache ou l'approche LSA sont des techniques stochastiques qui modélisent implicitement de l'information sémantique.

2.3.1 Anaphores

On appelle « anaphore » un élément du discours qui fait référence à un élément passé du discours. L'élément auquel l'anaphore fait référence est appelé l' « antécédent ». Voici un exemple tiré du corpus de validation d'ESTER 2 :

[...] l'opposition annonce ce soir qu'elle suspend ses actions de rue et elle a décidé de changer de tactique en appelant au boycott [...]

Le pronom « elle » est une anaphore dont l'antécédent est « l'opposition ». On parle pour ce cas de figure d'anaphore pronominale mais il existe d'autres types d'anaphore. La résolution est un problème complexe car la référence peut être très éloignée de l'anaphore, voire implicite. De plus pour pouvoir déterminer correctement la bonne référence, il faut parfois des informations de haut niveau, syntaxique, sémantique, voire pragmatique. Par exemple :
« Donnez les bananes aux guenons, même si elles ne sont pas mures, elles les mangeront puisqu'elles ont très faim. »

On comprend aisément que la première anaphore « elles » fait référence aux bananes car sémantiquement la propriété de maturité s'applique aux bananes qui sont des fruits et non aux guenons. De même, les deux dernière anaphores « elles » font bien évidemment référence aux guenons, car un animal contrairement à un végétal peut avoir faim et peut réaliser l'action de manger.

On peut se référer à l'article [45] de Ruslan Mitkov pour un état de l'art en TAL du problème des anaphores. Un atelier (*workshop*) d'EACL a été consacré à cette question en 2003. Brièvement on peut dire que les travaux ont suivi le progrès de l'intelligence artificielle en commençant par des heuristiques très simples, puis des systèmes experts pour plus récemment construire des systèmes stochastiques sans ou avec peu de connaissances *a priori*, par exemple en utilisant uniquement l'information morphologique. Les résultats

en précision/rappel des meilleurs systèmes sont au-dessus de 90 %, mais on ne peut pas parler de problème résolu car la plupart des anaphores sont immédiates et très simples à repérer. Or on a essentiellement besoin de résolution d'anaphore lorsqu'il y a ambiguïté.

Il n'existe pas de modèles de langue pour la reconnaissance de la parole grand vocabulaire qui intègre un traitement spécifique des anaphores, sans doute car l'impact potentiel sur les performances est faible, bien qu'un modèle de langue complet devra nécessairement modéliser ce phénomène. Toutefois plusieurs systèmes de compréhension de la parole (« *Speech Understanding System* ») intègrent de tels traitements pour des tâches spécifiques pour lesquelles le problème de compréhension est gérable comme la réservation d'un billet d'avion. On peut se référer par exemple aux travaux de Gerbino et Danieli [29], de Ward [62], de Issar et Ward [33], de Furui *et al.* [24].

Deuxième partie

Modèle de langue à base d'exemples

1

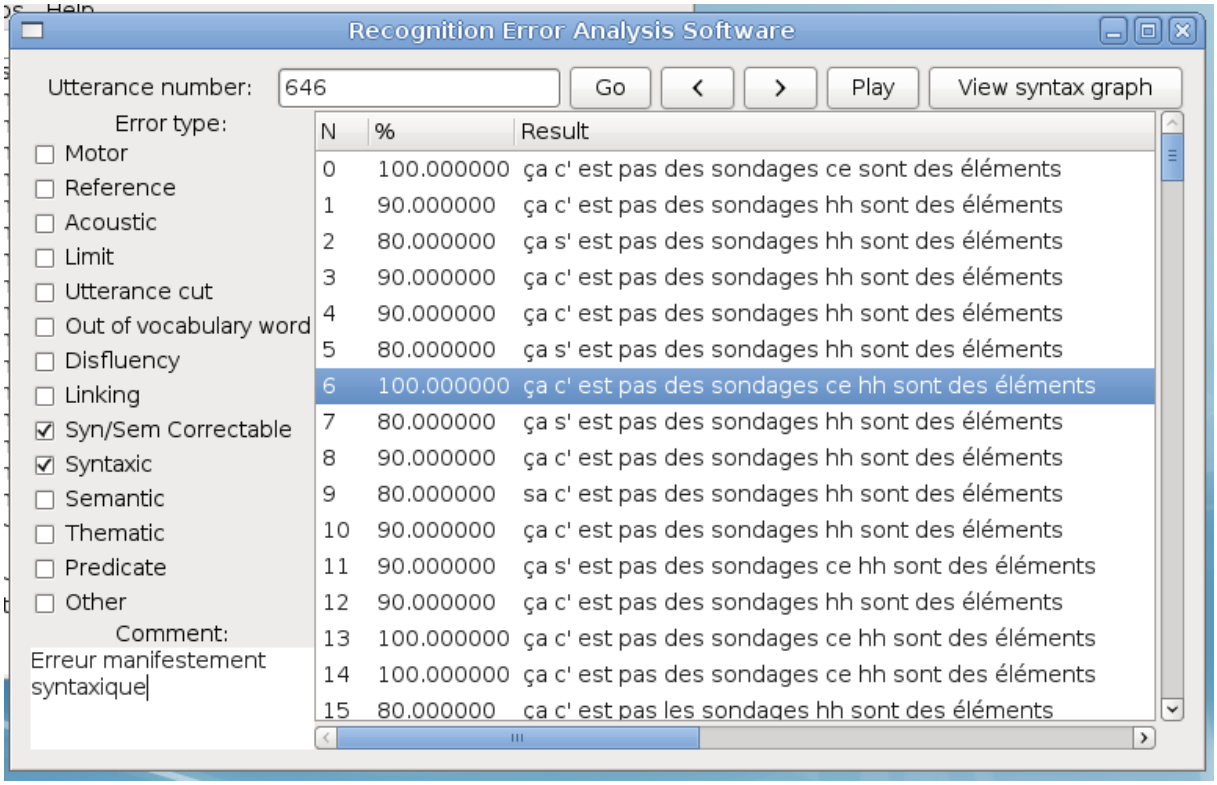
Analyse empirique des erreurs de reconnaissance

Tout en étudiant la littérature du domaine, j'ai réalisé au début de cette thèse une analyse empirique des erreurs de reconnaissance du système ANTS de l'équipe PAROLE dans le but de comprendre la nature de ces erreurs, de me familiariser avec la chaîne de traitement du moteur de reconnaissance ANTS[8] de l'équipe de recherche, de construire une intuition adaptée au problème de la modélisation de la langue pour la reconnaissance de la parole, et ainsi proposer des solutions adaptées.

Le système ANTS utilise le corpus de la campagne ESTER2, c'est-à-dire une grande quantité de textes écrits issus de journaux ainsi que du français oral d'émissions radiophoniques transcrites en quantité beaucoup plus limitée. Le système ANTS ainsi que les conditions expérimentales de la thèse sont décrits en détails dans le chapitre 5.

Pour réaliser cette analyse empirique des erreurs de reconnaissances du système ANTS, j'ai sélectionné dans le corpus d'évaluation deux heures d'émissions de radio de France Info dans lesquelles se trouvent notamment un journal et des interviews politiques.

Puis j'ai effectué la reconnaissance avec le système ANTS en demandant au système de produire pour chaque groupe de souffle (c'est-à-dire des mots qui se suivent sans silence prolongé, jusqu'à une vingtaine de mots en pratique) les 100 meilleurs candidats de résultats de reconnaissance pour pouvoir analyser les erreurs.
La démarche était constituée de deux passes d'écoute : la première visant à se familiariser avec le corpus et à réaliser une classification grossière des erreurs ; la seconde à annoter manuellement les erreurs de reconnaissance. J'ai écris un script en Ruby de 700 lignes permettant d'écouter le corpus, afficher les meilleurs candidats de reconnaissance, afficher l'analyse syntaxique produite par Syntex et d'annoter les erreurs selon la classification conçue à l'issue de la première passe d'écoute.



Le système ANTS effectue la reconnaissance par groupe de souffle et c'est naturellement que l'annotation se fait par groupe de souffle. Un type d'erreur est donc annoté comme source possible d'erreur si une des erreurs de reconnaissance dans le groupe de souffle peut lui être prêtée.

Voici la classification des erreurs adoptée :

- Erreurs de modélisation acoustique :
  - Erreur du moteur de reconnaissance : qui ne présente qu'un seul candidat manifestemant faux et inexploitable, ce qui arrive dans environ 10% des groupes de souffle, plus fréquemment sur des conversations téléphoniques. Exemple :

Référence	**je pense que nous maintenant depuis plusieurs mois** ...
Reconnu (candidat 1)	**or que nous depuis le mal dans le plus dur mois** ...

  - Erreur due à l'acoustique : la solution n'est pas dans la liste des meilleurs candidats. C'est le cas pour 73% des groupes de souffle. Exemple :

Référence	de Saddam Hussein **un temps** en poste à Genève ...
Reconnu (candidat 1)	de Saddam Hussein **encore** en poste à Genève ...

  - Erreur à la limite de groupe de souffle : présence d'une erreur sur le premier ou le dernier mot, ce qui arrive dans 35% des cas. Exemple :

Référence	**C'est cela** qu'il faudrait changer ...
Reconnu (candidat 1)	**si Sollac** il faudrait changer ...
Reconnu (candidat 2)	**si Solaire** il faudrait changer ...

- Erreur de coupure de groupe de souffle : lorsque le premier ou le dernier mot est incomplet, ce qui est le cas pour 24% des groupes de souffle. Ce n'est pas grave car le système ANTS utilise un découpage en groupe de souffle par des fenêtres temporelles recouvrantes. Exemple :

Référence	**l'objectif** c'est qu'il ...
Reconnu (candidat 1)	**objectif** ESSEC ...
Reconnu (candidat 4)	**objectif** et c'est que ...

- Erreurs de modélisation du discours :
  - Erreur de disfluence : lorsque le locuteur présente des disfluences (par exemple hésitation), ce qui se produit dans 9% des groupe de souffle. Exemple :

Référence	c'est l'épreuve **de ce fin de cette fin** de printemps ...
Reconnu (candidat 1)	c'est l'épreuve **de ce fin de de cette fin** de printemps ...
Reconnu (candidat 2)	c'est l'épreuve **de ce symbole de cette fin** de printemps ...

  - Erreur de liaison : en présence d'élision, liaison des mots, assimilations phonétiques et phénomènes apparentés, ce qui est le cas pour 8% des groupes de souffle. Exemple avec deux phonèmes « q » assimilés en un :

Référence	**donc quand** vous serez en excès de vitesse ...
Reconnu (candidat 1)	**donc au bout** son excès de vitesse ...
Reconnu (candidat 2)	**donc endosse** son excès de vitesse ...

  - Erreur sur la référence : lorsque la transcription de référence est incorrecte par rapport au son ce qui arrive dans 2% des groupes de souffle. Exemple :

Référence	où est saddam hussein **avec** son entourage ... ...
Reconnu (candidat 1)	où est saddam hussein **ainsi que** son entourage ...

- Erreurs de modélisation de la langue :
  - Mot hors vocabulaire : la reconnaissance se fait à vocabulaire fixé, donc on a forcément une erreur à chaque fois qu'apparaît dans la transcription un mot hors vocabulaire. Cela concerne 5% des groupes de souffle. Exemple :

Référence	ministres françois fillon et jean paul **delevoye** dévoilent ...
Reconnu (candidat 1)	ministres françois fillon et jean paul **donne voix** dévoilent ...
Reconnu (candidat 7)	ministres françois fillon et jean paul **ii** le voile ...

– Erreur sémantique : lorsque l'acoustique est correcte et qu'on considère qu'une meilleure modélisation de la sémantique permettrait de corriger l'erreur. C'est le cas pour 15% des groupes de souffle. Exemple :

Référence	**l'avenir** des retraites est en jeu ...
Reconnu (candidat 1)	**la venir** des retraites est en jeu ...
Reconnu (candidat 3)	**là venir** des retraites est en jeu ...
Reconnu (candidat 12)	**l'avenir** des retraites est en jeu ...

– Erreur thématique : une erreur sémantique qu'on estime possiblement corrigeable par une approche automatique de la thématique comme l'approche LSA de J.R. Bellegara[6]. Cela concerne 6% des groupes de souffle.

Référence	oui et lever le pied pour **arriver tranquillement** ...
Reconnu (candidat 1)	oui et lever le pied **pour euh aider** ...
Reconnu (candidat 2)	oui et lever le pied **pour euh rêver** ...

– Erreur syntaxique : lorsque l'acoustique est correcte et qu'on considère qu'une meilleure modélisation de la syntaxe lexicalisée permettrait de corriger l'erreur. C'est le cas pour 34% des groupes de souffle. Exemple :

Référence	la réforme des retraites **est le** dossier chaud ...
Reconnu (candidat 1)	la réforme des retraites **le** dossier chaud ...
Reconnu (candidat 2)	la réforme des retraites **est le** dossier chaud...

– Erreur de prédicat syntaxique : lorsque l'acoustique est correcte et qu'on considère qu'une modélisation des probabilités des prédicats syntaxiques permettrait de corriger l'erreur. C'est le cas pour 4% des groupes de souffle. Exemple :

Référence	où la rue est divisée sur **l'attitude à adopter** ...
Reconnu (candidat 1)	où la rue est divisée sur **l'attitude adopter** ...
Reconnu (candidat 2)	où la rue est divisée sur **l'attitude à adopter** ...

– Erreur syntaxico-sémantique : lorsque l'acoustique est correcte et qu'on considère qu'une modélisation conjointe de la syntaxe et la sémantique permettrait de corriger l'erreur. C'est le cas pour 17% des groupes de souffle. Exemple :

Référence	d'après **eux la gauche** n'a pas tiré les leçons ...
Reconnu (candidat 1)	d'après **la gauche** n'a pas tiré les leçons ...
Reconnu (candidat 2)	d'après **eux la gauche** n'a pas tiré les leçons ...

– Erreur de syntaxe ou de sémantique : lorsqu'on considère qu'une modélisation de la syntaxe ou de la sémantique permettrait de corriger l'erreur, avec plus de candidats acoustiques ou en modifiant les hypothèses proposées. C'est le cas pour 43% des groupes de souffle. Exemple :

Référence	entre la gauche **et la droite** ...
Reconnu (candidat 1)	entre la gauche **et droite** ...
Reconnu (candidat 17)	entre la gauche **et de droite** ...
Reconnu (candidat 22)	entre la gauche **et le droit** ...

Il s'agit bien sûr d'une analyse empirique et subjective puisque je suis le seul annotateur et que l'analyse ne porte que sur les 279 groupes de souffle erronés sur les 778 que comportent les deux heures d'émission de France Info annotées. Remarquons aussi que plusieurs erreurs peuvent se produire dans un même groupe de souffle. Le but n'est que d'avoir une estimation grossière de l'ampleur d'une catégorie d'erreur et donc l'impact potentiel d'une approche améliorant la modélisation de tel ou tel phénomène. Les résultats sont synthétisés dans la table 1.1.

On peut tirer plusieurs enseignements de ces résultats. Premièrement il semblerait qu'une très grande proportion des erreurs pourrait être corrigée avec un meilleur modèle acoustique : 73%. Et même si ce n'est pas l'objet de cette thèse, il y a encore énormément de travail de recherche à faire sur ce front.

On peut également constater que le système ANTS a encore des problèmes de segmentation puisque plus d'un tiers des erreurs se produisent en début ou fin de groupe de souffle. Certes le fait que la reconnaissance se fasse par des groupes de souffle recouvrants réduit ce phénomène mais il reste au final tout de même 11% des erreurs qui se produisent à cause d'une erreur de segmentation.

Les erreurs de défaut de modélisation de discours prises isolément ne concernent que peu de groupes de souffles. En effet, les phénomènes de disfluence et d'assimilation

Catégorie d'erreur	Type	Fréquence
Erreur du moteur de reconnaissance	acoutisque	10%
Erreur due à l'acoustique	acoustique	73%
Erreur à la limite de groupe de souffle	acoustique	35%
Erreur de coupure de groupe de souffle	acoustique	24%
Erreur de disfluence	discours	9%
Erreur de liaison	discours	8%
Erreur sur la référence	discours	2%
Mot hors vocabulaire	langue	5%
Erreur sémantique	langue	15%
Erreur thématique	langue	6%
Erreur syntaxique	langue	34%
Erreur de prédicat syntaxique	langue	4%
Erreur syntaxico-sémantique	langue	17%
Erreur d'origne syntaxique ou sémantique	langue	43%

TABLE 1.1 – Résumé des erreurs analysées par type

concernent respectivement 9% et 8% des groupes de souffle. D'autres phénomènes sur la prosodie, la prise de parole concernent moins de 5% des groupes de souffle. À l'heure actuelle, il n'y a pas de modélisation spécifique de ces phénomènes dans le système de reconnaissance. Et même si le potentiel d'amélioration des résultats est faible, il faudra sans nul doute traiter ces problèmes pour parvenir à des systèmes de reconnaissance réellement performants.

Remarquons qu'à hypothèses constantes, seules 17% des erreurs semblent corrigibles par une approche syntaxico-sémantique, ce qui pourrait sembler assez décourageant. De même, seuls 6% des erreurs semblent corrigibles par une approche de sémantique thématique type LSA, ce qui va dans le sens des résultats[6] de J. R. Bellegarda. Une hypothèse de travail était d'intégrer une modélisation des prédicats syntaxiques, mettant donc en lien un mot avec le mot dont il dépend le plus syntaxiquement dans le passé. Cette hypothèse ne semble pouvoir régler qu'au mieux 4% des erreurs et n'a donc pas été poursuivie plus en avant. En revanche 43% des erreurs semblent avoir, de par les hypothèses proposées par le système, pour origine un défaut de modélisation syntaxique ou sémantique.

Dans la perspective de travaux sur la sémantique et la syntaxe, plusieurs conclusions peuvent être tirées. Tout d'abord, pour avoir un impact substantiel sur les résultats de la reconnaissance, il faut des modèles plus complets qu'un modèle thématique ou qu'un modèle de prédicats syntaxiques. De plus, il apparaît clairement qu'on ne peut pas travailler uniquement sur les meilleurs candidats, puisque trop peu d'erreurs d'origine syntaxique

ou sémantique peuvent être corrigées par un modèle, même parfait, qui n'aurait en entrée que les meilleurs candidats.

Pour avoir un impact fort, il faut donc qu'une approche syntaxico-sémantique soit intégrée au cœur du moteur de reconnaissance pour guider la recherche d'hypothèse comme le fait le modèle de langue n-gramme. Cela a pour conséquence de limiter le temps de calcul disponible pour un calcul de probabilité par le modèle de langue, puisqu'il sera appelé des milliers de fois lors de la reconnaissance.

2

Modèle de similarité basé sur la théorie de transformation des chaînes

2.1 Motivation

Partant du corpus d'apprentissage, le modèle n-gramme construit un modèle probabiliste en utilisant les suites contiguës de mots de longueur 1 à n, n valant typiquement 4 ou 5 dans les systèmes états de l'art. Concrètement, cela signifie que pour prédire un mot, un tel modèle de langue utilise au mieux l'information contenue dans les 4 derniers mots de l'historique. Les mots précédents sont purement et simplement ignorés.

Or tous les exemples du corpus ne sont pas égaux : certains apportent plus d'informations car leur historique est plus proche de l'historique courant que la moyenne des historiques. Prenons par exemple la phrase « les hypothèses de cette théorie n'ont pas été étudiées », un modèle 4-gramme utilisera toutes les séquences de 4 mots commençant par « ont pas été » pour estimer la probabilité du dernier mot « étudiées ». Dans ce cas de figure d'un participe passé à accorder, il faudrait privilégier parmi les exemples du 4-gramme ceux ayant le même sujet « hypothèses » ou au moins les exemples ayant un sujet au féminin. Le problème ici est qu'il y a une dépendance syntaxique entre le mot à prédire et son sujet qui est hors de portée du modèle n-gramme.

Mais comment capturer ce genre de phénomène syntaxique lointain ? De façon plus générale, comment capturer les phénomènes linguistiques, qu'ils soient syntaxiques et/ou sémantiques, et tout spécialement ceux qui sont hors de portée du modèle n-gramme ? C'est un problème complexe car en reconnaissance de la parole, le modèle de langue a en entrée la liste de mots reconnus jusqu'à présent, et ce sans ponctuation ni majuscule. On n'a donc aucune frontière de phrase, d'ailleurs le concept de phrase est en lui-même ambigu : un locuteur et un auditeur ne placent pas toujours aux mêmes endroits les frontières de phrase. De plus, en reconnaissance de la parole on traite la transcription de la langue

orale, dans laquelle se produisent des phénomènes de disfluence tels que la correction, l'hésitation ou les répétitions qui sont beaucoup plus rares à l'écrit. Or les analyseurs syntaxiques de l'état de l'art en français attendent en entrée une phrase qui soit bien construite et complète et sont donc difficilement utilisables tels quels en reconnaissance de la parole. Des efforts sont entrepris pour créer des analyseurs syntaxiques de l'oral, mais il faudra sans doute encore beaucoup du temps et d'énergie pour aboutir à des outils suffisamment robustes et efficaces pour être intégrés dans un système de reconnaissance de la parole.

Ce qui fait la force du modèle n-gramme, c'est qu'il capture une grande partie de l'information syntaxico-sémantique de façon implicite. Il prend simplement en entrée un très grand corpus d'apprentissage de texte découpé en mots en ne faisant aucune hypothèse sur la forme des données, et sans effort d'annotation des données. L'inconvénient est qu'il ne modélise pas du tout l'information qui dépasse l'ordre du modèle, du fait de l'hypothèse de Markov. Toutefois cette hypothèse est rendue nécessaire par le fléau de la dimension (en anglais *curse of dimensionality*) : il n'y a tout simplement pas assez de données pour estimer de façon robuste un modèle n-gramme au-delà de l'ordre 4 ou 5.

À partir de ces différents constats, je me suis posé la question suivante : comment améliorer l'état de l'art pour les modèles de langue ? Pour avoir un effet positif sur les performances, il faut que la modélisation porte sur l'ensemble du corpus. Or un thésard, seul, ne peut pas entreprendre l'annotation d'un corpus de 700 millions de mots en information syntaxico-sémantique. Cela semble une évidence quand on se rend compte que simplement annoter les relations sujet-verbe en y consacrant une minute par phrase prendrait plus d'une centaine d'années-hommes. J'ai donc choisi de porter mes efforts de recherche uniquement vers des approches non-supervisées qui peuvent utiliser le corpus d'apprentissage dans son entièreté et ne demandent aucune annotation.

L'article [22] intitulé « *A Memory-Based Theory of Verbal Cognition* » de Simon Dennis a été pour moi une importante source d'inspiration. Il y présente le modèle SP qui est composé de deux types de représentation : d'une part une mémoire séquentielle modélisant les associations syntagmatiques au sein d'une phrase et capturant les régularités syntaxiques ; d'autre part une mémoire relationnelle modélisant les associations paradigmatiques au sein d'une phrase, quelle que soit la structure syntaxique de surface de celle-ci. Le contenu de ces mémoires est extrait d'un texte en le comparant avec le corpus d'apprentissage. Pour ce faire, il utilise la théorie de transformation des chaînes (en anglais *string edit theory*) pour aligner chaque phrase du texte avec les phrases du corpus d'apprentissage pour trouver les points communs et donc les informations pertinentes. La théorie de la cognition verbale présentée dans cet article est intéressante mais les expériences de validation ont été réalisées sur un très petit corpus de quelques articles

et ne sont pas applicables dans l'état au problème du modèle de langue.

Néanmoins la théorie de transformation des chaînes offre une base théorique solide pour l'extraction de plus d'informations que le modèle n-grammes pour le modèle de langue. Cette théorie est présentée en détails au paragraphe 2.2, puis un modèle de langue non probabiliste est proposé au paragraphe 2.3 dans le but d'améliorer la reconnaissance de la parole. Ce modèle est validé expérimentalement au paragraphe 2.4 dans une tâche de reconnaissance automatique de la parole grand vocabulaire. Enfin les résultats obtenus sont analysés au paragraphe 2.5 pour conclure sur la suite à mener à ces travaux.

2.2 Théorie de transformation des chaînes

La théorie de transformation des chaînes (*string edit theory* en anglais) est décrite en détails dans le livre [54] de Sankoff et Kruskal intitulé « Time Warps, String Edits and Macromolecules ». Elle est le résultat de nombreux travaux de recherche en informatique et en biologie [58] [46] [2] [41]. Soit deux suites (ou chaînes) de symboles quelconques, tels que des mots, des lettres, des acides aminées, des nucléotides, etc. La théorie de transformation des chaînes a pour but d'étudier la transformation de la première suite en la seconde. Ce processus, qu'on appelle alignement, est réalisé en utilisant différentes opérations élémentaires qui sont :

- appariement, abrégé en `A` (*matching*).
- modification, abrégée en `M` (*substitution*).
- suppression, abrégée en `S` (*deletion*).
- insertion, abrégée en `I` (*insertion*).

On peut par exemple réaliser la transformation d'une traduction d'un proverbe latin en une autre traduction :

Chacun	vaut	ce	que	valent	les	objectifs		de	son	effort.		
Chacun	vaut	ce	que	valent	les	buts	qu'	il	s'	efforce	d'	atteindre.
A	A	A	A	A	A	M	I	M	M	M	I	I

Il existe de nombreuses transformations possibles pour transformer une chaîne en une autre. Pour un alignement donné on définit la distance entre les deux chaînes comme étant le nombre d'opérations de transformation (modification, suppression et insertion). Plutôt que l'alignement `AAAAAAMIMMMII` choisi dans l'exemple ci-dessus de distance 7, on aurait pu choisir l'alignement `DDDDDDDDDDIIIIIIIIIIIII` de distance 23 qui consiste à supprimer tous les symboles de la première chaîne puis insérer tous les symboles de la seconde.

Néanmoins lorsqu'on aligne deux chaînes, c'est pour pouvoir les comparer et c'est pourquoi on s'intéresse aux alignements qui ont une distance minimale, qu'on appelle distance de Levenshtein. L'algorithme de Levenshtein [41] est un algorithme de programmation dynamique permet de calculer l'alignement optimal, celui minimisant donc la distance de transformation.

Comment utiliser cette théorie dans le cadre du modèle de langue ? Le but est, rappelons-le, d'obtenir un modèle plus efficace que le modèle n-gramme. Prenons par exemple un modèle n-gramme d'ordre 4, tous les exemples du corpus d'entraînement participant à l'estimation de la probabilité d'un 4-gramme ont pour propriété d'avoir un historique qui se termine avec les mêmes 3 derniers mots. Si on procède à l'alignement par l'algorithme de Levenshtein de ces exemples avec l'historique courant, on obtiendra par exemple :

- `M M M M A A A`
- `S S M M A A A`
- `I M M M A A A`
- `A A M M A A A`

Intuitivement les exemples ayant l'alignement `A A M M A A A` semblent plus informatifs que les autres puisqu'ils ont deux mots communs avec l'historique courant de plus que les autres. À partir de cette intuition, on va construire un premier modèle donnant d'autant plus de poids aux exemples que leur historique est similaire à l'historique actuel.

2.3 Modèle de langue non probabiliste intégrant la théorie de transformation des chaînes

Partons donc de l'hypothèse que l'alignement des exemples du corpus d'entraînement participant à un modèle n-gramme avec l'historique actuel permet d'extraire une information de similarité qui peut être utilisée pour obtenir un modèle de langue plus efficace. Pour mettre en œuvre l'intégration de la similarité dans un modèle n-gramme, il faut étudier deux problématiques :

- Comment mesurer la similarité entre l'historique d'un exemple du corpus d'entraînement et l'historique actuel ?
- Comment intégrer l'information de similarité pour rendre le modèle de langue plus performant ?

2.3.1 Principes généraux

Rappelons tout d'abord que dans un modèle n-gramme d'ordre o, l'estimation du maximum de vraisemblance est donnée par :

$$P(u_n|u_{n-o}^{n-1}) = \frac{C(u_{n-o}^{n})}{C(u_{n-o}^{n-1})}$$

où $C(u_i^j)$ est le nombre d'occurrences de la suite de mots $u_i u_{i+1} ... u_j$ qu'on note u_i^j dans le corpus d'apprentissage. Par cette estimation, chaque exemple contribue de façon égale à l'estimation de la probabilité pour l'historique u_{n-o}^{n-1}.

Avec l'hypothèse que les exemples n'apportent pas tous la même quantité d'information, on attribue donc à chaque exemple v constitué de la suite de mots $v_{i'}^{j'}$ un poids par la fonction w. Si on note $v \in \mathcal{C}$, toute sous-séquence des phrases du corpus appartenant au corpus d'entraînement $\mathcal{C}$, l'estimation du maximum de vraisemblance devient :

$$P(u_n|u_1^{n-1}) = \frac{\sum_{v \in \mathcal{C} \wedge v_{j'-o}^{j'} = u_{n-o}^{n}} w(v)}{\sum_{v \in \mathcal{C} \wedge v_{j'-o}^{j'-1} = u_{n-o}^{n-1}} w(v)}$$

On peut remarquer que si tous les exemples ont un poids identique, on retrouve l'estimation du maximum de vraisemblance du modèle n-grammes.

Comment affecter le bon poids à chaque exemple ? Pour l'apprendre automatiquement, il faut regrouper suffisamment d'exemples d'une manière ou d'une autre. Ici puisque l'on se situe dans le cadre de la théorie de transformation des chaînes, on considère l'ensemble $\mathcal{A}$ des alignements possibles et on définit $a(u, v)$ comme le meilleur alignement entre l'historique de u et l'historique de v. L'historique de v commence à partir du début de phrase et n'a pas de limite, il peut donc être long de dizaines de mots dans certains cas. On attribue donc le poids de chaque exemple comme le poids donné à son alignement et on obtient donc :

$$P(u_n|u_n^{n-1}) = \frac{\sum_{v \in \mathcal{C} \wedge v_{j'-o}^{j'} = u_{n-o}^{n}} w(a(u, v))}{\sum_{v \in \mathcal{C} \wedge v_{j'-o}^{j'-1} = u_{n-o}^{n-1}} w(a(u, v))}$$

Le poids de chaque alignement pourrait être appris par la technique du maximum de vraisemblance sur le corpus d'apprentissage. Toutefois procéder de cette manière entraînerait un surapprentissage possible des poids des alignements, c'est pourquoi nous allons apprendre ces poids au moyen de la technique de validation croisée. Il faut également se limiter aux alignements suffisamment fréquents pour être estimés.

On peut donner une estimation plus précise de la distance de transformation en attribuant à chaque opération de l'alignement un coût compris dans l'intervalle $]0, 1]$. Par exemple, il semble logique qu'il soit moins coûteux de remplacer le mot « café » par le mot « thé » plutôt que par le mot « Italie ».

Une fois l'ensemble $V = \{v \in \mathcal{C} \wedge v_{j'-o}^{j'-1} = u_{n-o}^{n-1}\}$ construit, on a pour chacun de ses éléments son identité, son estimation de probabilité classique et sa distance de transforma-

tion avec l'historique courant. On a donc un problème de régression classique : on cherche à connaître la valeur optimale d'une fonction en un point (ici la probabilité du n-gramme courant) avec pour information un certain nombre de points connus (ici les éléments de l'ensemble V).

2.3.2 Construction de l'ensemble des n-grammes similaires

Une difficulté du modèle proposé est de pouvoir construire efficacement l'ensemble des n-grammes similaires à un n-gramme donné. En effet, puisque notre modèle de langage est appris sur un corpus de texte d'environ 700 millions de mots, il contient beaucoup trop de 4-grammes pour qu'il soit réaliste de vouloir tous les comparer avec un n-gramme donné.

La solution implantée pour ce modèle est d'indexer le corpus d'apprentissage de façon à retrouver efficacement tous les n-grammes dont la distance de Levenshtein à un n-gramme $s = w_{k-n+1}...w_k$ est inférieure à p, et ce pour tout s. Les n-grammes $v \in V$ recherchés doivent nécessairement partager au moins $n-p$ mots avec s et sont de longueur comprise entre $n-p$ (correspondant au cas limite de p suppressions) et $n+p$ (correspondant au cas limite de p insertions). Les clés d'indexation d'un n-gramme s sont donc toutes les combinaisons de $n-p$ mots.

L'indexation est effectuée de la manière suivante : pour chaque n-gramme de longueur comprise entre $n-p$ et $n+p$, on génère les clés de recherche qui sont toutes ses combinaisons de $n-p$ mots. On insère ensuite chaque clé dans l'index approprié. On peut par exemple construire un index par longueur de n-gramme, mais d'autres implémentations sont possibles. Au final, le nombre total de clés dans les index est en $O(N)$, où N est le nombre de n-grammes du modèle de langage.

2.3.3 Estimation de la similarité de deux n-grammes

La distance de Levenshtein classique associe un même coût unitaire aux opérations modification, suppression et insertion. Mais dans la langue, toutes les opérations ne sont pas égales. Par exemple pour l'opération de modification de mot, si des synonymes peuvent être interchangeables, la plupart des substitutions sont impossibles. Il est donc important de pouvoir estimer plus précisément le coût de chaque opération de transformation entre 0 et 1. Par exemple, le coût d'insertion du mot « euh » correspondant à une hésitation doit être très faible puisqu'il peut arriver dans n'importe quel contexte à l'oral ou encore la substitution de « verte » par « rouge » doit être plus faible que la substitution de « verte » par « bleu » qui est incorrecte grammaticalement.

Nous introduisons deux heuristiques statistiques basées sur les occurences pour estimer le coût d'une opération de transformation : une pour les modifications et une autre

pour les insertions et suppressions. Lorqu'une de ces heuristiques ne peut pas être estimée de manière fiable à cause du manque de données, le coût maximal de 1 de la distance de Levenshtein classique est conservé.

L'heuristique du coût de modification du mot w par le mot w' est définie par le rapport du nombre d'occurences de contextes communs aux deux mots dans le corpus sur le nombre d'occurences de contextes de w. Formellement, si on note $s \in \mathcal{C}$ la propriété d'existence de la séquence s dans le corpus, on écrit :

$$cout(R(w, w')) = \frac{\#\{xy/(xwy, xw'y) \in \mathcal{C}^2\}}{0.5(\#\{xy/xwy \in \mathcal{C}\} + \#\{xy/xw'y \in \mathcal{C}\})} \tag{2.1}$$

L'heuristique du coût d'insertion ou de suppression d'un mot w est définie par le rapport du nombre d'occurences de contextes de w sur le nombre d'occurences de contextes sans w, formellement :

$$cout(I(w)) = cout(D(w)) = \frac{\#\{xy/xwy \in \mathcal{C}\}}{\#\{xy/xy \in \mathcal{C}\}} \tag{2.2}$$

Ces heuristiques sont très simples mais permettent néanmoins d'affiner les coûts d'édition efficacement. En effet, nous trouvons par exemple que le coût d'insertion ou de suppression des mots « euh » ou « écoutez » est faible ou encore que le coût de substitution des chiffres entre eux est également faible, ce qui était attendu. La distance $d(s_1, s_2)$ devient donc la somme des coûts des opérations de transformation ci-dessus.

2.3.4 Combinaison des n-grammes similaires

On a maintenant construit l'ensemble V des n-grammes similaires au n-gramme s dont on cherche à estimer la probabilité. Pour chacun des éléments de V, on a l'estimation de sa probabilité donnée par le modèle classique et sa similarité avec le n-gramme s. En faisant l'hypothèse que plus deux n-grammes sont similaires, plus leur estimation de probabilité doit être proche, on peut considérer qu'il s'agit d'un problème de régression locale. En effet, on cherche à connaître la valeur d'une fonction en un point en ayant une estimation de sa valeur en d'autres points. Ici la fonction à estimer est la probabilité d'un n-gramme et les différents points que l'on a à notre disposition sont les éléments de V.

De façon assez classique, inspiré par le modèle d'apprentissage local que décrivent Atkeson et al. dans [3], on combine les informations des éléments de V à l'aide d'une régression

par modèle à noyaux. La fonction noyau choisie ici est le noyau gaussien. Formellement, on a donc :

$$P(s) = \frac{\sum_{v \in V} \exp(\frac{-d^2(v,s)}{K^2}) P(v)}{\sum_{v \in V} \exp(\frac{-d^2(v,s)}{K^2})} \tag{2.3}$$

où K est la largeur du noyau définie par la distance du η-ième n-gramme le plus proche de s dans V (ou le plus éloigné s'il y en a moins), η est un paramètre du système optimisé sur le corpus de développement.

$P(s)$ n'est plus à proprement parler une probabilité, mais s'interprète plutôt comme un score, à la différence des techniques classiques de lissage. Néanmoins nous montrerons au paragraphe 2.4 l'intérêt d'utiliser malgré tout ce score pour améliorer les performances de la reconnaissance.

2.3.5 Intégration dans le système de reconnaissance

Pour la reconnaissance, le modèle proposé est combiné avec le modèle n-gramme classique par interpolation linéaire. L'approche la plus simple est de considérer un unique paramètre λ représentant le poids du modèle n-gramme classique pour l'interpolation.

Toutefois, afin d'augmenter la précision de l'interpolation linéaire, qui dépend de la qualité respective du n-gramme et des n-grammes similaires, nous considérons deux dépendances supplémentaires, en remplaçant le poids unique λ par l'ensemble de poids $\{\lambda_{|V|,d_{min}}\}$, où $|V|$ est la taille de V et d_{min} la plus petite distance entre V et s.

Intuitivement, le modèle des n-grammes similaires est d'autant meilleur que l'ensemble V est grand et proche du n-gramme dont on cherche à estimer la probabilité. Cet ensemble de paramètres $\{\lambda_{|V|,d_{min}}\}$ est estimé sur le corpus de développement en y minimisant le taux d'erreur en mots.

2.4 Résultats expérimentaux

Le modèle de langue proposé a été évalué en utilisant le système de reconnaissance ANTS de l'équipe PAROLE[8] et les données utilisées proviennent de la campagne ESTER2 [27]. Les conditions expérimentales sont présentées plus en détails au paragraphe 5.

Dans l'implantation de ce travail, l'indexation du corpus d'apprentissage pour construire efficacement l'ensemble des n-grammes similaires est conservé en mémoire, ce qui contraint fortement le volume de n-grammes traitables. Le modèle de n-grammes similaires a donc

été entraîné avec seulement 3 millions de n-grammes correspondant aux transcriptions des fichiers audio de la base d'apprentissage.

Nous avons entraîné deux modèles de langage classiques, l'un avec les mêmes 2,7 millions de mots que ceux utilisés pour apprendre le modèle de langage des n-grammes similaires, et l'autre avec la totalité du corpus à notre disposition avec un *cut-off* de 2 (tous les n-grammes ayant moins de 2 occurences ne sont pas intégrés dans le modèle de langue). Tous les modèles de langues sont des modèles 4-grammes, le paramètre η du modèle de langues des n-grammes similaire vaut 1, c'est-à-dire que l'on n'autorise qu'une seule opération de transformation.

Lorsque le modèle de langage des n-grammes similaires est interpolé avec le modèle de langage appris sur la totalité du corpus, les performances sont très légèrement améliorées : on passe du taux d'erreur en mots de 29,9 % avec le modèle classique seul à 29,8 % avec interpolation, ce qui n'est pas statistiquement significatif mais encourageant. Cela est raisonnable étant donné la différence de taille de base d'apprentissage entre les deux modèles. En revanche, à même taille de corpus (2,7 millions de mots), les performances sont améliorées avec un résultat statistiquement significatif. En effet, le taux d'erreur en mots est de 33,5 % avec le modèle de langage classique appris sur 2,7 millions de mots et de 32,3 % avec le modèle interpolé proposé, soit une baisse du taux d'erreur en mots de 1,2 % absolus. Les résultats sont synthétisés dans le tableau qui suit :

Table 2.1 – Taux d'erreurs en mots obtenus avec les différentes configurations testées, avec un modèle classique appris sur 3 ou 703 millions de mots seul ou interpolé avec le modèle des n-grammes similaires appris sur 3 millions de mots

Taille	Seul	Interpolé	Gain
2,7M	33,51 %	32,33 %	-1,18 %
703M	29,96 %	29,88 %	-0,08 %

2.5 Conclusion

Les résultats obtenus sont prometteurs puisqu'on obtient une amélioration statistiquement significative du taux d'erreur en mots lorsqu'on interpole un modèle de langage classique et le modèle de langage des n-grammes similaires appris sur le même nombre de mots.

Il existe plusieurs pistes d'améliorations du modèle de langage des n-grammes similaires décrit dans cette partie. Tout d'abord, améliorer l'implémentation de façon à pouvoir

entraîner le modèle sur les mêmes très gros corpus que le modèle classique. Ensuite, on peut envisager un apprentissage plus élaboré du coût d'édition de chaque opération, par exemple en prenant en compte des informations d'ordre syntaxique (par exemple étiquette morphosyntaxique) ou sémantique (par exemple proximité sémantique de deux mots), des contextes plus ou moins longs, etc. Enfin une autre piste importante est l'amélioration de la régression (équation 2.3). Il pourrait également être intéressant de filtrer V lorsqu'on y détecte des anomalies.

En conclusion, nous avons montré que l'utilisation des n-grammes similaires permet une approche de lissage plus performante que les techniques classiques de repli vers des n-grammes plus petits. Nous avons ainsi proposé un modèle de langage qui, une fois interpolé avec le modèle classique, permet d'améliorer significativement les performances d'un système de reconnaissance de la parole en taux d'erreur en mots. Nous décrivons dans la suite une extension probabiliste de ce modèle.

3

Modèle de langue de similarité

3.1 Motivation

Le modèle non probabiliste présenté dans le chapitre précédent est une preuve de concept qui montre qu'il est possible d'avoir un modèle de langue plus efficace pour la reconnaissance de la parole en utilisant la théorie de transformation des chaînes. Toutefois cette approche a principalement deux défauts : premièrement le modèle ne fait qu'exploiter un peu plus d'informations locales que le modèle n-gramme en permettant une seule opération de modification ; deuxièmement le modèle n'est pas probabiliste.

Comment modifier le modèle pour prendre en compte les dépendances à longue distance ? Prenons en exemple une citation de Marc-Aurèle : « Les effets de la colère sont beaucoup plus graves que les causes. ». Il est impossible de modéliser avec un modèle n-gramme la relation syntaxique entre le sujet « effets » et l'adjectif « graves » du fait du fléau de la dimension : il faudrait avoir assez de données pour estimer un modèle 8-gramme, qui est un espace de dimension $Card(\mathcal{V})^8$ où $\mathcal{V}$ est le vocabulaire. Dans notre système de reconnaissance de la parole, le vocabulaire est constitué de 63682 mots et le modèle 8-gramme a donc une dimension de $2,81.10^{38}$ alors que le corpus d'entraînement est constitué de bien moins d'un milliard de mots.

Or si parmi les exemples du corpus d'entraînement utilisés par le modèle n-gramme pour faire son estimation de probabilité certains ont le même sujet « effets », ils seront plus informatifs que les autres pour estimer la probabilité de l'adjectif « graves ». C'est d'ailleurs le cas puisque dans le corpus il y a 4 phrases ayant pour sujet « effets » et le trigramme « beaucoup plus graves » parmi 39 qui ont pour sujet « effets » et le bigramme « beaucoup plus ».

C'est précisément le fondement de l'approche du modèle de langue de similarité présenté dans ce chapitre : parmi les exemples du corpus d'entraînement qui participent à

l'estimation d'un modèle n-gramme certains sont plus similaires à l'historique actuel et donc informatifs que d'autres. Plus un exemple est similaire à l'historique actuel, plus il faut lui donner de poids dans l'estimation de la distribution de probabilité.

3.2 Modèle de langue probabiliste de similarité

On va maintenant présenter la formalisation du modèle de langue probabiliste de similarité. On note $sim(w,t) \in [1,+\infty[$ la similarité entre l'historique de la chaîne de mot $w = w_1 w_2 ... w_i$ dont on cherche à estimer la probabilité et une chaîne de mot de référence $t = t_1 t_2 ... t_j$ du corpus d'entraînement. Rappelons qu'on note $C(w_i^j)$ le nombre d'occurrences de la suite de mots w_i^j dans le corpus d'entraînement $\mathcal{C}$. On introduit la notion d'occurrence de similarité d'ordre n, qu'on note $C_s^n(w)$ défini par :

$$C_s^n(w) = \sum_{t \in \mathcal{C} / t_{j-n+1}^{j-i} = w_{i-n+1}^{i-1}} sim(w,t) \tag{3.1}$$

C'est-à-dire la somme des similarités attribuées à chacun des exemples d'apprentissage du modèle n-gramme d'ordre n. L'estimation de probabilité du modèle de langue de similarité à l'ordre n s'écrit donc :

$$P_s^n(w_i | w_1^{i-1}) = \frac{C_s^n(w)}{\sum_{x \in \mathcal{V}} C_s^n(w_1^{i-1} x)} \tag{3.2}$$

Il est important de remarquer que si toutes les similarités ont pour valeur 1, on retrouve l'équation classique du modèle n-gramme estimé par maximum de vraisemblance. C'est un point fort de l'approche de similarité : on n'affectera une valeur de similarité supérieure à 1 que si l'exemple contient plus d'information que celle contenue dans le n-gramme lui-même.

Comme pour les modèles de langue n-gramme, les modèles de similarité d'ordre 1 à n sont interpolés ensemble en utilisant la technique de lissage de Kneser-Ney. Le modèle d'ordre n s'écrit de façon récursive de la manière suivante :

$$\begin{aligned} P_{sabs}^n(w_i | w_{i-n+1}^{i-1}) = {} & \frac{C_s^n(w) - D(C_s^n(w))}{\sum_{x \in \mathcal{V}} C_s^n(w_1^{i-1} x)} . \frac{\lambda_{w_{i-n+1}^{i-1}}}{\lambda'_{w_{i-n+1}^{i-1}}} \\ & + (1 - \lambda_{w_{i-n+1}^{i-1}}) P_{sabs}^{n-1}(w_i | w_{i-n+2}^{i-1}) \end{aligned} \tag{3.3}$$

Les facteurs de décompte $D(C_s^n(w))$ et les facteurs $\lambda_{w_{i-n+1}^{i-1}}$ sont les mêmes que pour un modèle n-gramme lissé avec la technique de Kneser-Ney. Quant aux facteurs $\lambda'_{w_{i-n+1}^{i-1}}$ ils sont identiques aux facteurs $\lambda_{w_{i-n+1}^{i-1}}$ à la différence près qu'ils utilisent les occurrences de similarité $N_s^n(.)$. L'équation 3.3 est conçue de telle façon que la répartition de l'espace de probabilité entre les différents ordres reste la même qu'avec le modèle n-gramme classique.

Il faut également adapter les occurrences modifiées de Kneser-Ney à l'approche de similarité en modifiant $N_{1+}(w)$ par :

$$N_{1+sim}^n(w) = \sum_{v \in \mathcal{V}/N^n(vw_{i-n+1}^i)>0} \sum_{t \in T} \frac{sim(w,t)}{|T|} \tag{3.4}$$

Où T est l'ensemble des exemples du corpus d'entraînement qui finissent par la séquence vw_{i-n+1}^i. Alors que les occurrences modifiées de Kneser-Ney donnent une occurrence de 1 pour l'ensemble T dans l'approche de similarité on lui assigne la similarité moyenne de l'ensemble T.

3.2.1 Estimation de la similarité en utilisant la théorie de transformation des chaînes

Dans la description générale du modèle de langue de similarité, on a supposé que la mesure de similarité $sim(w,t)$ entre la séquence de mots w et la séquence de mots t était connue. Dans cette section, on va maintenant décrire la manière dont on calcule $sim(w,t)$ pour mesurer la similarité entre deux séquences de mots.

On peut choisir de calculer $sim(w,t)$ selon différentes approches et sources d'informations. On pourrait par exemple calculer une distance lexicale entre les mots des deux séquences ou encore faire l'hypothèse du « sac de mots » (en anglais *bag of words*) et mesurer le nombre de mots partagés par les deux séquences, sans prendre en compte leur position. Dans le modèle de langue de similarité, il a été choisi de modéliser la fonction $sim(.,.)$ par une distribution multinomiale utilisant un nombre restreint de paramètres entraînés sur le corpus de développement $\mathcal{D}$.

Contrairement à l'hypothèse du « sac de mots », dans le modèle de langue de similarité les paramètres prennent en compte la position relative des mots dans les deux séquences. En effet comme l'a montré R. Rosenfeld dans sa thèse [52], plus un mot est distant dans l'historique plus sa distribution de probabilité est éloignée de celle du prochain mot. En effet la perplexité d'un modèle bigramme distant augmente avec la distance au mot à prédire. C'est pourquoi comme pour le modèle présenté au chapitre 2, les séquences de mots w et t sont alignées en minimisant la distance de Levenshtein. Comme par exemple :

w	voilà	la	parole	n'	est	pas	la	manière	la	plus
t		la	parole		est	réellement	la	manière	la	plus
$\gamma_{w,t}$	I	A	A	I	A	M	A	A	A	A

Toutes les séquences de mots t du corpus d'entraînement qui partagent le même alignement $\gamma_{w,t}$ avec une séquence w sont regroupées dans une même classe ayant la même valeur de similarité : $sim(w,t) = F(\gamma_{w,t})$ où $F(.)$ est une fonction discrète. Ainsi cette approche permet de discriminer par exemple les alignements `I M A M A A` et `I M M A A A` et d'assigner une valeur de similarité plus importante au second alignement, ce qui est conforme à l'intuition que le second alignement est plus proche de w que le premier.

Le nombre total de paramètres dépend de la longueur des deux séquences considérées et croit exponentiellement. C'est pourquoi il faut fixer une longueur maximale avec un paramètre L et ne considérer que des alignements qui occurrent suffisamment d'un minimum décidé par un paramètre K pour pouvoir les estimer sur le corpus d'entraînement $\mathcal{C}$.

3.2.2 Choix des valeurs des paramètres

Le paramètre L a été fixé empiriquement à 12, il détermine la longueur maximale des séquences et n'est pas choisi par hasard : d'après les expériences de la thèse [52] de R. Rosenfeld si l'on considère des modèles bigrammes distants entre le mot à prédire et le i-ième mot le précédant dans l'historique on s'aperçoit que l'essentiel de l'information est contenue dans les quatre mots précédents le mot à prédire. Toujours d'après ces expériences, il y a encore de l'information, quoique bien moins, dans les modèles bigrammes distants du cinquième au dixième mot précédent le mot à prédire. Donc en fixant le paramètre L à 12, on capture l'essentiel de l'information nécessaire pour prédire le mot suivant un historique.

Le modèle apprend la valeur de la fonction $\gamma_{w,t}$ pour les alignements qui existent au minimum K fois dans le corpus d'entraînement $\mathcal{C}$. Ce paramètre est un compromis entre le nombre de paramètres $sim(\gamma_{w,t})$ à apprendre et le nombre d'occurrences pour apprendre chacun d'eux. Chaque valeur de K demande ensuite un apprentissage long et c'est pourquoi une dizaine de valeurs ont été essayées entre 10 et 10000 au début du développement du modèle et c'est la valeur de 1000 qui a été choisie car elle offre le meilleur compromis entre temps d'apprentissage et performance du modèle en termes de perplexité sur le corpus de développement.

L'ensemble des autres paramètres du modèle sont appris sur le corpus de développement $\mathcal{D}$, à savoir les valeurs $sim(.,.)$ et les facteurs de décompte. Au départ l'ensemble des valeurs $sim(.,.)$ sont fixées à 1 et le modèle de langue de similarité est alors strictement égal au modèle n-gramme lissé avec la technique de Kneser-Ney. Puis on optimise

la fonction objectif qui est la perplexité du modèle de langue, comme dans la plupart des approches existantes.

Quatre algorithmes différents ont été utilisés. Le premier algorithme itératif que j'ai implanté réalise à chaque itération :

1. les facteurs de décompte sont optimisés par une recherche par grille, pour chaque ordre, de l'unigramme au n-gramme maximal.
2. les alignements $\gamma_{w,t}$ sont triés par ordre décroissant.
3. les paramètres $sim(\gamma_{w,t})$ sont optimisés les uns après les autres par recherche par grille, du $\gamma_{w,t}$ le plus fréquent au moins fréquent.

Et ce tant que la perplexité continue à diminuer entre deux itérations d'un paramètre ε fixé à $0,01$.

Puis, à l'aide de la bibliothèque de développement libre `GSL`[25] (`GNU Scientific Library`), trois algorithmes ont été utilisés :

- un algorithme de descente de gradient, concrètement l'algorithme de Broyden-Fletcher-Goldfarb-Shanno. Selon la documentation de cette bibliothèque, c'est une méthode quasi Newtonnienne qui construit une approximation de la dérivée seconde et qui ensuite optimise les variables les unes après les autres avec la méthode d'optimisation de Newton.
- l'algorithme du Simplexe de Nelder et Mead
- l'algorithme de recuit simulé

Dans son livre « Statistical methods for speech Recognition »[35], F. Jelinek explique que l'optimisation des paramètres d'un modèle de langue statistique, quelqu'il soit, est un problème convexe qui se prête donc bien à l'utilisation d'algorithmes classiques d'optimisation de fonctions multi-variées. Cela reste vrai pour le modèle de langue de similarité puisqu'il s'agit également d'un modèle de langue statistique.

Au final, l'algorithme ad-hoc a une performance légèrement inférieure aux trois algorithmes de la `GL`[25] qui convergent vers des performances très similaires dans les expériences préliminaires. En conséquence j'ai choisi l'algorithme du recuit simulé qui est le plus simple à mettre en œuvre et qui converge le plus rapidement pour les expériences finales.

3.2.3 Structures de données

La véritable gageure pour le modèle de langue de similarité est de pouvoir construire efficacement l'ensemble des exemples du corpus d'entraînement $\mathcal{C}$ nécessaire pour l'estimation d'une distribution de probabilité du modèle.

La manière la plus courante d'entraîner un modèle n-gramme d'ordre n est de construire une structure de données sous forme d'arbres spécialisés qui contiennent la probabilité de tous les n-grammes d'ordre 1 à n du corpus $\mathcal{C}$ ainsi que les facteurs d'interpolations. C'est un processus long qui prends plusieurs heures sur une machine performante sur notre corpus d'entraînement. Mais à l'exécution, les performances en temps sont très bonnes, de complexité $O(NlogN)$ où N est le nombre total de n-grammes. Cependant cette structure de données n'est pas très pratique pour explorer différentes hypothèses de recherche de modèle de langue, puisque pour le moindre changement de paramètre il faut recommencer le processus d'entraînement pour tout le modèle de langue.

Mon implantation de modèle de langue emploie une approche hybride d'apprentissage paresseux. On commence par apprendre de façon classique un modèle de langue bigramme, en effet notre corpus d'entraînement est de taille suffisante pour estimer correctement le modèle et les occurrences sont si nombreuses qu'il serait trop coûteux à l'exécution d'employer un modèle plus complexe.

Ensuite, on construit un index du corpus d'entraînement $\mathcal{C}$ de façon à pouvoir obtenir toutes les phrases participant à un modèle trigramme. C'est-à-dire que si l'on cherche à estimer la probabilité $P(w_i|w_1^{i-1})$, alors la requête sur l'index $w_{i-2}w_{i-1}$ renverra toutes les phrases du corpus $\mathcal{C}$ qui contiennent la séquence de mots $w_{i-2}w_{i-1}$. La complexité en temps à l'exécution est en O(NlogN)+ O(R) où N est le nombre de clés trigrammes de l'index (c'est-à-dire tous les bigrammes w_1w_2 tels qu'il existe au moins un trigramme $w_1w_2w_3$) et R le nombre de phrases renvoyées. Une fois que la distribution de probabilité est estimée pour un historique, elle est conservée dans un cache. En effet c'est important pour la reconnaissance de la parole puisque le moteur va comparer différentes possibilités de mot suivant un historique donné. Sur un ordinateur personnel avec un processeur Intel Core2 Duo à 2,8 Ghz et 4 Go de mémoire, il faut compter une heure et demie pour calculer la perplexité du corpus d'évaluation.

3.3 Résultats expérimentaux

Pour évaluer l'intérêt du modèle de langue de similarité, on a calculé la perplexité du corpus d'évaluation de trois systèmes. Premièrement deux modèles de langue de base : un modèle 4-gramme et un modèle 3-gramme lissé avec la technique de Kneser-Ney, le modèle 4-gramme étant le modèle habituellement utilisé dans le moteur de reconnaissance de la parole de l'équipe PAROLE. Le modèle de langue de similarité part d'un modèle 4-gramme pour lequel on a entraîné 1456 paramètres de similarité sur le corpus de développement $\mathcal{D}$. Les résultats obtenus sont donnés dans la table 3.1 :

On peut constater une baisse de 3,54 % de la perplexité entre le système de base 4-gramme et le modèle de langue de similarité. La baisse est de 7,82 % par rapport au

TABLE 3.1 – Résultats de perplexité sur le corpus d'évaluation

Modèle de langue	Perplexité
3-gramme	156.33
Modèle de base 4-gramme	149.38
Modèle de similarité 4-gramme	144.09

modèle 3-gramme.

3.4 Conclusion

Le nouveau modèle de langue de similarité permet de prendre en compte des historiques beaucoup plus longs que les modèles n-grammes classiques dans l'estimation de probabilité d'une chaîne de mots. Pour ce faire le modèle de langue de similarité aligne en minimisant la distance de Levenshtein l'historique de la chaîne de mot actuelle avec les historiques de toutes les phrases du corpus d'entraînement participant au modèle 3-gramme. Les exemples sont regroupés lorsqu'ils partagent le même alignement, c'est-à-dire la suite d'opérations de transformation. Chacun de ces groupes a une valeur de similarité propre qui est utilisée pour modifier la distribution de probabilité du modèle de langue.

Ce modèle part de l'hypothèse que l'ordre des mots est une information importante lorsque l'on compare deux historiques. Ce modèle permet en particulier de discriminer des historiques qui ont des mots en commun dans un passé plus ou moins lointain mais aussi de discriminer des historiques qui ont plus ou moins de mots en commun.

La mesure de similarité implantée dans le modèle actuel est très générale et peut être améliorée de multiples façons. Premièrement, l'information lexicale n'est aucunement utilisée dans le calcul de similarité. Il semble raisonnable de penser qu'intégrer cette information pourrait améliorer la qualité de la mesure de similarité par exemple en tenant compte des synonymes et d'autres relations lexicales entre les mots. Plutôt que d'utiliser une mesure continue entre les mots, on peut aussi utiliser des classes, comme par exemple l'étiquette morphosyntaxique des mots. Au niveau plus haut de l'alignement, il faudrait également prendre en compte pour chaque opération l'identité du mot qui est apparié, inséré, remplacé ou supprimé. Cela pourrait être intéressant, par exemple le fait d'insérer un mot d'hésitation comme « euh » peut moins influer sur la similarité de deux historiques que d'insérer un adjectif comme « bleu ». Une autre manière de considérer l'information au niveau de l'alignement serait, plutôt que de considérer la distance entre les mots de façon purement topologique, utiliser une distance syntaxique pour hiérarchiser différemment les mots dans l'historique.

4

Combinaison de modèles de langue spécialisés syntaxiques

4.1 Motivation

La vaste majorité des modèles de langue stochastiques proposés dans la littérature comme alternative au modèle n-gramme ont pour point commun de regrouper des exemples autrement que par l'hypothèse de Markov. Par exemple, les modèles syntaxiques vont regrouper les exemples qui partagent des mêmes traits syntaxiques. Les modèles à base de réseaux de neurones vont regrouper les exemples dont les mots ont une représentation distribuée proche. Les modèles de classe vont regrouper les exemples dont les mots appartiennent à la même classe. Quant au modèle de langue de similarité proposé dans le chapitre 3, il regroupe les exemples partageant la même édition optimale vers le contexte dont on estime la probabilité. Toutes ces approches, bien que de nature différente, partagent une même idée : inclure de l'information plus spécialisée que le modèle n-gramme pour mieux estimer la distribution de probabilité du prochain mot. De prime abord, cela semble une bonne idée mais cette approche comporte en réalité un écueil : l'information supplémentaire modélisée est potentiellement superflue pour un sous-ensemble de la distribution de probabilité réelle. Et par conséquent le coût de modélisation de cette information supplémentaire peut être plus important que le bénéfice qu'elle apporte.

Pour illustrer ce problème, prenons par exemple la distribution de probabilité à estimer $P(w|\text{je voudrais acheter une chemise vert})$. Suivant les mots du vocabulaire, l'intuition donne le sentiment que les mots de l'historique sont plus ou moins pertinents pour l'estimation de la distribution de probabilité. Ainsi pour les mots « foncé » et « clair » modéliser le sujet ou même le groupe verbal est contre-productif car cela décroît l'importance accordée aux exemples du corpus d'apprentissage tels que « il voulait repasser sa chemise vert clair » ou « M$^{\text{me}}$ Tartempion portait une improbable chemise vert foncé ». En revanche, pour les mots « faire », « aller », il semble primordial de modéliser le groupe

verbal pour faire apparaître la dépendence syntaxique avec le mot « voudrais ».

C'est vraisemblablement pour cette raison que les résultats du modèle de langue de similarité présenté au chapitre précédant et plus généralement des autres modèles de langue stochastiques proposés comme alternative au modèle n-gramme ne sont pas plus probants. En effet, de façon générale le modèle de langue de similarité est un progrès par rapport au modèle n-gramme car il modélise plus d'information présente dans le corpus. Mais en réalité cette approche entraîne trop souvent une surspécialisation qui dégrade les performances pour le sous-ensemble de la distribution de probabilité pour lequel l'information supplémentaire modélisée est inutile, comme nous l'avons montré ci-dessus.

En réalité, on se retrouve confronté à la problématique mise en évidence par le principe de parcimonie (ou rasoir d'Ockham) énoncé par Aristote : « Il vaut mieux prendre des principes moins nombreux et de nombre limité ». Ou plus poétiquement comme Antoine de Saint-Exupéry l'écrit[21] : « Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher ». Dit de façon plus moderne, lorsqu'on a plusieurs hypothèses de même vraisemblance, il faut favoriser la plus simple. Cela a été démontré formellement par l'induction de Solomonoff [44]. Un modèle efficace ne doit donc modéliser que ce qui est nécessaire quand c'est nécessaire. Toute modélisation d'information superflue entraîne une dégradation des performances, car pour estimer les distributions de probabilités il faut plus de données. C'est pour cette raison que l'on peut contaster une si faible amélioration des résultats des modèles de langue stochastiques proposés dans la littérature comme alternative au modèle n-gramme.

C'est pourquoi je propose dans ce chapitre une approche de la modélisation de la langue différente de l'existant et qui prend en compte ce principe de parcimonie. Dans la section 4.2, une première expérience est présentée pour rendre plus claire l'intuition derrière l'approche et démontrer son intérêt potentiel sur des données concrètes. Puis dans la section 4.3, je présenterai la généralisation de la méthode puis sa mise en œuvre pratique dans la section 4.5.

4.2 Démonstration de l'intérêt de l'approche

Avant d'entreprendre des expériences complexes sur un modèle global, il est toujours intéressant de valider l'intérêt d'une nouvelle approche sur un sous-ensemble du problème du modèle de langue. En effet, travailler sur un sous-problème demande moins de temps et permet de mieux appréhender le nouveau problème, même si bien sûr le résultat sur un sous-problème ne peut être qu'une indication de résultat sur le problème global.

L'hypothèse exposée précédemment est que pour obtenir un modèle plus efficace, il

n'est pas suffisant de prendre en compte plus d'informations, il faut également que cette information soit pertinente. Or on se trouve dans des situations de « conflits d'intérêts » dans le sens où pour prédire un verbe, intuitivement il est plus important de bien modéliser le sujet alors que pour un adjectif, intuitivement il est plus important de bien modéliser le nom auquel il se rattache. Et donc si l'on fait un choix global d'information supplémentaire à modéliser, comme c'est le cas pour la plupart des modèles alternatifs au modèle n-gramme, cela va améliorer les performances pour une partie de la distribution de probabilité à modéliser mais au prix d'une dégradation des performances du reste de la distribution de probabilité.

Le sous-problème que j'ai choisi d'explorer est l'utilisation d'un modèle spécifique modélisant le sujet pour le sous-ensemble du vocabulaire constitué par les verbes à l'indicatif et le modèle classique n-gramme pour le reste du vocabulaire, lorsque le sujet n'est pas modélisé par le modèle n-gramme. De façon formelle, on n'étudie donc que les distributions de probabilités suivantes :

$$\{P(w_i|w_1w_2...w_{i-1})/sujet \in \{w_1, w_2, ..., w_{i-1}\}\}$$

Où *sujet* est le sujet de la phrase actuelle, en considérant que le corpus est découpé en phrases.

Le modèle proposé est le suivant :

$$P(w_i|w_1w_2...w_{i-1}) = \begin{cases} P_{ngr}(w_i|w_{i-3}^{i-1}) \quad si \quad w_i \in V - VI \\ \lambda P_{ngr}(w_i|w_{i-3}^{i-1}) + (1-\lambda)P_{vbi}(w_i|w_1^{i-1}) \quad si \quad w_i \in VI \end{cases} \tag{4.1}$$

Où VI est l'ensemble des mots du vocabulaire pouvant être un verbe à l'indicatif et où le modèle spécialisé P_{vbi} est défini comme suit :

$$P_{vbi}(w_i|w_1w_2...w_{i-1}) = \frac{\#\{w_iw_jw_{j-1}w_{j-2}...w_{j-k}\}}{\sum_{w_i \in VI} \#\{w_iw_jw_{j-1}w_{j-2}...w_{j-k}\}}$$

Où les mots $w_jw_{j-1}...w_{j-k}$ représentent l'information principale du groupe sujet constitué par le sujet principal, son déterminant si c'est un nom commun ou deux noms propres apposés (typiquement le cas d'un nom de personne). Par exemple dans la phrase : « le ministre de l'économie demande une enquête », cela sera la séquence « le ministre ». Pour la phrase « Jules Martin anxieux et jaloux ajoute », ce sera « Jules Martin ».

Pour déterminer la catégorie grammaticale de chaque mot, on utilise le POS Tagger de Stanford[61] [60] ce qui implique de travailler sur l'écrit, c'est-à-dire de garder la casse des caractères, puisque il a été entraîné sur de l'écrit.

Il reste à pouvoir déterminer le sujet principal dans une séquence de mots, ce qui est en soi un problème très complexe. Pour cette expérience d'exploration, j'ai écris un certain nombre de règles à la main, qui ne couvrent évidemment qu'une partie des situations réelles mais suffisantes pour le but de l'expérience. Ces règles sont des expressions régulières comportant des étiquettes morphosyntaxiques et des mots qui identifient le sujet principal. Elles ne sont pas non plus nécessairement fiables mais couvrent bien les situations les plus courantes. Voici quelques exemples de règles ainsi définies :

TABLE 4.1 – Règles d'extraction du sujet principal d'un verbe à l'indicatif

Règle	Sujet principal	Exemple
D A NC A* VI	[0,2]	le petit chien mignon mange
NP NP A*	[0, 1]	Jean Valjean impatient prend

Le corpus d'apprentissage est le corpus Le Monde de la campagne ESTER2 (428 millions de mots), le corpus de développement le corpus Ester du corpus d'apprentissage de la campagne d'ESTER2 (2,7 millions de mots) et le corpus d'évaluation Ester2 est le corpus d'évaluation de celle-ci (59562 mots).

Le corpus de développement est utilisé pour apprendre le coefficient d'interpolation λ entre le modèle n-gramme et le modèle spécialisé pour les verbes indicatifs tel que présenté dans cette section.

À partir des règles définies, on optimise la perplexité en modulant le coefficient d'interpolation λ sur les événements pour lesquels le modèle spécialisé s'applique sur le corpus de développement. Au final, on calcule sur le corpus d'évaluation la perplexité du modèle n-gramme standard et du modèle spécialisé interpolé avec le modèle n-gramme sur les 440 évéments pour lesquels le modèle spécialisé s'applique. On obtient les résultats suivants :

TABLE 4.2 – Résultats de perplexité sur le corpus d'évaluation

Modèle de langue	Perplexité
4-gramme	171
Modèle spécialisé	112

Cette première expérience est très concluante puisque la perplexité est fortement diminuée d'environ 27% pour les évènements pour lesquels le modèle spécialisé s'applique. Toutefois ce résultat n'est calculé que sur 440 évènements, ce qui est un nombre très faible d'évènements et le résultat est rendu possible par l'injection de connaissances des

règles syntaxiques d'extraction du sujet principal. Il est intéressant de noter que les règles sont locales et indépendantes du contexte syntaxique, comme celles définies par le modèle n-gramme qui se limitent aux n derniers mots. La contribution ici est d'avoir choisi plus efficacement que par l'hypothèse de Markov les mots à modéliser pour estimer la distribution de probabilité pour un sous-ensemble de la distribution de probabilité (ici les verbes à l'indicatif) grâce à une dépendance syntaxique locale.

Ce résultat n'est pas généralisable en l'état. En effet, l'intuition correcte dans le cas présent dans l'écriture des règles, ne le sera pas toujours. En effet *errare humanum est* et les règles dépendent trop de la personne qui les écrit. De plus, il ne s'agit ici que d'un seul modèle spécialisé pour les verbes à l'indicatif alors que potentiellement d'autres modèles spécialisés peuvent être construits pour d'autres catégories de mots.

Une solution générale doit donc pouvoir proposer une solution à ces problèmes, en introduisant d'une part pour chaque catégorie de mot un modèle spécialisé qui intègre des dépendances locales plus pertinentes que celles modélisées par le modèle n-gramme, quand cela améliore les performances du modèle de langue. D'autre part, les règles d'extraction de l'information pertinente elles-mêmes doivent être apprises automatiquement, en utilisant au mieux les informations présentes dans le corpus d'apprentissage. Une proposition de solution est présentée dans les sections suivantes de ce chapitre.

4.3 Principes généraux

Forts du résultat positif de l'expérience précédente utilisant un modèle spécialisé uniquement pour l'ensemble des verbes à l'indicatif, nous proposons un modèle de langue le plus général possible. Nous nous basons donc sur l'hypothèse que les mots de l'historique ont une pertinence plus ou moins importante pour chacun des mots de la distribution de probabilité à estimer et sur le principe de parcimonie. L'approche radicalement différente que je propose consiste donc en l'introduction d'un modèle de langue lui-même constitué de plusieurs modèles spécialisés utilisant différemment l'information de l'historique. Chaque modèle spécialisé a pour objectif de mieux estimer la distribution de probabilité que le n-gramme modélise pour un sous-ensemble du vocabulaire. Chacun des modèles spécialisés est une mise en œuvre d'une évolution du modèle de langue de similarité présenté dans le chapitre 3, restreinte donc à un sous-ensemble du vocabulaire. Les sous-ensembles sont définis disjoints pour plus de simplicité, bien qu'il n'y ait pas de contrainte théorique.

On peut donc par exemple proposer un modèle utilisant de l'information syntaxique pour mieux modéliser les verbes à l'indicatif lorsque le sujet syntaxique est hors de portée du modèle n-gramme comme cela a été démontré dans la section précédente. On peut également penser à un autre modèle spécialisé pour les adjectifs modélisant quant à lui le

nom auquel se rattache syntaxiquement l'adjectif en question et ainsi de suite.

Plusieurs questions se posent immédiatement :
- Combien de modèles spécialisés faut-il construire et sur quelle base ?
- Pour chaque modèle quelle information pertinente faut-il modéliser ?
- Comment répartir l'espace de probabilité entre les différents modèles spécialisés ?

4.3.1 Nombre de modèles spécialisés

De façon générale, comme pour toute approche d'apprentissage automatique combinant plusieurs modèles, on peut continuer à créer de nouveaux modèles spécialisés s'appliquant à un sous-ensemble des évènements tant que cela rend plus efficace le modèle de langue ; c'est-à-dire tant que la perplexité sur le corpus de développement du modèle global diminue. Chaque nouveau modèle spécialisé doit regrouper un sous-ensemble cohérent de l'espace de probabilité à estimer, un sous-ensemble qui ait un même besoin de modélisation d'une information spécifique pertinente.

4.3.2 Information modélisée

On peut construire un modèle spécialisé sur n'importe quelle base du moment qu'il y ait une cohérence de besoin d'information pour un sous-ensemble donné de l'espace de probabilité. Il y a alors matière à créer un modèle spécialisé. On peut donc concevoir des modèles spécialisés qui soient à base :
- thématique : par exemple la présence de mots du vocabulaire scientifique entraîne une plus grande probabilité d'apparition de ceux-ci.
- sémantique : par exemple en modélisant plus spécifiquement les entités nommées.
- syntaxiques : par exemple en modélisant les relations sujet - verbe - complément ou tout autre trait syntaxique.
- etc

Dans un premier temps, je propose d'explorer uniquement la construction de nouveaux modèles spécialisés sur des critères syntaxiques. Et ce tout d'abord car il y a un vrai besoin : comme on a pu le voir au chapitre 1, de nombreuses erreurs de reconnaissance de la parole sont d'origine syntaxique. Ce n'est pas vraiment une surprise puisque de nombreux phénomènes syntaxiques distants sont hors de portée de modélisation d'une chaîne de Markov d'ordre 4 ou 5. Par exemple, l'accord du sujet « maire » avec le participe passé « annoncé » dans la phrase : « Le maire PS de la Rochelle a finalement annoncé une nouvelle candidature. ». De plus, le potentiel d'une telle modélisation a été démontré dans la section précédente.

Il semble donc raisonnable de partir de l'hypothèse par exemple que pour tout l'espace de probabilité consacré aux participes passés, il est bénéfique de modéliser la relation sujet-

verbe si elle est présente dans l'historique et extractible. Ou, autre exemple, qu'il semble raisonnable que pour l'espace de probabilité consacré aux adjectifs, il soit intéressant de modéliser la relation nom-adjectif. Par exemple le sujet « équipe » avec son attribut « seule » dans la phrase : « l' équipe se sent parfois bien seule et démunie ».

4.3.3 Modèles syntaxiques

Pour construire un modèle de langue spécialisé syntaxique, il faut extraire de l'information syntaxique de l'historique actuel. Ce n'est pas chose aisée pour de l'information orale : il n'y a ni segmentation des phrases, ni ponctuation, ni casse des caractères dans la suite de mots prononcée. On y trouve également des fautes de français comme « la ville a été pris par », des phénomènes de disfluence comme le mot « euh », ou des corrections du genre « du non de la république » (où "du" ne devait pas être énoncé). Le plus logique semble d'utiliser des outils existants mais comme nous allons le voir ce n'est pas si simple étant donné l'état actuel des outils syntaxiques.

Analyseurs syntaxiques

Le plus logique lorsque l'on veut étudier les relations syntaxiques d'un énoncé est d'utiliser un analyseur syntaxique. En effet, un analyseur syntaxique est un logiciel qui étant donné une phrase en entrée produit en sortie une structure de donnée très riche décrivant la syntaxe de celle-ci.

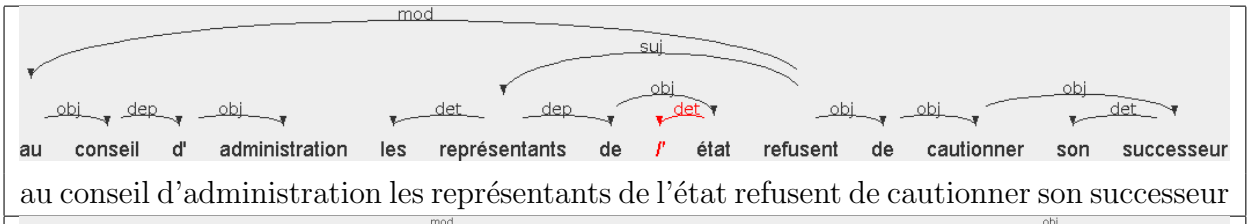
Cela reproduit par exemple ce que chacun de nous a fait à l'école primaire à savoir à partir d'une phrase construire un arbre syntaxique. Pour pouvoir être intégré à un moteur de reconnaissance de la parole, l'analyseur syntaxique doit pouvoir traiter des phrases incomplètes. Par exemple pour qu'un modèle syntaxique puisse aider à choisir le participe passé adéquat de la phrase « un accord en ce sens a été signé entre la commune urbaine de la ville et la fondation américaine genesis at the crossroads » il faut pouvoir réaliser l'analyse partielle du début de phrase « un accord en ce sens a été ».

Or l'analyse syntaxique est déjà une tâche fort complexe pour des phrases de l'écrit grammaticalement justes et correctement ponctuées.

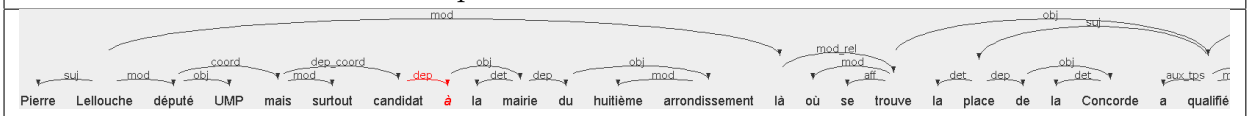
Des travaux de recherche sont actuellement entrepris pour réaliser des analyses syntaxiques de l'oral. Par exemple les résultats de JSafran[12], l'analyseur syntaxique en dépendances spécialisé pour le traitement de l'oral développé au sein du LORIA dans l'équipe SYNALP, se basant entre autres sur l'analyseur syntaxique stochastique MALT[48].

Dans la table 4.3, on peut observer les résultats de l'analyse syntaxique obtenue avec JSafran sur quelques exemples tirés du corpus d'évaluation de la campagne ESTER 2.

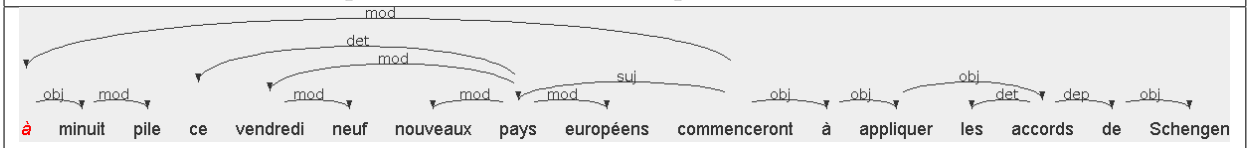
Comme on peut le constater les analyses sont tout à fait pertinentes et JSafran parvient à extraire par exemple la relation sujet-verbe, même lorsque cette relation est distante.



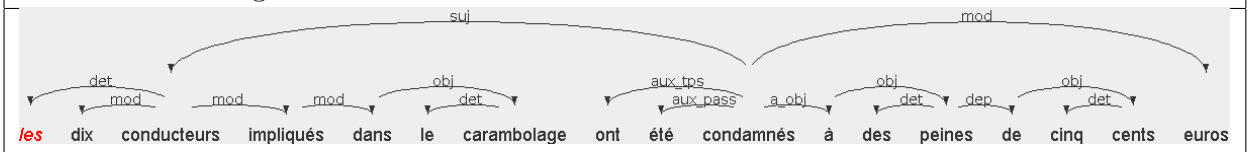
au conseil d'administration les représentants de l'état refusent de cautionner son successeur



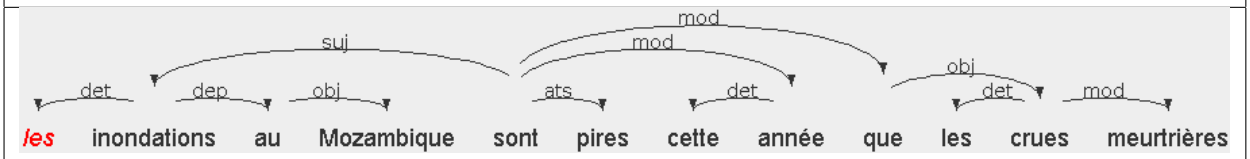
Pierre Lelouche député UMP mais surtout candidat à la mairie du huitième arrondissement là où se trouve la place de la Concorde a qualifié



à minuit pile ce vendredi neuf nouveaux pays européens commenceront à appliquer les accords de Schengen



les dix conducteurs impliqués dans le carambolage ont été condamnés à des peines de cinq cents euros



les inondations au Mozambique sont pires cette année que les crues meurtrières

TABLE 4.3 – Analyses syntaxiques de JSafran sur des exemples tirés du corpus d'évaluation de la campagne ESTER 2

Malheureusement il est difficile de parvenir à de tels résultats sur des sorties de moteur de reconnaissance de la parole qui contiennent de nombreuses erreurs, les analyseurs syntaxiques supposant que tous les mots sont corrects. Et donc à l'heure actuelle, les résultats de l'analyse ne sont pas suffisamment robustes pour être exploitables en l'état. Par exemple dès que le sujet et le verbe sont séparés de plusieurs mots, quelques erreurs de reconnaissance suffisent à perturber l'analyseur, précisemment là où la modélisation syntaxique a le plus à apporter par rapport au modèle n-gramme. Ainsi la relation sujet-verbe n'est pas extraite de façon suffisamment fiable pour être exploitée sur les données

de la campagne ESTER2.

Étiqueteurs morphosyntaxiques

Étant donné qu'il semble difficile et peu robuste d'intégrer un analyseur syntaxique, le pas suivant est d'extraire de l'information syntaxique à un niveau inférieur et donc de procéder à un plus simple étiquetage morphosyntaxique du texte.

En effet c'est le processus qui précède à l'analyse syntaxique dans une chaîne classique de TAL sur de l'écrit. Pour étudier cette possibilité, j'ai essayé trois étiqueteurs morphosyntaxiques prêts à être utilisés :

1. le LIA tagger [10] développé à Avignon.
2. le Tree Tagger [55] [56], très utilisé.
3. le Stanford Postagger [61] [60], introduit plus récemment.

Les essais menés sur les données de la campagne ESTER2 comme on peut le voir sur le tableau 4.4, montrent qu'il y a presque toujours une erreur ou deux d'étiquetage par phrase. Cela s'explique pour les mêmes raisons que pour l'analyse syntaxique : des mots inconnus, des disfluences, l'absence de ponctuation, l'absence de majuscules. Toutefois les résultats sont bien plus fiables que ceux d'une analyse syntaxique. Remarquons également que le traitement des mots inconnus est mieux géré par les étiqueteurs Stanford POS Tagger et Tree Tagger qui attribuent au mot inconnu l'étiquette qui leur semble le plus probable au regard du contexte.

Pour palier à ces problèmes, il faudrait pouvoir ré-entraîner l'étiqueteur pour prendre en compte les spécificités du problème de l'étiquetage de l'oral. Mais si le Stanford POS Tagger et le LIA Tagger sont des logiciels libres, les corpus d'apprentissage de ces étiqueteurs morphosyntaxiques ne le sont pas. Cela complique grandement le processus d'adaptation de l'étiqueteur pour cette tâche. À ce titre, on peut remarquer aussi que chaque étiqueteur propose également son propre jeu d'étiquettes morphosyntaxiques, plus ou moins précis, tiré très certainement de son propre corpus, par définition plus petit que la somme des corpus de tous les étiqueteurs.

Il serait en outre intéressant de modifier les étiqueteurs de façon à obtenir en sortie un treillis probabiliste d'étiquettes pour mieux traiter les ambiguïtés plutôt que d'utiliser uniquement la meilleure hypothèse qui, comme nous l'avons montré, contient relativement souvent des erreurs.

Malgré ces erreurs, les étiqueteurs morphosyntaxiques sont le meilleur outil disponible pour lever les ambiguïtés lorsqu'un même mot peut avoir plusieurs étiquettes morphosyntaxiques. Par exemple dans la phrase : « l' opposition annonce ce soir », le mot « annonce » ici verbe à l'indicatif présent mais pouvant également être dans d'autres contextes un nom,

est correctement étiqueté par tous les logiciels. Le problème est que leur utilisation requiert la casse des caractères et rend donc impossible l'application directe pour la reconnaissance de la parole. Et par conséquent restreint leur utilisation à de l'écrit pour le moment.

Expressions régulières sur traits syntaxiques

En raison des problèmes de robustesse et de fiabilité des analyseurs syntaxiques et des étiqueteurs morphosyntaxiques disponibles, il me semble intéressant de proposer une approche alternative. Premièrement étant donné le manque de robustesse et les nombreuses erreurs des analyses syntaxiques réalisés sur du français oral issue des sorties du moteur de reconnaissance de la parole, du coût prohibitif en temps processeur, et de leur besoin de beaucoup de contexte, j'ai décidé de ne pas utiliser d'analyseur syntaxique. Deuxièmement, si l'information plus bas niveau apportée par un étiqueteur morphosyntaxique est plus fiable que celle apportée par un analyseur syntaxique, elle comporte toujours des erreurs. C'est pourquoi j'ai décidé d'utiliser un étiqueteur morphosyntaxique tout en faisant le maximum pour en fiabiliser les résultats.

L'approche proposée doit donc permettre de modéliser une partie de l'information syntaxique et ce de la manière la plus fiable possible. Elle consiste à procéder à un étiquetage morphosyntaxique en utilisant le Stanford POS Tagger puis de fiabiliser et d'enrichir sa sortie en utilisant le lexique libre Morphalou [51]. En effet, ce lexique permet d'obtenir la liste des traits syntaxiques possibles de chaque mot. Les étiquettes définies par Morphalou sont données en annexe A 6.

Une fois l'étiquetage réalisé, notre approche consiste à définir des expressions régulières sur les étiquettes pour chaque modèle spécialisé pour extraire de l'information syntaxique pertinente. Plutôt que de chercher à comprendre tout le contexte syntaxique (ce qui est très difficile comme on l'a vu en étudiant les sorties d'analyseur syntaxiques), l'important ici est de ne chercher à extraire que quelques informations locales au sens syntaxique (mais pouvant être distantes en position).

Par exemple, l'expression régulière « `pronom_personnel verbe_actif déterminant nom` » permet d'extraire un triplet syntaxique `predicat_actif(agent, prédicat, patient)` où `agent` est `pronom_personnel`, `prédicat` est le lemme de `verbe_actif`, c'est-à-dire sa forme infinitive et `patient` est le lemme de `nom`, c'est-à-dire sa forme au masculin singulier.

Cette expression régulière s'applique par exemple sur la phrase « madame Pilar del Castillo vera je vous rappelle l'heure de votre rendez-vous de ce cet après-midi » d'où on extrait donc le triplet syntaxique `predicat_actif(je, rappeler, heure)`.

En appliquant de telles expressions régulières sur l'ensemble du corpus d'apprentis-

sage, on peut ainsi en extraire une base de données d'information syntaxique qu'on peut par la suite interroger. Par exemple, sachant que le sujet est `roman`, quels sont les verbes susceptibles d'apparaître ?

L'utilisation d'un regroupement des mots, ici celui induit par les étiquettes morphosyntaxiques, est nécessaire pour pouvoir utiliser un historique plus long tout en ayant suffisamment d'échantillons dans le corpus pour pouvoir en extraire de l'information de façon robuste. Toutefois, l'identité de chaque mot est porteuse d'informations propres, informations qu'on choisit d'ignorer si l'on utilise uniquement l'étiquette morphosyntaxique. C'est pourquoi je propose de construire des expressions régulières utilisant l'identité des mots pour les plus fréquents d'entre eux (notamment les mots outils) et les étiquettes morphosyntaxiques pour les autres.

Chaque expression régulière doit être précise et robuste. Pour cela, chaque règle n'extrait qu'un petit nombre d'informations syntaxiques à la fois et les mots inconnus sont simplement ignorés. De plus comme les expressions régulières ne modélisent qu'un faible nombre d'informations syntaxiques, elles sont par essence locales et donc relativement insensibles aux mots les entourant. Si une phrase est trop complexe pour être couverte par les expressions régulières, elle n'est tout simplement pas traitée. En effet, dans de telles situations, il vaut mieux préserver les résultats du modèle n-gramme qu'on sait relativement robuste que de vouloir dogmatiquement appliquer un modèle différent dont on ne soit pas sûr de la fiabilité.

Il est intéressant de remarquer que des expressions régulières permettent de capturer très simplement des relations de longue distance, par exemple : « `déterminant nom (complement_nom)* verbe_actif déterminant nom` » où `complement_nom` est défini très incomplètement (et naïvement pourrait-on dire) par « `(de|du) déterminant? nom)? adjectif*` ». Cette expression régulière s'applique par exemple à la phrase « mais aujourd'hui les pédiatres du ministère de la santé cherchent les solutions ».

Après application des expressions régulières sur chacune des phrases du corpus d'apprentissage, on obtient alors une base de données d'informations syntaxiques qui n'est pas exhaustive mais qui est précise, ce qui lui permet d'être complémentaire par rapport à l'information capturée par le modèle n-gramme.

On peut ensuite construire un modèle de langue spécialisé pour un sous-ensemble de l'espace de probabilité qui s'applique lorsqu'une expression régulière correspond à l'historique actuel et qui utilise l'information idoine capturée sur le corpus d'apprentissage par l'étape précédente.

Par exemple si l'expression régulière `déterminant nom (complement_nom)* verbe_actif déterminant` s'applique à l'historique actuel alors pour le sous-ensemble de l'espace de probabilité assigné aux noms, on peut utiliser la base de connaissance des triplets syntaxiques `predicat_actif(agent, prédicat, patient)` extraite du corpus d'apprentissage pour mieux prédire le patient.

Ainsi si l'historique actuel est par exemple « madame Pilar del Castillo vera je vous rappelle l' », alors on peut utiliser les triplets syntaxiques `predicat_actif` où l'agent est `je`, le prédicat `rappeler` pour obtenir l'ensemble des patients observés dans le corpus d'apprentissage. Puis on filtre cet ensemble pour ne garder que les patients qui peuvent suivre le déterminant `l'`. On définit la probabilité d'un agent comme le rapport de fréquence entre son nombre d'occurrences et le nombre d'occurrences de tous les agents.

4.3.4 Répartition de l'espace de probabilité

S'il y a de multiples manières de choisir la répartition de l'espace de probabilité, une solution simple et sensée que nous proposons consiste à utiliser le modèle n-gramme classique pour estimer l'espace de probabilité à attribuer à chaque modèle spécialisé.

Par exemple on assigne la masse de probabilité assignée par le modèle n-gramme à l'ensemble des noms communs au modèle de langue spécialisé syntaxique prédisant le patient lorsqu'on observe un prédicat actif suivi d'un déterminant.

4.4 Formalisation

Soit le vocabulaire du modèle de langue défini par l'ensemble V, pour définir un modèle de langue constitué par une combinaison de l modèles spécialisés, on définit une partition du vocabulaire par des sous-ensembles T_i, $i \in [0, l-1]$, tel que donc :

$$V = \sum_{0 \leq i < l} T_i \quad \wedge \quad T_i \cap T_j = \emptyset, \ \forall (i,j) \in [0, l-1]^2, i \neq j \tag{4.2}$$

On décompose la fonction de répartition de probabilité P du modèle de langue par sous-ensemble T_i :

$$P(w_j | w_1 w_2 ... w_{j-1}) = \sum_{0 \leq i < l} P_{|T_i}(w_j | w_1 w_2 ... w_{j-1}) \tag{4.3}$$

Sachant bien sûr que seule la fonctions restreintes à l'ensemble T_i auquel appartient le mot w_j renverra une probabilité non-nulle.

Définissons ensuite la fonction de probabilité restreinte à un ensemble T_i comme une combinaison de la fonction de probabilité donné par le modèle n-gramme classique noté

P_{ngr} et la fonction de probabilité donné par le modèle spécialisé P_{spe}, ce que l'on écrit par :

$$P_{|T_i}(w_j|w_1^{j-1}) = \begin{cases} \lambda P_{spe_i}(w_j|w_1^{j-1}) + (1-\lambda)P_{ngr}(w_j|w_1^{j-1}) \; si \; 1_{\Phi_{t_i}}(w_1^{j-1}) = 1 \\ P_{ngr}(w_j|w_1^{j-1}) \; si \; 1_{\Phi_{t_i}}(w_1^{j-1}) = 0 \\ 0 \; si \; w_j \notin T_i \end{cases} \tag{4.4}$$

La fonction indicatrice $1_{\Phi_{t_i}}(w_1^{j-1})$ permet de distinguer les historiques traitées par le modèle spécialisé P_{spe_i} des historiques pour lesquels le modèle spécialisé n'apporte pas d'information et pour lesquels on n'utilise donc que l'estimation donnnée par le modèle n-gramme.

Enfin la fonction spécialisée regroupe des exemples du corpus d'apprentissage de façon différente du modèle n-gramme ce que l'on peut noter par :

$$P_{spe_i}(w_j|w_1^{j-1}) = \frac{\#\Phi_{t_i}(w_1^j)}{\sum_{w_k \in T_i} \#\Phi_{t_i}(w_1^{j-1}w_k)} \tag{4.5}$$

La fonction Φ_{t_i} regroupe de manière spécialisée les exemples du corpus d'apprentissage pour les mots appartenant au sous-ensemble T_i du vocabulaire. On peut imaginer par exemple pour l'ensemble VI des verbes conjugés à l'indicatif une fonction Φ_{VI} qui ignore les adjectifs du nom principal du sujet, qu'on peut définir par :

$$\Phi_{VI}(w_1^j) = \#w_k w_{k+1} w_{k+2}^{j-1} w_j \tag{4.6}$$

Où w_k est un déterminant, w_{k+1} est un nom commun et w_{k+2}^{j-1} est une liste d'adjectif, éventuellement vide. La fonction indicatrice est définie par $1_{\Phi_{VI}}$ par w_k est un déterminant et w_{k+1} est un nom commun, dans tous les autres cas, seul le modèle n-gramme s'applique. Ainsi dans l'exemple donné, on introduit un modèle spécialisé qui utilise de l'information syntaxique donné par un étiqueteur syntaxique pour regrouper ensemble les exemples qui partagent le même sujet principal nom commun et qui ignore donc les adjectifs associés.

Ce mécanisme très général permet d'introduire soit manuellement des fonction Phi_{T_i} soit de les apprendre automatiquement. Puisque les modèles spécialisés sont toujours interpolés avec les modèles n-grammes et que ce facteur est appris automatiquement en utilisant le corpus de développement, ils permettent toujours d'améliorer les performances du modèle de langue obtenu par rapport au modèle n-gramme. Dans le pire des cas, où le modèle spécialisé n'est pas pertinent, le facteur d'interpolation λ_{T_i} obtenu sera si petit que le modèle spécialisé sera ignoré. En revanche, plus le modèle spécialisé apporte d'information par rapport au modèle n-gramme, plus le facteur λ_{T_i} sera important. Ainsi les facteurs d'interpolation permettent d'adapter l'utilisation des modèles spécialisés à la quantité d'information supplémentaire qu'ils apportent effectivement sur le corpus de développement par rapport au modèle n-gramme.

4.5 Mise en œuvre

Pour étudier le potentiel de cette approche de modélisation de la langue par des modèles spécialisés syntaxiques, je propose de mettre l'accent sur deux situations mal gérées par le modèle n-grammes :

- les participes passés des constructions passives
- l'attribut du sujet

En effet pour ces deux situations, on a fréquemment des dépendances de longue distance comme on peut le voir sur ces quelques exemples tirés du corpus de développement :

- un mémorandum d' entente dans le secteur de l' offshoring a été signé
- la commission européenne a été condamnée aujourd'hui
- un employé local de la section suisse de médecins sans frontières a été tué
- des prévisions à l' horizon 2010 ont également été exposées
- alan johnston libéré mercredi dernier est attendu
- la première phase des opérations est terminée
- celles qui ne rentrent pas dans les rangs seront nationalisées a déclaré le chef de l' état

4.6 Apprentissage automatique des sous-modèles syntaxiques

On commence par apporter de l'information en définissant le cadre des sous-modèles.

4.6.1 Participe passé des constructions passives

Ce modèle specialisé doit donc pouvoir prédire le participe passé d'une construction passive. Pour cela, il faut construire une base de donnée syntaxique des relations prédicat-patient du corpus, c'est-à-dire de bigrammes `(predicat, patient)`. Et ce en conservant l'information de l'expression régulière d'origine, ainsi que la forme (active ou passive) de la construction. Un extrait de la base est donné pour le patient `thèse` dans la table 4.5.

Le modèle spécialisé est défini sur le sous-ensemble du vocabulaire constitué par les participes passés à l'exception de ceux des verbes d'états (être, demeurer, rester, ...) lorsque l'historique s'il est suivi d'un participe passé correspond à une des expressions régulières de la forme passive.

L'ensemble de ces expressions régulières peut être construit à la main, mais cela devient vite fastidieux (et la précision d'un humain diminue à mesure que son ennui s'accroit). La solution préconisée consiste donc à utiliser une procédure automatique pour construire ces expressions, selon les règles suivantes.

Une première règle est que c'est toujours l'expression régulière la plus longue qui prime, à la manière d'un algorithme glouton. Par exemple pour la phrase : « la thèse du ministère de la défense a été rejetée par le tribunal d'instance » la simple expression régulière `la nom_commun a été participe_passé_féminin` s'applique tout comme l'expression régulière `la nom_commun complement_du_nom* a été participe_passé_féminin`. Puisque la seconde est plus longue, c'est celle qui est choisie pour cette phrase.

Une deuxième règle est que les formes passives traitées par le modèle spécialisé défini sont de la forme `verbe_avoir_indicatif été participe_passé` et que le patient extrait fait au plus un mot.

Une troisième régle est qu'une forme active est définie par un verbe à la forme active et que le patient apparaît après ce verbe.

On peut partir ensuite de l'hypothèse que le patient précède immédiatement la construction verbale à la forme passive et qu'il suit immédiatement la construction verbale à la forme active, que c'est soit un nom commun, soit un nom propre ou soit un pronom. Puis à chaque itération de l'algorithme de construction des expressions régulières, on considérera une expression régulière comme valide si la perplexité du modèle de langue baisse.

On pourrait se poser la question de savoir si un tel algorithme permettra de trouver par exemple que dans l'expression régulière « avait adverbe été participe passé » on peut extraire les mêmes informations pour le modèle spécialisé que dans la construction « avait été participe passé ». Il s'agit ici d'une transformation de degré qui ne consiste qu'à ajouter un adverbe à une expression régulière existante et qui est suffisamment fréquente dans les données pour être extraite.

Rappelons enfin que dans une expression régulière, on utilise soit l'étiquette morpho-syntaxique, soit le mot lui-même s'il fait partie des n mots les plus fréquents.

Nous n'avons que brièvement décrit ci-dessus une piste de recherche permettant de généraliser les premiers résultats présentés dans ce chapitre sur l'intégration de modèles de langue spécialisés pour la reconnaissance automatique de la parole, en présentant comment appliquer ce principe à la conception d'un modèle dédié à l'analyse des participes passés des formes passives. Bien entendu, ce travail doit être étendu à de nombreuses autres structures grammaticales avant de pouvoir évaluer son impact général sur le système. Il s'agit cependant d'un travail de longue haleine, et les contraintes de la thèse ne permettant pas de le réaliser dans les temps, ce chapitre constitue donc une perspective importante pour le travail réalisé dans cette thèse.

Référence	avec euh la médiation de kofi annan au kenya
LIA Tagger	avec PREP euh ADV la DETFS médiation NFS de PREPADE **kofi MOTINC annan MOTINC** au PREPAU **kenya MOTINC**
Stanford POS Tagger	avec P **euh N** la D médiation N de P kofi N **annan A** au P kenya N
Tree Tagger	avec ADV euh INT la DET,ART médiation NOM de PRP **kofi NOM annan ADJ** au PRP,det **kenya NOM**
Référence	contre la dernière gagnante du concours valérie bègue la réunionnaise de vingt-deux ans
LIA Tagger	contre PREP la DETFS dernière AFS gagnante NFS du PREPDU concours NMS **valérie MOTINC bègue AMS** la DETFS réunionnaise AFS de PREPADE vingt CHIF **- MOTINC** deux CHIF ans NMP
Stanford POS Tagger	contre P la D dernière A **gagnante A** du P concours N **valérie V bègue V** la D réunionnaise A de P **vingt-deux D** ans N
Tree Tagger	contre PRP la DET,ART dernière ADJ gagnante NOM du PRP,det concours NOM **valérie ADJ bègue ADJ** la DET,ART réunionnaise NOM de PRP vingt-deux NUM ans NOM
Référence	alors euh moi je je vais pas me faire ici leur porte-parole moi j'ai été témoin de de de euh
LIA Tagger	alors ADV euh ADV moi PPOBJMS je PPER1S je PPER1S vais V1S pas ADVPAS me PPOBJMS faire VINF ici ADV leur DETFS **porte NFS - MOTINC parole NFS** moi PPOBJMS j' PPER1S ai VA1S été VPPMS témoin NMS de PREPADE de PREPADE de PREPADE euh ADV
Stanford POS Tagger	alors ADV **euh C** moi PRO je CL je CL vais V pas ADV me CL faire V ici ADV leur D porte-parole N moi PRO j' CL ai V été V témoin N de P de P de P **euh N**
Tree Tagger	alors KON euh INT moi PRO,PER je PRO,PER je PRO,PER vais VER,pres pas ADV me PRO,PER faire VER,infi ici ADV leur DET,POS porte-parole NOM moi PRO,PER j' PRO,PER ai VER,pres été VER,pper témoin NOM de PRP de PRP de PRP euh INT
Référence	par exemple des déboulonneurs l'idée de barbouiller des panneaux
Référence	ici baba camara réunit toutes les filles de l' orphelinat
LIA Tagger	ici ADV **baba NMS camara MOTINC** réunit V3S toutes les AINDFP filles NFP de PREPADE l' DETMS orphelinat NMS
Stanford POS Tagger	ici ADV **baba V camara V** réunit V toutes A les D filles N de P l' D orphelinat N
Tree Tagger	ici ADV **baba ADJ** camara NOM **réunit VER :simp** toutes PRO,IND les DET,ART filles NOM de PRP l' DET,ART orphelinat NOM

TABLE 4.4 – Exemples d'étiquetage morphosyntaxique sur le corpus d'évaluation

thèse	défendre	passif	la nom_commun a été participe_passé
thèse	soutenir	actif	il verbe_indicatif la nom_commun
thèse	rejeter	passif	cette nom_commun avait été participe_passé

TABLE 4.5 – Exemples d'une base de donnée syntaxique de bigrammes (`predicat, patient`)

5

Conditions expérimentales

Dans la section 1.3, on a vu que la qualité d'un modèle de langue pour la reconnaissance de la parole est généralement évaluée par deux mesures : la perplexité qui se calcule sur du texte et le taux d'erreur en mots du système de reconnaissance de la parole sur des enregistrements audios.

Travaillant pour l'équipe PAROLE, j'ai naturellement utilisé le système de reconnaissance de la parole en français de l'équipe : ANTS[23]. Il est basé sur le logiciel libre de reconnaissance Julius développé au Japon, principalement par Akinobu Lee[40] qui prend en entrée un modèle acoustique, un modèle de langue et un fichier audio et retourne en sortie la transcription du fichier audio en mots.

Le système ANTS est une solution complète qui apprend plusieurs modèles acoustiques et de langues, segmente le flux audio en parole/non parole puis en locuteurs et enfin effectue la reconnaissance pour obtenir au final une transcription la plus complète possible de ce qui est dit dans un fichier audio. C'est un système complexe, état de l'art, avec de nombreuses optimisations, qui est le fruit d'années-hommes de travail au sein de l'équipe. Il nous sert à étudier la validité des hypothèses de recherches étudiées.

Les données utilisées proviennent de la campagne d'évaluation ESTER 2[28] de 2009. Le but de cette campagne était de comparer différents systèmes de reconnaissance de la parole en français notamment sur une tâche de transcription de transmission d'émissions de radios françaises :

- France Inter
- France Culture
- France Info
- RFI
- Radio Classique

Ainsi que des émissions étrangères plus difficiles à transcrire de part les accents franco-

phones et le vocabulaire spécifiques :

- RTM (Radio Télévision Maroc)
- Radio Africa numéro 1

Pour l'apprentissage, on dispose de transcriptions d'émissions de radio. Si elles sont proches des émissions à reconnaître et donc idéales pour l'entraînement du système, elles coûtent chères à réaliser. Or pour entraîner un modèle de langue efficace, il faut de très grande quantités de textes, c'est pourquoi on dispose également de données textuelles provenant de journaux. La table 5.1 donne la taille en mots des différents corpus utilisés dans nos expériences :

TABLE 5.1 – Taille des corpus

-

Nom	Type	Taille
Journal « Le Monde »	écrit	428 millions de mots
Journal « L'humanité »	écrit	148 millions de mots
Presse Plus	textuel	34 millions de mots
TNS	transcriptions audios	91 millions de mots
Émissions de radios d'ESTER 2	transcriptions audios	2,7 millions de mots
Total du corpus d'entraînement $\mathcal{C}$	mixte	703 millions de mots
Total du corpus de développement $\mathcal{D}$	transcriptions audios	40871 mots
Total du corpus d'évaluation	transcriptions audios	59562 mots

Conclusions

Le travail de recherche présenté dans ce mémoire se focalise sur l'étude des similarités entre les n-grammes de mots estimés sur un corpus textuel et classiquement utilisés pour calculer les probabilités du modèle de langage dans les systèmes de reconnaissance automatique de la parole. L'objectif est de proposer de nouvelles manières d'exploiter ces similarités pour améliorer l'estimation des probabilités n-grammes en réduisant la sensibilité de ces modèles à la quantité de données requise pour les apprendre de manière fiable. L'approche traditionnelle pour atteindre cet objectif est de réaliser un lissage dans l'espace des paramètres. Nous proposons une alternative dans ce mémoire, qui consiste à considérer, pendant la phase de reconnaissance, les exemples d'apprentissage individuellement, afin de ne retenir que les plus proches de l'exemple d'apprentissage pour estimer la probabilité n-gramme cible. Autrement dit, plutôt que de lisser les modèles en moyennant et en interpolant les exemples d'apprentissage selon leur degré de similarité, nous nous inspirons des méthodes d'apprentissage dites « à base d'exemples » pour filtrer les données d'apprentissage à prendre en compte en fonction de leur pertinence par rapport à un contexte précis apparaissant pendant la phase de test. Bien sûr, une seconde phase d'interpolation plus traditionnelle entre modèles est ensuite réalisée afin d'éviter les problèmes liés au sur-apprentissage typiques des approches à base de mémoire, mais l'essentiel de la contribution de ce travail concerne la première partie, à savoir comment identifier les contextes pertinents, les exemples d'apprentissage à retenir et comment intégrer ces informations dans un modèle de langage probabiliste classique.

L'un des éléments les plus importants de cette approche est la notion de similarité, qui est développée en détails dans la deuxième partie de ce mémoire, en particulier au chapitre 2 ainsi qu'au paragraphe 3.2.1. Il est relativement commun dans le monde du traitement automatique des langues, et en particulier de la parole, de s'appuyer sur l'hypothèse simplificatrice du « sac de mots » (*bag of words*) pour définir la similarité lexicale, notamment dans de nombreuses approches distributionnelles comme la LSA. L'alternative la plus utilisée en modèles de langage est de considérer une similarité binaire et un contexte linéaire, comme dans le cas du n-gramme. Nous avons proposé dans ce travail une notion de similarité plus riche qui tient mieux compte de la réalité des structures surfaciques de la langue et notamment des insertions de mots optionnels qui arrivent très fréquemment à l'oral. Nous nous sommes donc tournés vers la distance de Levenshtein et la théorie de transformation des chaînes pour modéliser la similarité entre les mots, ce qui réduit de beaucoup les contraintes des hypothèses de modélisation par rapport aux contextes linéaires traditionnels.

L'une des principales difficultés des modèles à base de mémoire est liée au fait que ces modèles diffèrent fondamentalement des modèles probabilistes. En effet, alors que les premiers considèrent chaque observation individuellement et dans sa spécificité, les seconds manipulent des ensembles les plus grands possibles de données pour calculer des

distributions. Il est donc toujours difficile et délicat de tenter la combinaison de ces deux paradigmes si dissemblables. Cette difficulté a été résolue en deux étapes successives dans ce mémoire : tout d'abord au chapitre 2 en considérant les mesures issues de méthodes à base de mémoire dans leur cadre définitoire non probabiliste, puis au chapitre 3 en construisant des distributions de probabilités basées sur ces mesures qui soient plus facilement intégrables dans les modèles probabilistes qui constituent l'essentiel des systèmes de reconnaissance automatique de la parole. Notons que dans les deux cas, l'intégration a été menée jusqu'au bout, c'est-à-dire jusqu'au développement d'un système de reconnaissance automatique de la parole grand vocabulaire incluant ces nouveaux modèles au cœur d'un module de langage fondé sur des n-grammes interpolés selon la méthode état de l'art de Kneser-Ney.

Le chapitre 4 ouvre de nouvelles perspectives en proposant d'étendre l'application du paradigme des modèles à base d'exemple, utilisé dans les précédents chapitres pour définir la notion de similitude entre n-grammes, au modèle de langue lui-même. Le principe est donc d'exploiter la spécifité des structures surfaciques non plus « seulement » pour estimer la similarité entre deux n-grammes, mais également pour construire un nouveau modèle de langage dédié à un contexte particulier. De plus, nous proposons dans ce cadre d'enrichir encore la mesure permettant de caractériser ces contextes en passant de la métrique de Levenshtein à des caractéristiques véritablement syntaxiques. Le cadre théorique permettant de réaliser ceci est explicité, mais nous n'avons malheureusement pas eu le temps de valider expérimentalement totalement cette approche. Nous tenions néanmoins à présenter ce travail dans le mémoire, car il constitue un aboutissement de la réflexion menée tout au long de la thèse sur la notion de stucture spécifique et sur son intégration dans les systèmes de reconnaissance automatique de la parole. Ce chapitre constitue également une ouverture et des perspectives particulièrement intéressantes pour poursuivre les travaux décrits dans le mémoire.

Les contributions principales des travaux décrits dans ce mémoire sont les suivantes :

- Analyse détaillée des erreurs issues d'un système de reconnaissance automatique de la parole état de l'art en français, selon leur source d'erreur estimée, en s'interrogeant en particulier sur le gain en performances réalisable selon qu'un type d'erreur ou un autre est traité en priorité. En particulier, l'impact des erreurs dont la composante syntaxique et sémantique est prépondérante est estimé.
- Proposition d'un nouveau paradigme d'estimation des probabilités des modèles de langue fondé sur l'apprentissage à base d'exemples qui permette de sélectionner les exemples d'apprentissage les plus pertinents pour chaque exemple d'évaluation.
- Proposition d'une nouvelle mesure de similarité entre n-grammes basée sur la distance de Levenshtein plus riche que les mesures classiquement utilisées dans ce domaine.

- Développement d'une structure de données efficace permettant d'interroger des bases d'exemples très grandes en temps « raisonnable », si ce n'est en temps réel et qui permet d'explorer de nouvelles hypothèses de recherche sans une longue phase d'apprentissage comme c'est le cas classiquement.
- Proposition d'une nouvelle approche permettant d'intégrer ces scores mesurant la similarité entre n-grammes au sein d'un système de reconnaissance automatique de la parole probabiliste.
- Implémentation et intégration de ces approches dans un système état de l'art sans compromis quant aux nombreuses optimisations classiquement réalisées dans un tel système.
- Proposition d'une nouvelle approche permettant d'étendre ce modèle de langage à base d'exemples à la définition et à l'intégration de modèles dédiés à la modélisation de structures syntaxiques ou sémantiques spécifiques.

Le cheminement scientifique réalisé au long de ce travail de thèse et décrit dans ce mémoire peut se résumer ainsi. L'hypothèse de départ étant la disponibilité d'un système de reconnaissance automatique de la parole état de l'art, la première étape a consisté à étudier les erreurs de reconnaissance typiques commises par ce système et à identifier celles pouvant potentiellement être corrigées grâce à une amélioration du modèle de langue impliquant l'utilisation d'information de plus haut niveau, en particulier syntaxique et sémantique. Suite à cette étude, il est apparu que de nombreux phénomènes langagiers dépendaient fortement d'un contexte spécifique qui n'est pas régulier, ne peut se résumer simplement aux n mots précédents et qu'il est donc difficile de prendre en compte avec des modèles aussi simples que les n-grammes. La spécificité des structures à traiter nous a amené à considérer une approche duale du paradigme statistique dominant dans le domaine, les modèles à base de mémoire, dont la caractéristique est justement de bien traiter les phénomènes spécifiques. Ce choix est par ailleurs justifié par deux autres considérations : tout d'abord, la complémentarité théorique existant entre les approches statistiques et celles à base d'exemples qui plaide en faveur de leur combinaison, la première modélisant les grandes tendances tandis que la seconde se focalise sur les exceptions ; et également par plusieurs initiatives internationales qui ont également conclu positivement quant à l'importance des modèles à base d'exemples, comme par exemple les partenaires du projet européen Sound2Sense.

Nous avons donc tout d'abord développé la notion de similarité entre n-grammes, qui permet d'identifier les contextes particuliers pris en compte dans notre modèle, puis combiné ces informations via un noyau Gaussien pour calculer un nouveau score issu seulement des contextes les plus semblables à l'exemple d'évaluation de la base d'apprentissage. Ce score a été ensuite intégré dans le système de reconnaissance, aux côtés du modèle statistique de langue. Cette nouvelle information a permis d'améliorer le taux de reconnaissance

en mots d'un système de transcription grand vocabulaire de nouvelles radiophoniques, ce qui confirme l'importance que peut avoir la prise en compte des contextes spécifiques dans une tâche de transcription de grande ampleur. L'intégration de ces approches théoriques au sein d'une plate-forme établie de la complexité de celle d'un système de transcription grand vocabulaire de la parole a soulevé inévitablement de nombreuses difficultés techniques d'implémentation, qui bien que n'apparaissant que très peu dans ce document, requièrent le développement d'approches élaborées pour notamment réduire la complexité des méthodes envisagées. Ce travail est d'ailleurs un élément incontournable des approches à base de mémoire, et constitue souvent également un frein au développement de celles-ci. Le système développé pour effectuer les expériences décrites dans ce mémoire a en particulier requis un effort très important pour concevoir et implémenter des structures de données et des heuristiques permettant de réduire suffisamment le temps de calcul pour envisager et réaliser ces expériences en un temps raisonnable.

Malgré des résultats prometteurs pour les deux modèles (non probabiliste et probabiliste) validés dans ce mémoire, l'approche suivie ne pourra être pleinement satisfaisante que lorsque plusieurs extensions de ces travaux auront été réalisés. Nous dégageons trois grands axes de développement de poursuite de travaux de recherche :

- Premièrement, l'utilisation de structures syntaxiques plus riches que celles provenant de la métrique de Levenshtein. Cette piste de recherche est décrite en détail au dernier chapitre de ce mémoire. Cette extension ouvre également des possibilités d'intégration de connaissances syntaxiques dans les systèmes de reconnaissance, et constitue donc une voie de recherche potentiellement très intéressante.

- Deuxièmement, la prise en compte de la sémantique dans le modèle de langue. En effet, à l'oral la syntaxe est beaucoup moins contrainte que pour l'écrit, et la sémantique joue donc un rôle relatif plus important. Ainsi, les notions de similarité développées dans ce mémoire doivent intégrer *a minima* une composante de sémantique lexicale, voire, en fonction du développement des méthodes actuelles de sémantique distributionnelle de la phrase, une composante de sémantique structurée.

- Troisièmement, une meilleure prise en compte du coût de modélisation de nouvelles informations par le modèle de langue. En effet, on ne peut intégrer efficacement d'informations plus riches que celles capturées par le modèle de langue n-gramme que s'il y a suffisamment de données dans le corpus pour correctement estimer les statistiques idoines. Une première réponse est développée dans le dernier chapitre de ce mémoire par l'introduction de modèles spécialisés, qui peuvent être aussi bien de nature syntaxique que sémantique et qui permettent de ne modéliser une information supplémentaire que pour un sous-ensemble du vocabulaire pour lequel elle est pertinente et pour lequel son rapport coût de modélisation / gain en efficacité

est positif. Ce nouveau champ de recherche a beaucoup de potentiel pour les modèles de langue stochastiques et restera pertinent quelque soit la taille des corpus d'apprentissage du fait de la combinatoire exponentielle de la langue.

Les modèles de langue sont de plus en plus utilisés quotidiennement dans de nombreux domaines : reconnaissance de la parole, traduction automatique, numérisation de l'écrit, indexation d'Internet, systèmes de recommandation, synthèse vocale, etc. Il ne se passe vraisemblablement pas une journée sans que vous utilisiez, consciemment ou non, les résultats de modèles de langue. De plus, les modèles de langues sont au cœur de la thématique *Big Data* qui constitue à l'heure actuelle un des principaux enjeux des développement de l'Internet actuel.

C'est pourquoi, malgré la difficulté de développement lié à la complexité de l'environnement d'un système de reconnaissance de la parole et la taille très importante des corpus qui demande des structures de données appropriées ; malgré le nombre important de travaux de recherche sur le domaine qui n'ont pas « détrôné » l'état de l'art toujours constitué par les modèles n-grammes lissés par la technique de Kneser-Ney (illustré par exemple par le pessimisme de Goodman dans son état de l'art [31]) et donc malgré le caractère exploratoire (et *a priori* non rentable), il est essentiel de poursuivre les recherches sur les modèles de langue.

Nous avons donc démontré qu'il était possible d'améliorer les performances des modèles de langue en introduisant un modèle probabiliste inspiré des modèles à base d'exemple. En effet, ces modèles sont complémentaires aux modèles n-grammes et ainsi peuvent être interpolés pour obtenir un modèle de langue plus efficace. Nous avons enfin démontré dans ce mémoire l'intérêt d'une modélisation différenciée de l'historique suivant les mots du vocabulaire en introduisant des modèles spécialisés, ce qui ouvre des perspectives encourageantes pour les recherches futures sur les modèles de langue.

Bibliographie

[1] A. Abeillé, L. Clément, and F. Toussenel. Building a treebank for french. *Treebanks : building and using parsed corpora*, pages 165–188, 2003.

[2] L. Allison, CS Wallace, and CN Yee. Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution*, 35(1) :77–89, 1992.

[3] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial intelligence review*, 11(1) :11–73, 1997.

[4] L.R. Bahl, F. Jelinek, and R.L. Mercer. A maximum likelihood approach to continuous speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 4(2) :179–190, 1983.

[5] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1) :164–171, 1970.

[6] J.R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8) :1279–1296, 2000.

[7] P.F. Brown, V.J.D. Pietra, R.L. Mercer, S.A.D. Pietra, and J.C. Lai. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1) :31–40, 1992.

[8] A. Brun, C. Cerisara, D. Fohr, I. Illina, D. Langlois, O. Mella, K. Smaïli, et al. Ants le système de transcription automatique du loria. In *Workshop ESTER, Avignon, France*, 2005.

[9] H. Bunt, P. Merlo, and J. Nivre. *Trends in parsing technology. Dependency parsing, domain adaptation and deep parsing*. Springer Verlag, 2010.

[10] Frédéric Béchet. Lia_tagg : a statistical pos tagger + syntactic bracketer. `http://www.lia.univ-avignon.fr/chercheurs/bechet/download_fred.html`.

[11] C. Cerisara, C. Gardent, C. Anderson, et al. Building and exploiting a dependency treebank for french radio broadcasts. In *Proc. Intl Workshop on Treebanks and Linguistic Theories (TLT), Tartu, Estonia*, 2010.

[12] C. Cerisara, C. Gardent, et al. The jsafran platform for semi-automatic speech processing. *Interspeech 2011*, 2011.

[13] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Morgan Kaufmann Publishers Inc., 2000.

[14] E. Charniak. Immediate-head parsing for language models. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 39, pages 116–123, 2001.

[15] C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, et al. Structure and performance of a dependency language model. In *In Proceedings of Eurospeech*, 1997.

[16] C. Chelba and F. Jelinek. Structured language modeling for speech recognition. *Arxiv preprint cs/0001023*, 2000.

[17] S.F. Chen. *Building Probabilistic Models for Natural*. PhD thesis, Harvard University Cambridge, Massachusetts, 1996.

[18] S.F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[19] K.W. Church and W.A. Gale. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech & Language*, 5(1) :19–54, 1991.

[20] M.J. Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 184–191. Association for Computational Linguistics, 1996.

[21] Antoine de Saint Exupery. *Terre des Hommes*. Gallimard, 1939.

[22] S. Dennis. A Memory-Based Theory of Verbal Cognition. *Cognitive Science*, 29(2) :145–193, 2005.

[23] D. Fohr, O. Mella, C. Cerisara, and I. Illina. The automatic news transcription system : Ants, some real time experiments. In *Eighth International Conference on Spoken Language Processing*, 2004.

[24] S. Furui, K. Iwano, C. Hori, T. Shinozaki, Y. Saito, and S. Tamura. Ubiquitous speech processing. In *icassp*, pages 13–16. IEEE, 2001.

[25] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. *GNU scientific library*. Network Theory, 2002.

[26] W.A. Gale and G. Sampson. Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3) :217–237, 1995.

[27] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Proc. Interspeech*, volume 5, pages 1149–1152, 2005.

[28] S. Galliano, G. Gravier, and L. Chaubard. The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[29] E. Gerbino and M. Danieli. Managing dialogue in a continuous speech understanding system. In *Third European Conference on Speech Communication and Technology*, 1993.

[30] I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4) :237, 1953.

[31] J.T. Goodman. A bit of progress in language modeling extended version. 2001.

[32] S. Huet, P. Sébillot, and G. Gravier. Utilisation de la linguistique en reconnaissance de la parole : un état de l'art. *Arxiv preprint cs/0605147*, 2006.

[33] S. Issar and W. Ward. Cmlps robust spoken language understanding system. In *Third European Conference on Speech Communication and Technology*, 1993.

[34] F. Jelinek. Interpolated estimation of Markov source parameters from sparse data. *Pattern recognition in practice*, pages 381–397, 1980.

[35] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1998.

[36] D. Jurafsky, J.H. Martin, and A. Kehler. *Speech and language processing : an introduction to natural language processing, computational linguistics and speech recognition*, volume 2. MIT Press, 2002.

[37] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(3) :400–401, 1987.

[38] D. Klein. *The unsupervised learning of natural language structure*. PhD thesis, Citeseer, 2005.

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

[40] A. Lee, T. Kawahara, and K. Shikano. Julius—an open source real-time large vocabulary recognition engine. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[41] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[42] C.D. Manning. Probabilistic syntax. *Probabilistic linguistics*, pages 289–341, 2003.

[43] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english : The penn treebank. *Computational linguistics*, 19(2) :313–330, 1993.

[44] J.J. McCall. Induction : From kolmogorov and solomonoff to de finetti and back to kolmogorov. *Metroeconomica*, 55(2-3) :195–218, 2004.

[45] R. Mitkov. Anaphora resolution : The state of the art. *Unpublished Manuscript*, 1999.

[46] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3) :443–453, 1970.

[47] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8(1) :1–38, 1994.

[48] J. Nivre, J. Hall, and J. Nilsson. Maltparser : A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219, 2006.

[49] Peter Norvig. On Chomsky and the two cultures of statistical learning. `http://norvig.com/chomsky.html`, 2011.

[50] B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2) :249–276, 2001.

[51] L. Romary, S. Salmon-Alt, and G. Francopoulo. Standards going concrete : from lmf to morphalou. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 22–28. Association for Computational Linguistics, 2004.

[52] R. Rosenfeld. *Adaptive statistical language modeling : a maximum entropy approach*. PhD thesis, Carnegie Mellon University, 1992.

[53] S. Salmon-Alt, E. Bick, L. Romary, and J.M. Pierrel. La freebank : vers une base libre de corpus annotés. In *Traitement Automatique des Langues Naturelles-TALN'04*, 2004.

[54] D. Sankoff and J.B. Kruskal. Time warps, string edits, and macromolecules : the theory and practice of sequence comparison. *Reading : Addison-Wesley Publication, 1983, edited by Sankoff, David ; Kruskal, Joseph B.*, 1, 1983.

[55] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49. Manchester, UK, 1994.

[56] H. Schmid. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*. Citeseer, 1995.

[57] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proc. International Conference on New Methods in Language Processing*, pages 44–49, 1994.

[58] P.H. Sellers. An algorithm for the distance between two finite sequences. *J. Comb. Theory, Ser. A*, 16(2) :253–258, 1974.

[59] Z. Solan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33) :11629, 2005.

[60] K. Toutanova, D. Klein, C.D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

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

[62] W. Ward et al. The cmu air travel information service : Understanding spontaneous speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 127–129, 1990.

[63] J. Wu and S. Khudanpur. Combining nonlocal, syntactic and n-gram dependencies in language modeling. In *Sixth European Conference on Speech Communication and Technology*, 1999.

Résumé

Un modèle de langue stochastique a pour rôle de donner la meilleure estimation possible de la probabilité d'une suite de mots dans une langue donnée. C'est un composant essentiel des logiciels de reconnaissance de la parole duquel dépend grandement leurs performances. Les modèles état-de-l'art les plus utilisés sont les modèles n-grammes lissés par la technique de Kneser-Ney. Ces modèles utilisent les statistiques d'occurrence de séquences de mots d'une longueur maximale typiquement de 5, statistiques calculés sur un vaste corpus d'apprentissage.

Cette thèse commence par une étude empirique des erreurs d'un système de reconnaissance de la parole état-de-l'art en français. Il est apparu que de nombreux phénomènes langagiers réguliers sont hors de portée des modèles n-grammes. Cette thèse explore donc une approche duale du paradigme statistique dominant en utilisant des modèles à base de mémoire qui ont traitent efficacement les phénomènes spécifiques en synergie avec les modèles n-grammes qui modèlisent efficacement les grandes tendances.

La notion de similarité entre longs n-grammes est étudiée, de façon à identifier les contextes particuliers à prendre en compte dans un premier modèle de langue de similarité. Les informations ainsi extraites du corpus sont combinés via un noyau Gaussien pour calculer un nouveau score. L'intégration de ce modèle non probabiliste, a permis d'améliorer les performances d'un système de reconnaissance. Un deuxième modèle est proposé, probabiliste et permettant ainsi une meilleur intégration de l'approche par similarité avec les modèles existants et qui améliore les performances en perplexité sur du texte.

Mots-clés: modèle de langue, similarité, modèle stochastique, reconnaissance de la parole, théorie de l'édition des séquences, modèle de mémoire, n-gramme

Abstract

The role of a stochastic language model is to give the best estimation possible of the probability of the sequence of words in a given language. It is an essential component of any speech recognition software and has a great influence on performance. The state-of-the-art models most commonly used are the n-gram models smoothed using the Kneser-Ney technique. These models use occurrence statistics of word sequences typically up to a length of 5, statistics computed on a large training corpus.

This thesis starts by an empirical study of the errors of a state-of-the-art speech recognition system in French, which shows that there are many regular language phenomena that are out of reach of the n-gram models. This thesis thus explores a dual approach of the prevailing statistical paradigm by using memory models which process efficiently specific phenomena, in synergy with the n-gram models which efficiently main trends.

The notion of similarity between long n-gram is studied in order to identify the relevant contexts to take into account in a first similarity language model. The data extracted out of the corpus is combined via a Gaussian kernel to compute a new score. The integration of this non-probabilistic model improves the performance of a recognition system. A second model is then introduced, probabilistic and thus allowing a better integration of the similarity approach with the existing models and improves the performance in perplexity on text.

Keywords: language model, similarity, stochastical model, speech recognition, string edit theory, memory model, n-gram

6

Annexe - étiquettes syntaxiques de Morphalou

Le lexique Morphalou[51] décrit pour chaque entrée toutes les étiquettes syntaxiques connues lui étant applicables. Toutefois il n'est pas exhaustif ne contenant aucun nom propre ou abbréviation, ni toutes les formes fléchies possibles d'un mot. Les étiquettes possibles pour la catégorie grammaticale (*grammaticalCategory*) de chaque mot sont :

- nom commun (*commonNoun*)
- verbe (*verb*)
- adjectif (*adjective*)
- adverbe (*adverb*)
- mots fonctionnels (*functionalWord*)
- interjection (*interjection*)
- onomatopées (*onomatopea*)

Puis pour chacune de ces catégories, des informations syntaxiques plus précises peuvent être renseignées telles que :

- genre (masculin, féminin) (*grammaticalGender*)
- nombre (singulier, pluriel) (*grammaticalNumber*)
- temps (imparfait, présent, futur, etc) (*grammaticalTense*)
- mode (indicatif, conditionnel, etc) (*grammaticalMode*)
- etc