Production de paraphrases pour les systèmes vocaux humain-machine

Jonathan Chevelu

Université de Caen
Basse-Normandie

UNIVERSITÉ de CAEN BASSE-NORMANDIE

U.F.R. DE SCIENCES

ÉCOLE DOCTORALE

STRUCTURE, INFORMATION, MATIÈRE ET MATÉRIAUX

THÈSE

présentée par

JONATHAN CHEVELU

et soutenue

le 17 MARS 2011

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE CAEN

Spécialité : informatique et applications

Arrêté du 7 août 2006

PRODUCTION DE PARAPHRASES POUR LES SYSTÈMES VOCAUX HUMAIN-MACHINE

MEMBRES DU JURY

M. Arnaud LALLOUET, professeur, université de Caen Basse-Normandie
M. Marc DYMETMAN, docteur d'État, Xerox Research Center Europe *(rapporteur)*
M. Olivier BOËFFARD, professeur, ENSSAT *(rapporteur)*
M. Aurélien MAX, maître de conférences, université Paris XI
M. Thierry MOUDENC, docteur, Orange Labs
M. Yves LEPAGE, professeur, université du Caen Basse-Normandie et université de Waseda *(directeur de thèse)*

Million-to-one chances crop up nine times out of ten.
— Granny Weatherwax

Dans *Equal Rites* de Terry PRATCHETT, 1987.

Vous dites pas : « qu'est ce qu'il fait chaud »,
vous dites : « la chaleur est un plat qui se mange froid ».
— Karadoc

Dans *Kaamelott*, Livre IV, L'Échelle de Perceval,
écrit par Alexandre ASTIER, 2005.

REMERCIMENTS

J'ai l'habitude de dire que la majorité des bonnes choses qui m'arrivent dans ma carrière sont dues à la chance. C'est en partie vrai, mais elles sont tout autant dues aux personnes admirables que j'ai rencontrées et qui m'ont soutenu. Je souhaite donc profiter de cet espace pour les remercier.

Je remercie avant tout Yves Lepage d'avoir bien voulu m'encadrer malgré l'éloignement géographique et la paperasse administrative pour monter une thèse CIFRE. Plus sérieusement, il m'a énormément appris sur la rigueur scientifique et sur le traitement automatique des langues, un domaine que je ne connaissais pas avant ma thèse. Nos échanges m'ont toujours été très profitables et m'ont guidé dans mes recherches. Même si je n'acquerrais probablement jamais les qualités rédactionnelles espérées, j'ai beaucoup appris sous sa direction et je le remercie grandement du temps passé à m'encadrer.

Je tiens aussi à remercier chaleureusement mes encadrants d'Orange Labs : Thierry Moudenc et Franck Panaget. Même s'ils n'ont pas toujours pu me suivre à plein temps, ils ont toujours été là aux bons moments. Je ne les remercierai jamais assez de m'avoir permis de réaliser cette thèse dans de si bonnes conditions. Je les remercie aussi pour ce sujet qui, même s'il m'a parfois fait pester au début, a fini par réellement me passionner.

Je remercie particulièrement Olivier Boëffard. Il m'a fait découvrir le domaine de la synthèse vocale et il m'a également fait le plaisir d'être rapporteur de ma thèse. Il fait partie de ces personnes m'ayant offert les opportunités qui m'ont fait avancer dans ma carrière.

Je souhaite aussi remercier mon second rapporteur, Marc Dymetman, ainsi qu'Aurélien Max. Nos différentes rencontres à l'autre bout de la planète m'ont toujours fait extrêmement plaisir aussi bien sur le plan scientifique et technique, qu'humain. Probablement sans le savoir, ils ont eu une influence non négligeable sur mes travaux de thèse et je les remercie de faire maintenant partie de mon jury. Un grand merci également à Arnaud Lallouet d'avoir bien voulu présider ce jury.

Plus généralement, je veux remercier ceux qui m'ont fait découvrir et apprécier le métier de chercheur. Je pense en particulier à Luc Bougé et Claude Jard, ainsi que l'ensemble des personnes de l'antenne de Bretagne de l'ENS Cachan qui m'ont offert l'opportunité de m'épanouir dans un métier formidable. Je pense aussi à l'ensemble des gens de l'ENSSAT, comme Nelly Barbot, Arnaus Delhay, Laurent Miclet et tous les autres.

J'aimerais remercier l'ensemble des personnes d'Orange Labs avec qui j'ai vécu cette thèse. Merci à toute l'équipe Voice, l'équipe NADIA ou ADN et en particulier à Philippe Bretier. Je ne suis toujours pas certain que nos longues conversations sur le sens en ait eu un mais elles étaient passionnantes. Les longues discussions avec toute la bande du midi, en particulier sur les feuilles CAP, me manqueront. À tous merci. Je remercie aussi tous les membres du Greyc et en particulier de l'équipe Island, j'aurais aimé passer plus de temps en leur compagnie.

Je souhaite bien évidemment remercier mes parents, sans qui je ne serais pas là. Merci à ma sœur et à tous mes amis. Ils sont pour moi un soutien inestimable.

Pour finir, je tiens à remercier particulièrement trois personnes qui m'ont marqué, ainsi que les travaux présentés dans cette thèse. Je regrette de m'éloigner de deux d'entre eux. Tout d'abord, Thomas Lavergne, c'est grâce à sa passion du go que j'ai pu progresser mais aussi avoir l'idée d'utiliser un algorithme de jeu de go pour la production de paraphrases. Ghislain Putois, qui a toujours été un formidable contradicteur mais aussi une aide précieuse et un véritable ami. Enfin, je remercie Honorine du plus profond de mon cœur. Elle a tellement fait pour moi et pour cette thèse que je ne peux que la lui dédier.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

Système :

SVI Serveur Vocal Interactif

Domaine de recherche :

TAL Traitement Automatique des Langues

Mesures :

SMO Score d'Opinion Moyenne

TEC Taux d'Erreur en Caractères

Algorithmes et programmes :

ASM Alignement Séquentiel Multiple

EM Estimation-Maximisation

RAMC Recherche dans un Arbre par échantillonnage de Monte- Carlo

RAVE *Estimation rapide des valeurs des actions*[1]

GPMC Générateur de Paraphrases par échantillonnage de Monte-Carlo

UCB *borne supérieure de confiance*[2]

UCT *borne supérieure de confiance appliquée aux arbres*[3]

SVM Séparateur à Vaste Marge

1. *Rapid Action Value Estimate*
2. *Upper Confidence Bound*
3. *Upper Confidence bound for Tree search*

INTRODUCTION

Les systèmes vocaux humain-machine sont de plus en plus présents autour de nous, que ce soit pour informer les voyageurs dans les gares ou pour négocier un rendez-vous avec un technicien. Pour échanger des informations avec une machine, ils offrent une interface naturelle pour l'homme, n'occupant pas les mains et ne nécessitant pas de formation. Ils permettent aussi de réduire les coûts de tâches systématiques et d'associer une identité vocale à un service. Mais y a-t-il un lien entre ce qui est dit et le système qui va le dire ?

Prenons un exemple légèrement caricatural. Faire prononcer « Salut mes supers potes ! » avec la voix de synthèse de Jacques Chirac sera probablement humoristique. *A contrario*, si l'objectif n'est pas comique, alors ce choix de formulation est au mieux perturbant, au pire il peut dégrader l'image qu'a l'auditeur du système. De plus, la synthèse risque de produire des erreurs acoustiques car ces mots – et leur enchaînement – ont peu de chance d'avoir été observés lors de l'apprentissage de la voix. Il en va de même si l'on cherche à faire prononcer « Mes chers confrères, je vous adresse mes salutations. » avec la voix de synthèse de Homer Simpson. Alors que l'on peut considérer que ces deux messages *disent presque la même chose*, ils ne sont pas interchangeables dans certains contextes d'utilisation.

L'approche classique en synthèse vocale consiste à vouloir créer des systèmes capables de tout prononcer dans toutes les situations. Par défaut, le système est spécialisé pour un sous-langage ou se limite à certains cadres applicatifs. Le problème est que plus ce sous-langage est vaste et plus il faut de données lors de l'apprentissage pour obtenir une voix correcte. De plus, certaines caractéristiques ne peuvent pas être appréhendées par la synthèse vocale. Par exemple c'est le cas de la cohérence stylistique entre différents messages. L'objectif de cette thèse et de renverser la perspective. Plutôt que de vouloir un système capable de tout prononcer, nous nous intéressons au problème inverse : adapter le message au système et à son contexte d'utilisation.

En fait, ce qui compte généralement dans un message, c'est le sens qu'il doit faire passer et non les mots utilisés. Il serait donc possible de choisir une nouvelle forme pour un message si celle-ci améliore certaines caractéristiques *a priori* secondaires comme la qualité acoustique. Proposer une forme alternative sans changer le sens, c'est produire une paraphrase.

Cette thèse traite donc de la production automatique de paraphrases pour les systèmes vocaux humain-machine. Mais pour pouvoir utiliser des paraphrases dans de tels systèmes, il faut d'abord être capable de les produire et de les évaluer. C'est pourquoi nous nous intéressons principalement à ces deux prérequis tout en visant une utilisation pour un système vocal.

Notre démarche s'articule en trois étapes : définir, étudier et proposer.

La partie I vise à définir les problématiques et les objets que nous traiterons dans cette thèse. Ainsi, le chapitre 1 décrit le fonctionnement des systèmes vocaux humain-machine et plus particulièrement la synthèse vocale par concaténation d'unités acoustiques. Il traite du lien entre le message émis et la qualité des systèmes

vocaux. Il définit donc notre contexte applicatif de travail et les problèmes que l'on peut espérer résoudre. Le chapitre 2 présente une réflexion autour de la paraphrase et les usages que l'on peut en faire en général. Il délimite notre cadre d'usage des paraphrases pour l'amélioration des systèmes vocaux humain-machine. Enfin, il complète la définition des paraphrases par un tour d'horizon des différentes questions qui se posent lorsque l'on aborde le problème de la production automatique de paraphrases.

La partie II vise à étudier les points forts et les limites d'une approche particulière de la production de paraphrases : la production statistique de paraphrases. Elle utilise les analyses réalisées dans la partie I pour mettre au point un cadre expérimental d'évaluation qui permet l'analyse des résultats d'un outil au niveau de l'état de l'art. Le chapitre 3 présente une étude des problématiques liées à l'évaluation des paraphrases. Il décrit la réalisation d'outils et de corpus nécessaires pour étudier et comprendre un système de production de paraphrases lors de son évaluation. Le chapitre 4 vise à construire un générateur de paraphrases à partir d'éléments de l'état de l'art. L'étude de cette réalisation doit permettre de comprendre en détails les mécanismes internes d'un tel système. Enfin, le chapitre 5 rend compte de l'évaluation du système de référence mis au point dans le chapitre 4 grâce aux outils du chapitre 3. L'ensemble des expériences menées permet de préciser les limites de la production statistique de paraphrases.

Enfin la partie III présente une proposition pour pallier les lacunes rencontrées dans la partie II. Le chapitre 6 remet en question la vision classique de la paraphrase ; il propose un cadre différent pour l'évaluation des systèmes et la production de paraphrases. Le chapitre 7 présente un nouvel algorithme pour la production de paraphrases ; il produit des paraphrases comparables à celles de l'état de l'art mais il s'affranchit des contraintes du modèle de la production statistique de paraphrases. Grâce à cet algorithme, il est désormais possible de modifier librement le modèle de la production statistique de paraphrases. Cet algorithme est évalué dans le chapitre 8. En fonction des résultats obtenus plusieurs améliorations sont proposées et elles aussi évaluées.

Nous conclurons en rappelant nos principales contributions et esquisserons des perspectives sur la production des paraphrases et leurs utilisations dans des systèmes vocaux humain-machine.

Première partie

DÉFINIR : SYSTÈMES VOCAUX HUMAIN-MACHINE ET PARAPHRASES

1 LES SYSTÈMES VOCAUX HUMAIN-MACHINE

Les systèmes vocaux humain-machine sont des systèmes automatisés d'interaction avec l'humain via une interface vocale. Ils prennent de plus en plus de place dans le quotidien car ils permettent une interaction rapide qui ne nécessite pas de formation. De plus, ils restent peu coûteux à mettre en place.

L'interaction avec l'utilisateur peut se restreindre à l'émission d'un message vocal par un système de synthèse vocale. Nous présenterons la synthèse vocale dans la section 1.1. L'interaction peut aussi être plus complexe et être constituée de plusieurs tours de parole pour former un dialogue. On a alors affaire à des serveurs vocaux interactifs que nous présenterons dans la section 1.2.

Un point commun de ces systèmes est qu'ils vocalisent un message en direction de l'utilisateur. Nous verrons dans la section 1.3 en quoi le choix de ce message peut influencer les performances globales des systèmes vocaux humain-machine.

1.1 SYNTHÈSE VOCALE

La synthèse vocale consiste à transformer un texte en signal de parole acoustique. La figure 1.1 présente les deux modules traditionnels composant les systèmes de synthèse vocale :

- le module de traitement linguistique. Il réalise une analyse grammaticale de la phrase ; la phonètise ; détermine la prosodie ; la durée et la fréquence fondamentale des composantes du signal ;
- le module de traitement acoustique. Il réalise la production du signal acoustique du message.

La synthèse du signal peut être réalisée par plusieurs techniques :

- la synthèse articulatoire [Rubin et coll., 1981] ;
- la synthèse par formants [Allen et coll., 1987] ;
- la synthèse de diphones [Moulines et Charpentier, 1990] ;
- la synthèse par sélection d'unités acoustiques [Hunt et Black, 1996] ;
- la synthèse par modèles de Markov cachés [Yoshimura et coll., 1999] ;
- la synthèse hybride [Pollet et Breen, 2008].

Pour nos travaux, nous utilisons le système de synthèse vocale BARATINOO développé par Orange et utilisé dans un grand nombre de systèmes industriels. BARATINOO est fondé sur une approche par sélection d'unités acoustiques.

1.1.1 *Synthèse par sélection d'unités acoustiques*

Nous allons maintenant préciser le fonctionnement de la synthèse par sélection d'unités acoustiques. Cette approche est largement utilisée dans l'industrie et permet d'avoir des voix expressives de qualité.

La synthèse par sélection d'unités s'appuie sur un corpus de parole enregistrée sur plusieurs heures. Ce corpus est annoté et segmenté en unités acoustiques. Cette

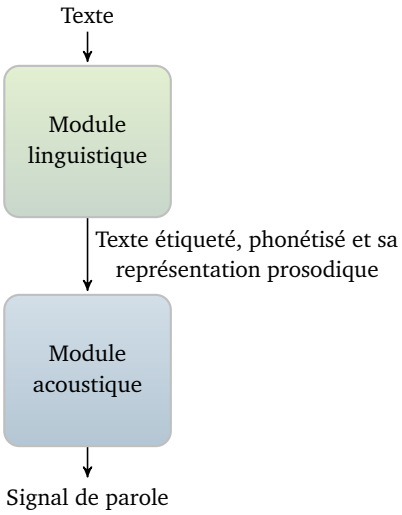


FIGURE 1.1: Architecture d'un système de synthèse vocale.

opération est souvent réalisée manuellement afin d'obtenir une meilleure qualité de voix. Les unités acoustiques peuvent être des phones, c'est-à-dire des réalisations acoustiques d'un phonème ; des diphones, c'est-à-dire des unités allant de la moitié d'un phone à la moitié du phone suivant ; des syllabes ; etc. Notons que le corpus de parole doit contenir plusieurs réalisations d'une même classe d'unités afin de permettre au système de sélectionner celle qui est la plus appropriée, en particulier pour pouvoir produire différentes prosodies d'un même mot.

Le module de traitement acoustique concatène les unités du corpus pour produire le signal le plus proche possible de la cible fournie par le module de traitement linguistique. En fait, cette cible définit un ordre partiel sur les unités acoustiques ce qui permet de construire un treillis de production. La sélection d'unités consiste à chercher le meilleur chemin dans ce treillis, relativement à une fonction de coûts. Typiquement, la recherche est réalisée grâce à un algorithme de type Viterbi [Viterbi, 1967].

La fonction des coûts de sélection comprend traditionnellement deux composantes :

- le coût cible. Celui-ci mesure l'adéquation entre une unité potentielle et la cible définie par le module de traitement linguistique. Cette mesure peut être calculée sur la fréquence fondamentale, la durée, le contexte phonétique de l'unité, etc.
- le coût de concaténation. Celui-ci mesure l'adéquation entre deux unités potentielles consécutives. Il permet d'évaluer la qualité de la concaténation de ces deux unités. Cette mesure peut être calculée sur la différence de fréquence fondamentale, d'énergie, de caractéristique spectrale, etc.

L'avantage de la synthèse par sélection d'unités est que la parole à l'intérieur d'une unité est naturelle, puisqu'elle provient d'un enregistrement. De même, si le système concatène deux unités qui se suivaient dans le corpus d'origine – par exemple pour produire un mot qui a été prononcé lors de l'enregistrement du

corpus – alors la parole produite est *a priori* de très bonne qualité. En revanche, les principales erreurs acoustiques risquent d'apparaître lors de la concaténation d'unités qui ne se suivaient pas dans le corpus. La taille du corpus de parole est donc un facteur primordial de la qualité de la synthèse. Mais le coût de constitution d'une voix dépend lui aussi de la taille du corpus : il faut un comédien pour l'enregistrer et rémunérer les personnes qui doivent l'annoter.

La constitution de corpus est donc un problème difficile. En général, les phrases qui constituent le corpus de parole proviennent d'un grand corpus textuel, comme les articles du journal *Le Monde* couvrant une année. Les phrases à enregistrer sont choisies afin de couvrir un ensemble de classes d'unités acoustiques, par exemple, tous les diphonèmes. Le problème consiste donc à obtenir le sous-corpus le plus petit possible, en terme de temps d'enregistrement, qui couvre au mieux les unités nécessaires à une bonne synthèse vocale. De nombreux travaux cherchent donc à optimiser le corpus acoustique en terme :

- d'unités à couvrir [Cadic et coll., 2009] ;
- de distribution des unités [Krul et coll., 2006] ;
- d'algorithme de sélection [François, 2002], domaine auquel nous avons contribué [Chevelu et coll., 2008].

1.1.2 *Outil d'aide à la conception de messages vocaux*

Généralement, on n'utilise pas de système de synthèse vocale entièrement automatique pour les applications où la qualité acoustique des messages est très importante. Une approche plus sûre consiste à pré-enregistrer la partie fixe des messages et à laisser un système automatique les concaténer à des parties variables. Ces parties de messages pré-enregistrées peuvent tout de même être produites grâce à un système de synthèse vocale, pour éviter de faire revenir le comédien pour chaque nouveau message. En revanche, elles sont contrôlées par le concepteur de message qui s'assure de leur qualité acoustique. Pour pouvoir corriger les éventuelles erreurs acoustiques des messages, on utilise un outil d'aide à la conception.

SPEECH ONLINE est le système d'aide à la conception d'Orange fondé sur le système de synthèse BARATINOO présenté à la section 1.1.1. Une capture d'écran de l'interface en ligne de SPEECH ONLINE est présentée à la figure 1.2.

Cet outil permet de proposer une phrase et d'obtenir le signal acoustique correspondant. Il est possible de corriger la phonétisation de la phrase ainsi que d'ajouter des pauses dans la prononciation.

La fonctionnalité principale est la possibilité de modifier la sélection réalisée par le synthétiseur. Ainsi, si la synthèse proposée n'est pas satisfaisante au niveau d'une unité, l'utilisateur n'a qu'à l'indiquer. Le programme relance alors l'algorithme de sélection en interdisant l'unité incriminée et propose une nouvelle vocalisation de la phrase. Par corrections successives, le concepteur peut obtenir un message correctement synthétisé.

1.2 SERVEURS VOCAUX INTERACTIFS

Comme nous l'avons dit, nous parlons de serveurs vocaux interactifs (SVI) lorsque l'interaction est constituée de plusieurs tours de parole. Les SVI entrent dans la

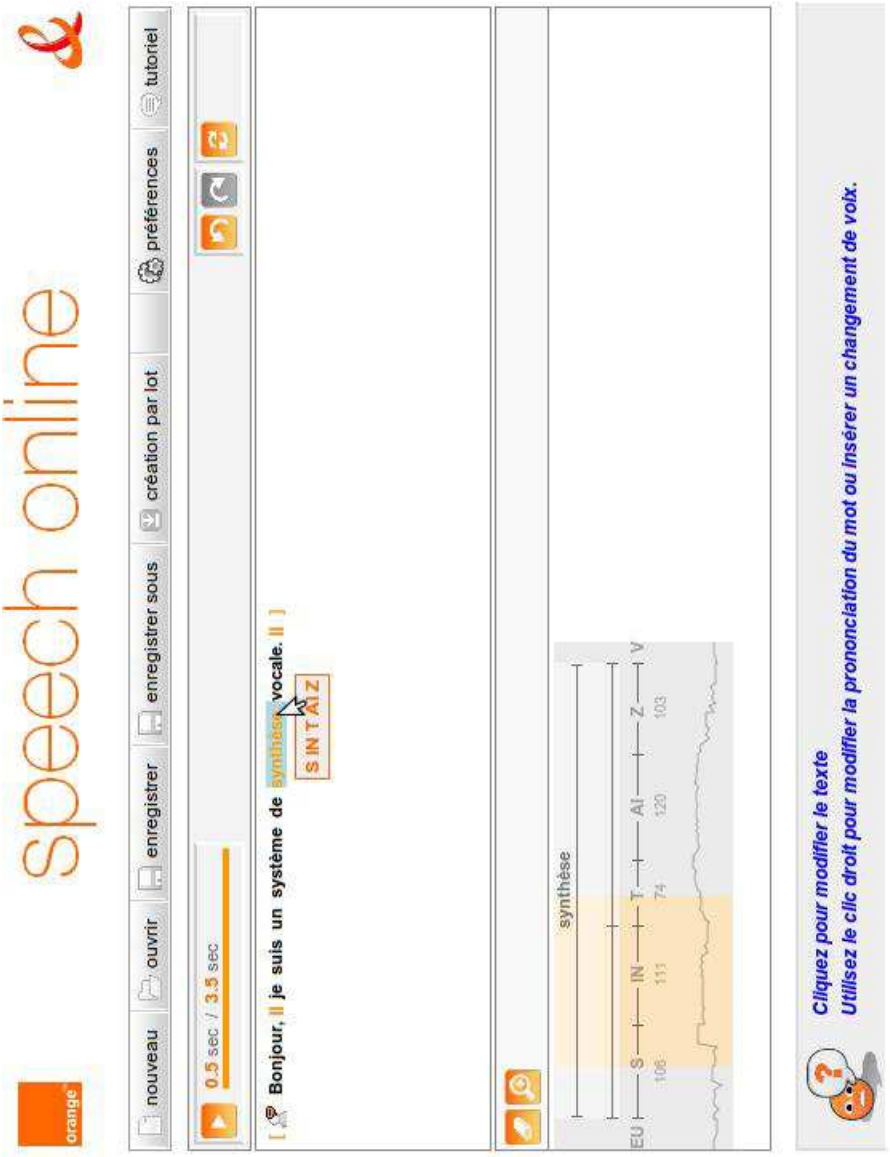


FIGURE 1.2: Capture d'écran de l'outil d'aide à la conception de messages vocaux SPEECH ONLINE. Dans cet exemple, l'utilisateur vérifie la phonétisation du mot «synthèse» qu'il pourra éventuellement modifier.

catégorie des systèmes de dialogue humain-machine. Ils permettent à des utilisateurs d'interagir avec un système d'information en utilisant la modalité orale – souvent par l'intermédiaire d'un téléphone.

Par exemple les SVI sont utilisés dans l'industrie pour :

- de l'orientation de trafic téléphonique. Ils tentent d'identifier le correspondant le plus approprié pour l'utilisateur ;
- de la prise automatique de rendez-vous ;
- de l'assistance automatique ;
- de la réservation automatique de ticket ;
- etc.

On peut identifier trois niveaux de complexité pour ces systèmes. Ceux-ci correspondent aux actions permises à l'utilisateur :

1. choix multiples au clavier. Le système énonce plusieurs choix à l'utilisateur. Celui-ci répond en utilisant un terminal – comme les touches de son téléphone. Cette approche est simple à mettre en œuvre et très fiable. En revanche, elle limite fortement l'interaction et peut perturber l'utilisateur. En effet, celui-ci doit mémoriser l'association choix-touches du terminal et il doit attendre l'énoncé du choix correspondant à ses souhaits avant de pouvoir poursuivre le dialogue ;
2. prononciation de mots clefs. L'utilisateur prononce un ou plusieurs mots de commande définis par le système pour indiquer son souhait. Cette approche est légèrement plus complexe. Son principal avantage par rapport au système à choix multiples est que l'utilisateur emploie lui aussi la parole pour communiquer ce qui est souvent plus naturel ;
3. énonciation en langue naturelle. Pour ces systèmes, l'utilisateur est libre de formuler son message. Cette approche, qui cherche à rendre l'interaction beaucoup plus naturelle, est aussi beaucoup plus difficile à mettre en œuvre.

Dans tous les cas, le retour du système est réalisé à l'oral.

Nous traiterons dans ce document des systèmes de dialogue vocaux en langue naturelle. Nos propos seront souvent valables pour les systèmes à choix multiples et les systèmes par mots clefs. La principale différence vient du fait que ces deux derniers peuvent faire abstraction du problème d'interprétation du message de l'utilisateur. Les systèmes à choix multiples évitent également le problème de la reconnaissance de la parole.

Traditionnellement, les SVI sont découpés en cinq composants. Cette architecture traditionnelle est présentée à la figure 1.3. Notons qu'en fonction du système de dialogue, il arrive que certains composants soient absents ou légèrement différents.

Nous allons maintenant préciser le rôle de chaque composant ainsi que l'approche retenue dans la solution industrielle d'Orange : DISSERTO.

La reconnaissance vocale :

L'objectif d'un module de reconnaissance vocale est de transcrire le signal de parole émis par l'utilisateur en texte. Il réalise l'opération inverse de la synthèse vocale.

Les approches utilisées de nos jours sont fondées sur des modèles statistiques appris sur des corpus de parole transcrite. Lors d'une première étape, le signal est

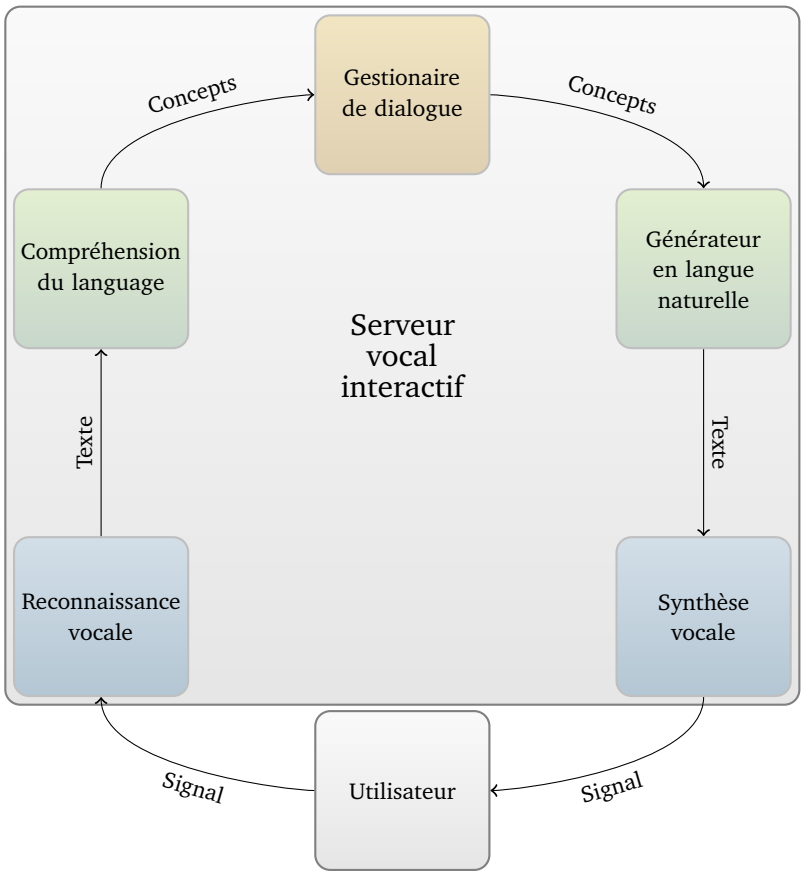


FIGURE 1.3: Schématisation du fonctionnement d'un SVI.

numérisé en vecteurs acoustiques. Ensuite, un algorithme de décodage est utilisé pour trouver la séquence de mots la plus probable en fonction d'un modèle de langue phonétisée et d'un modèle acoustique. Le modèle acoustique est appris à partir d'un bi-corpus alignant signal acoustique de parole et transcription textuelle.

Lorsque la situation le permet, le modèle de langue est souvent réduit à une grammaire construite manuellement. Par exemple, après une question fermée posée à l'utilisateur, le système peut attendre une réponse en « Oui », « Non » et une troisième catégorie pour le reste. Ce type d'optimisation permet d'améliorer grandement les performances de la reconnaissance vocale.

Le composant de compréhension du langage :

À partir du texte en sortie du module de reconnaissance vocale, le composant de compréhension du langage cherche à extraire la sémantique du message. Il en produit une transcription dans un langage symbolique qui peut être doté d'opérateurs logiques.

Dans DISSERTO, ce module est réalisé manuellement grâce à des connaissances expertes. Pour simplifier, un ensemble de règles associe des séquences de mots à

des concepts. Les performances de ce module dépendent donc de l'anticipation du champ lexical des messages de l'utilisateur.

Le gestionnaire de dialogue :

Au centre du serveur vocal interactif se trouve le gestionnaire de dialogue. C'est ce module qui gère l'historique, la stratégie de dialogue et les réponses que le système peut formuler. À partir de l'historique du dialogue et de la conceptualisation du message utilisateur, le gestionnaire de dialogue met à jour ses connaissances sur le dialogue et produit une réponse sous forme d'un ensemble de concepts. Ce module est souvent couplé à un système d'information qui peut être vu comme l'ensemble des connaissances auquel il a accès. Par exemple, pour une application de prise automatique de rendez-vous, le système d'information contiendra l'ensemble des plages horaires disponibles.

Dans l'approche industrielle DISSERTO, le gestionnaire de dialogue est un automate construit manuellement. Les états représentent les différents états possibles du dialogue. La transition de l'état courant à l'état suivant est réalisée en fonction du message utilisateur et des réponses du système d'information. L'interface de l'outil de conception sous la forme de logigramme de cet automate est présentée à la figure 1.4.

Le générateur en langue naturelle

À l'inverse du composant de compréhension du langage, le générateur en langue naturelle cherche à produire une phrase textuelle à partir d'un ensemble de concepts décrivant le message à transmettre à l'utilisateur.

Cette tâche très complexe n'est pas traitée dans le système DISSERTO. En effet, les messages sont définis par des textes à trous avant le déploiement d'un service. Ils sont introduits dans le système directement depuis le gestionnaire de dialogue. Ce dernier produit donc directement une phrase qu'il transmet à la synthèse vocale.

La synthèse vocale

Comme le présente la section 1.1, le rôle d'un synthétiseur vocal est de produire un signal acoustique de parole à partir d'un texte.

La suite DISSERTO utilise le système de synthèse BARATINOO. En fait, les messages ne sont pas produits automatiquement à la demande. Les parties des messages non variables sont produites avant le déploiement d'un service. Elles sont contrôlées et éventuellement corrigées grâce à l'outil d'aide à la conception SPEECH ONLINE qui a été présenté dans la section 1.1.2. Cette étape permet de garantir le niveau minimum de qualité acoustique imposé par le monde industriel.

1.3 PROBLÉMATIQUES LIÉES AU MESSAGE

Les systèmes de synthèse vocale, et plus généralement les systèmes vocaux humain-machine, soulèvent un certain nombre de questions liées au texte qu'ils vocalisent. Nous présentons dans cette section certaines des problématiques que nous avons identifiées.

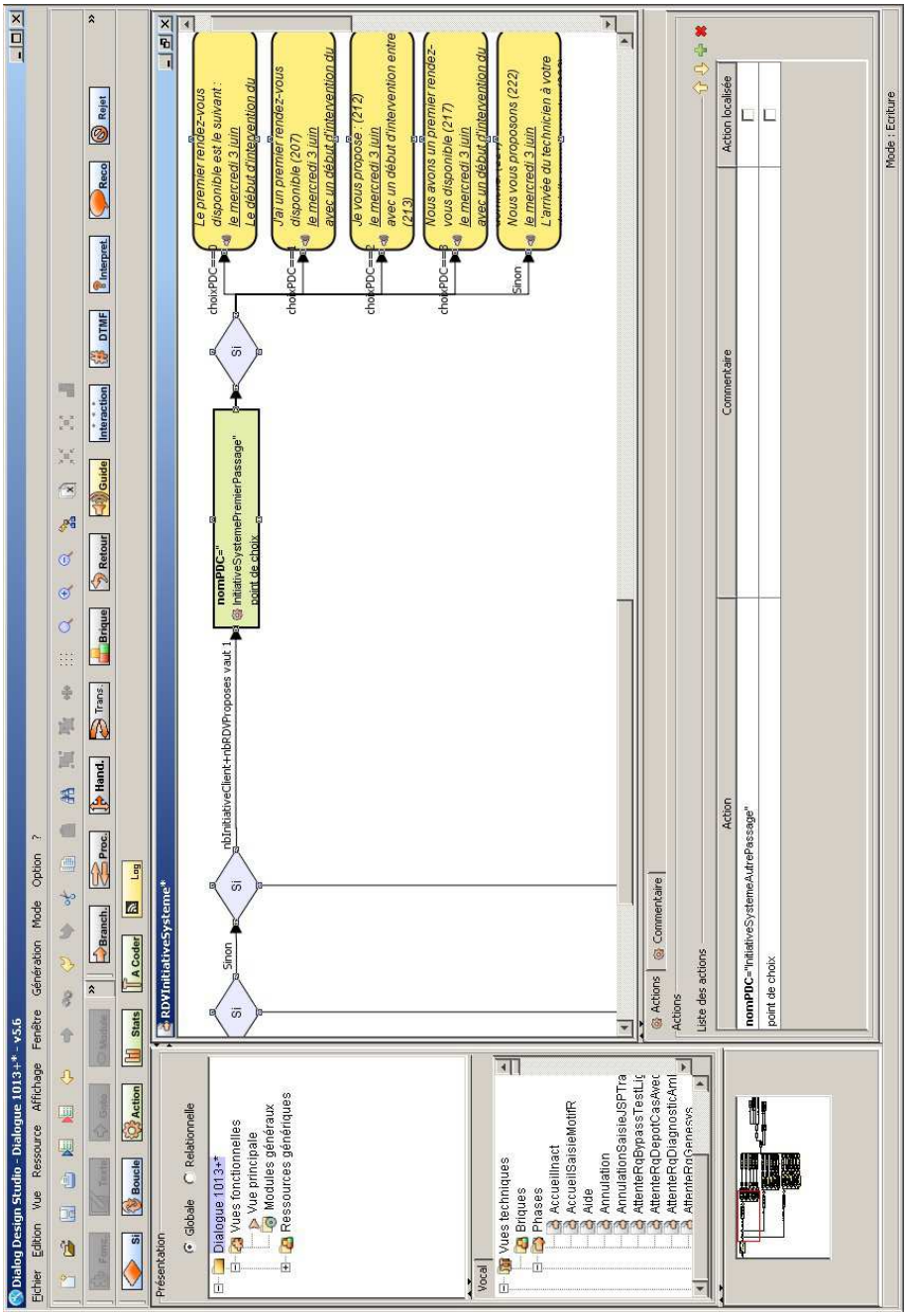


FIGURE 1.4: Capture d'écran de l'outil de conception de dialogue de DISSERTO.

1.3.1 *Problèmes liés à la synthèse vocale*

Comme l'a présenté la section 1.1.1, la taille du corpus de parole est un facteur important quant au coût de développement d'une voix de synthèse. De nombreux travaux cherchent à réduire les corpus de parole ou à mieux les construire [Chevelu et coll., 2008; Cadic et coll., 2009].

Notre hypothèse de départ est qu'en fait, la qualité de la synthèse vocale est dépendante de l'adéquation entre le corpus de synthèse et les phrases à synthétiser. En effet, pour le cas dégénéré où la phrase à synthétiser est entièrement présente dans le corpus, alors la qualité sera excellente. À l'inverse, faire prononcer à un système une phrase dans une langue différente de celle du corpus d'apprentissage entraînera un résultat de très mauvaise qualité. Ceci même en supposant que le module de traitement linguistique soit adapté. Voilà pourquoi beaucoup de travaux tentent de résoudre le problème suivant :

Problème 1.1 *Produire le corpus le plus en adéquation avec le sous-langage des phrases à synthétiser – et plus généralement à une langue dans son entier.*

Mais, il existe des circonstances où le corpus de synthèse n'est pas contrôlable. C'est le cas, par exemple, lorsque l'on souhaite construire une voix à partir d'enregistrements publics d'une personne. Dans ces circonstances, il n'est pas possible d'imposer le texte du locuteur. Si la quantité d'enregistrement est trop faible ou s'ils ne sont pas assez variés, alors le système risque de produire des messages de mauvaise qualité.

C'est pourquoi, nous nous demandons s'il n'est pas possible de résoudre le problème dual du problème 1.1 :

Problème 1.2 *Trouver les phrases les plus adaptées à un système de synthèse construit à partir d'un corpus de parole donné.*

Dans cette vision, les phrases à synthétiser ne sont plus vues comme des contraintes mais comme des variables.

À partir du moment où les phrase produites s'adaptent au système de synthèse, il serait théoriquement possible de réduire la taille du corpus tout en conservant la qualité acoustique en sortie du système. Le système serait aussi plus à même de travailler avec un corpus où le contenu phonétique n'est pas contrôlé.

Il est aussi possible d'essayer d'adapter la forme du message à synthétiser aux caractéristiques de la voix – le timbre, le débit, etc. – afin d'augmenter la « naturalité » de la voix produite. De même, si la voix est celle d'un personnage connu, il peut être souhaitable d'adapter le message au « style » du locuteur originel.

1.3.2 *Problèmes liés aux serveurs vocaux interactifs*

Les SVI partagent bien entendu les problématiques de la synthèse que nous venons de présenter. Mais l'enchaînement des interactions est aussi sujet à des problématiques spécifiques.

Tout d'abord, les systèmes de dialogue deviennent de plus en plus complexes. Certains SVI peuvent être constitués de plusieurs milliers de messages. Il est donc difficile pour un concepteur d'avoir une vision d'ensemble du service, *a fortiori*

lorsqu'il n'intervient que pour mettre à jour ou étendre le service. Parallèlement à cela, des sociétés mettant en place des SVI imposent une homogénéité entre les messages en terme de style et de vocabulaire. Il est donc de plus en plus difficile de satisfaire ce type de contraintes.

De plus, pour certaines langues comme le français, les répétitions dans des messages consécutifs peuvent être considérées comme des lourdeurs stylistiques. Or, la complexité des systèmes fait qu'il n'est pas toujours possible de connaître les messages précédents dans le dialogue.

Enfin, les interactions en dialogue sont soumises à un phénomène nommé « amorçage structurel ». Celui-ci traduit le fait que la syntaxe d'une interaction est influencée par la forme de l'interaction précédente [Pickering et Ferreira, 2008]. Il serait donc possible d'encourager la forme de la réponse de l'utilisateur. Comme nous l'avons vu dans la section 1.2, le module de reconnaissance vocale est limité par son corpus d'apprentissage et par le modèle de langue qui ont été utilisés pour son entraînement. Ceci est aussi vrai pour le module de compréhension des messages. Choisir une formulation où la réponse sera plus probablement en adéquation avec ces deux modules permettrait d'améliorer les performances d'un SVI.

Plus généralement se pose la question du meilleur message pour un dialogue – et aussi de la meilleure stratégie de dialogue. Par exemple, les travaux de Putois et coll. [2010] intègrent des alternatives à l'intérieur d'un SVI. Un module d'apprentissage par renforcement est utilisé pour sélectionner la meilleure alternative observée au fil des dialogues. La mise au point d'un générateur de paraphrases permettrait d'offrir automatiquement des alternatives à ce type de système.

1.4 CONCLUSION

Dans ce chapitre, nous avons présenté les systèmes vocaux et précisé leur fonctionnement dans l'industrie. Ces systèmes sont au cœur d'enjeux scientifiques par leur complexité et économiques par les services qu'ils peuvent rendre. L'amélioration de ces systèmes et de leur conception est donc un problème important.

Dans nos travaux, nous nous intéressons aux messages émis. Nous avons mis en évidence dans la section 1.3 leurs impacts possibles sur les systèmes vocaux humain-machine. Alors qu'ils sont souvent vus comme des constantes du système, nous proposons de les voir comme des variables.

D'un autre côté, le fait qu'un message doive être émis n'est pas anodin. Une part du message doit impérativement être conservée. Nous définissons cet élément à garder comme l'intention communicative, ou sens, du message.

En résumé, nous nous posons la question suivante : est-il possible de proposer des formulations alternatives, conservant le sens du message d'origine, mais prenant en compte un ensemble de critères calculables dans un système vocal humain-machine ?

2 LES PARAPHRASES

Comme nous l'avons vu dans le chapitre précédent, l'élément important d'un message à vocaliser est son sens. En revanche, si cela ne change pas le sens d'origine, il peut être souhaitable de modifier la forme de surface du message – c'est-à-dire les mots choisis ou l'ordre de ces mots – pour résoudre un des problèmes que nous avons identifiés, comme la qualité acoustique du message ou la cohérence intra-message du vocabulaire. En d'autres mots, nous souhaiterions produire une paraphrase du message et que celle-ci améliore certains critères non liés au sens.

Dans ce chapitre, nous traiterons plus en détail des paraphrases dans le domaine du traitement automatique des langues (TAL). Dans la section 2.1 nous proposons de regarder ce que sont les paraphrases et à quoi elles sont utilisées. La section 2.2 décrit notre cadre de production de paraphrases pour les systèmes vocaux humain-machine et détaille certains travaux réalisés dans ce domaine. La section 2.3 présente plus généralement les principales problématiques autour de la production de paraphrases et décrit certaines pistes explorées dans des travaux antérieurs.

2.1 DÉFINITIONS ET USAGES

Avant de pouvoir utiliser des paraphrases pour améliorer des systèmes de synthèse vocale, il faut se demander ce que sont ces objets et à quoi ils peuvent servir. La section présente vise à répondre à ces deux interrogations. Nous présenterons d'abord une réflexion sur la définition des paraphrases et sur la production de paraphrases dans la section 2.1.1. Puis, dans la section 2.1.2, nous verrons les tâches du TAL où elles sont utilisées, ou pour lesquelles la communauté souhaiterait les utiliser.

2.1.1 *La paraphrase*

Des définitions variées de la paraphrase existent dans le domaine de la production automatique de paraphrases.

Barzilay et McKeown [2001]; Fujita et Inui [2005]; Bannard et Callison-Burch [2005] définissent « *les paraphrases [comme] des moyens alternatifs de transmettre la même information* » ou « *le même sens* » pour Zhao et coll. [2009]. Pour Quirk et coll. [2004], une paraphrase est « *une séquence de mots distincte [. . .] « signifiant » la même chose* ». Sekine [2005] la définit ainsi : « *un ensemble de phrases qui exprime la même chose ou le même évènement* ». Toutes ces définitions insistent sur une relation de conservation, qu'elle soit du sens, de l'information, du message ou de l'intention communicative, entre la phrase d'origine et la paraphrase produite.

D'un point de vue étymologique, le mot « paraphrase » dérive du latin *paraphrasis* emprunté au grec παράφρασις. Il est composé de *para*, « à côté de », et de *phrasis*, « discours » [Académie française, 2006].

Le dictionnaire TLFi [site internet] propose la définition suivante comme sens premier de *paraphrase* :

Définition 2.1 *Paraphrase [TLFi, site internet] (tiret A) : « Souvent avec une connotation péjorative. Développement explicatif d'un texte, souvent verbeux et diffus, qui ne fait qu'en délayer le contenu sans que rien ne soit ajouté au sens ou à la valeur. »*

Une paraphrase est souvent vue au sens péjoratif comme une *dilution* du sens dans une phrase plus longue. La définition 2.2 du domaine de la linguistique n'ajoute pas cette connotation.

Définition 2.2 *Paraphrase [TLFi, site internet] (en linguistique) : « Opération de reformulation aboutissant à un énoncé contenant le même signifié (ou encore ayant une même structure profonde), mais dont le signifiant est différent, notamment plus long (autrement dit, dont la structure de surface est différente). »*

Ces définitions reposent sur la notion de sens d'une phrase qui est difficile à définir.

D'autre travaux définissent la paraphrase à partir de la notion plus « restreinte » d'inférence sémantique [1] [Androutsopoulos et Malakasiotis, 2010]. La définition suivante reprend celle donnée par Dagan et coll. [2006] :

Définition 2.3 *L'inférence sémantique est définie comme une relation orientée entre des couples d'expressions textuelles, notées T pour le « texte » inférant et H pour « l'hypothèse » inférée. On dit que T infère H si le sens de H peut être déduit du sens de T, au sens communément admis.* [2]

Cette relation définit donc un ordre partiel sur le sens entre phrases : une phrase peut permettre d'en inférer une seconde mais l'inverse n'est pas nécessairement vrai. C'est pourquoi la relation de paraphrase est parfois définie comme la composante connexe de cette relation d'ordre [Androutsopoulos et Malakasiotis, 2010] :

Définition 2.4 *Si une phrase permet d'inférer une seconde et que celle-ci permet aussi d'inférer la première, alors ces deux phrases sont paraphrases l'une de l'autre.*

Certains travaux considèrent donc que la relation de paraphrase est une relation d'équivalence alors que d'autres la considèrent plutôt comme notion de ressemblance sémantique. Ces derniers arguent qu'il n'existe pas d'égalité stricte de sens [Eco, 2007]. Il nous semble dangereux de rendre transitive la relation de paraphrase ce qui devrait être le cas si on considère que c'est une relation d'équivalence. C'est pour cela que nous parlerons de ressemblance sémantique entre une phrase et une paraphrase plutôt que d'équivalence ou d'égalité. Ceci laisse déjà entrevoir des difficultés pour déterminer si le sens de la phrase d'origine ressemble suffisamment à celui d'une phrase candidate pour considérer cette dernière comme une paraphrase.

Dans nos travaux, nous nous intéressons plus à la production de paraphrases qu'à la paraphrase en elle-même. L'observation des précédentes définitions nous conduit à proposer la définition suivante pour les générateurs de paraphrases :

1. L'inférence sémantique est appelée *entailment* en anglais.
2. [*entailment is defined as a directional relationship between pairs of text expressions, denoted by T - the entailing "Text", and H - the entailed "Hypothesis". We say that T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people.*]

Définition 2.5 *Un générateur de paraphrases est un outil qui, à partir d'une phrase, produit au moins une phrase différente qui a un sens ressemblant à celui de la phrase d'origine.*

Cette définition nous semble contenir les quatre termes clefs de la production de paraphrases :

- phrase : un générateur de paraphrases produira des phrases à partir de phrases ;
- différente : comme nous le montrons dans la section suivante, si l'on souhaite produire une paraphrase c'est que la phrase d'origine ne convient pas complètement ;
- sens : bien que difficile à définir, cette relation porte sur la notion de « sens ». Notons que celui-ci peut être dépendant du contexte de la phrase ;
- ressemblant : la relation de paraphrase définit une relation de ressemblance sémantique et non pas d'équivalence. Cette relation entre phrase est, pour nous, réflexive puisqu'une phrase à le même sens qu'elle-même. Cette relation est aussi symétrique, c'est-à-dire que si une phrase est une paraphrase d'une autre, alors l'inverse est aussi vrai. En revanche, pour nous, cette relation n'est pas transitive. Enfin, une phrase et une de ses paraphrases n'ont pas forcément exactement le même sens mais ceux-ci sont considérés comme « suffisamment ressemblants ».

2.1.2 *Usages des paraphrases*

Les paraphrases ont déjà attiré l'attention de la communauté du TAL. Nous proposons ici un rapide tour d'horizon des applications qui ont été envisagées pour les paraphrases. Nous distinguons deux types d'applications pour les paraphrases : celles reposant sur la détection de paraphrases dans un corpus et celles fondées sur la production de paraphrases.

D'après la littérature, un système de détection de paraphrases pourrait permettre, par exemple, de détecter les plagiats [White et Joy, 2004; Uzuner et coll., 2005]. Il est aussi possible d'envisager une utilisation de cette relation de ressemblance sémantique pour faire du regroupement de documents. En fait, la plupart du temps, l'acquisition de paraphrases est présentée comme un premier pas dans la conception des systèmes de production de paraphrases fondés sur l'apprentissage [Shinyama et Sekine, 2003; Brockett et Dolan, 2005; Bouamor et coll., 2007]. Dans ce cas, c'est en fait, non plus une application, mais une problématique préalable.

La production de paraphrases vise différents types d'applications que nous classons en trois catégories.

Le premier groupe d'applications vise à produire des paraphrases pour remplacer la phrase de départ. Ainsi, en produisant une paraphrase plus courte que la phrase d'origine, il est possible de réduire la longueur d'un texte, c'est une première approche du résumé de texte [Knight et Marcu, 2000]. Les paraphrases peuvent aussi servir à normaliser un texte pour que les phrases qui le composent utilisent un vocabulaire contrôlé. On rencontre ce type de contrainte lors de l'élaboration de manuels procéduraux en aéronautique par exemple [Nasr, 1996]. À l'inverse, il peut être difficile d'éviter des répétitions lors de l'écriture d'un texte. Un système de

production de paraphrases peut aider à la rédaction en proposant des alternatives à un rédacteur [Max, 2008].

L'autre grand groupe d'applications pour la production de paraphrases consiste à améliorer les systèmes de traitement automatique de texte. En effet, un grand nombre de ces outils repose sur la recherche de schémas dans un texte ou une phrase. Un générateur de paraphrases permettrait alors de normaliser ou de proposer plusieurs alternatives du texte en entrée du système pour améliorer l'espace couvert et donc les performances de l'outil de TAL. Parmi les applications de TAL fréquemment citées qui pourraient intégrer un générateur de paraphrases, nous trouvons les systèmes de question-réponse [Duclaye et coll., 2003], les systèmes d'extraction d'information [Sekine, 2005] ou encore les systèmes de traduction automatique [Callison-Burch et coll., 2006].

Une dernière application, plus anecdotique, consiste à produire des paraphrases pour aider l'évaluation d'outils de TAL. Par exemple, en traduction automatique, les mesures d'évaluation automatique comparent les solutions proposées à une ou plusieurs traductions de référence. Il a été montré que plus le nombre de références est important et plus la mesure obtenue est fiable et proche d'une évaluation humaine [Papineni et coll., 2002]. Or, les corpus d'évaluation avec un grand nombre de références sont rares et complexes à construire. Un système de production automatique de paraphrases permettrait éventuellement d'augmenter simplement le nombre de références disponibles [Lepage et Denoual, 2005; Kauchak et Barzilay, 2006; Zhou et coll., 2006].

2.2 PRODUCTION DE PARAPHRASES POUR LES SYSTÈMES VOCAUX

Cette thèse vise à étudier le problème de la production de paraphrases dans le but d'améliorer les systèmes vocaux humain-machine. Il est donc nécessaire de définir précisément notre problématique de production. La section 2.2.1 présente le cadre applicatif dans lequel se placent nos travaux. Dans le section 2.2.2 nous présenterons les travaux qui se sont aussi intéressés à la paraphrase dans le contexte de la synthèse vocale.

2.2.1 *Cadre applicatif*

Avant d'envisager l'utilisation de paraphrases dans les systèmes vocaux – comme ceux présentés dans le chapitre 1, il est nécessaire de circonscrire le cadre applicatif que nous souhaitons traiter. L'utilisation d'un outil de production de paraphrases vise à répondre aux problématiques décrites dans la section 1.3. L'amélioration visée, qu'elle soit l'amélioration de la qualité acoustique ou de la cohérence syntaxique des messages, sera appelée la « tâche » du problème de production.

Dans le cadre de nos travaux, nous envisageons deux types de systèmes intégrant un générateur de paraphrases :

– un système entièrement automatisé. Ici, le générateur paraphrase la phrase d'entrée. Le résultat est transmis directement au moteur de synthèse vocale. Cette paraphrase est construite de façon à être la plus adaptée possible à la tâche tout en conservant le sens de la phrase d'origine. La figure 2.1 présente un schéma de cette application ;

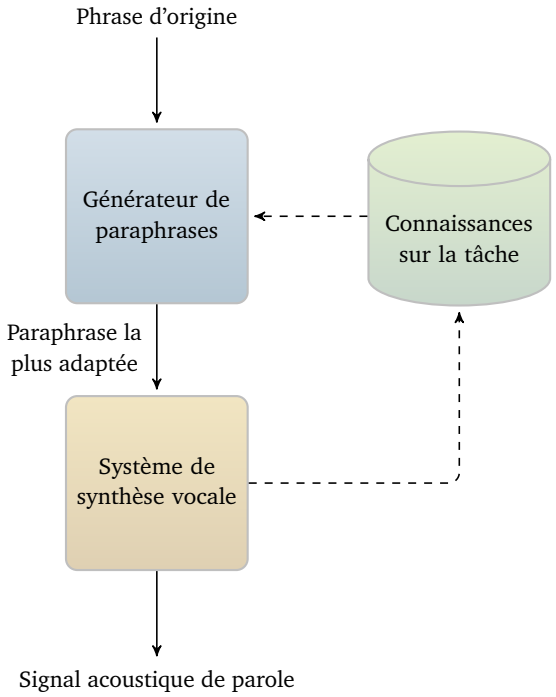


FIGURE 2.1: Application de la production de paraphrases dans un système vocal automatique.

– un outil d'aide à la conception de messages vocaux. Ici, un utilisateur propose une phrase au système. Ce dernier retourne une ou plusieurs paraphrases plus adaptées à la tâche. L'utilisateur peut alors corriger une paraphrase et la soumettre à nouveau au système ou valider une des propositions pour produire le message vocal. La figure 2.2 présente un schéma de cette application.

Notons qu'une application entièrement automatique soulève plus de difficultés qu'un outil d'aide à la conception. En effet, pour être utilisé dans un contexte commercial, un tel système doit être extrêmement fiable : les paraphrases proposées ne doivent pas comporter d'erreur syntaxique ou grammaticale et le sens de la phrase d'origine doit être bien préservé. À l'inverse, la qualité des paraphrases pour un système d'aide est moins critique puisqu'un utilisateur contrôle et corrige éventuellement les paraphrases proposées. En revanche, nous sommes conscients qu'il faut que le concepteur s'approprie et prenne en compte les propositions d'un tel système. Bien qu'important, nous ne traiterons pas, dans ce document, de ce problème d'acceptation par le concepteur et nous nous focaliserons sur le problème de la production des paraphrases.

Les paragraphes suivants présentent quelques choix, contraintes et observations qui nous permettront de mieux délimiter notre problématique de production de paraphrases. Nous traiterons du générateur de paraphrases, du format de son entrée et du type de modification qu'il introduit.

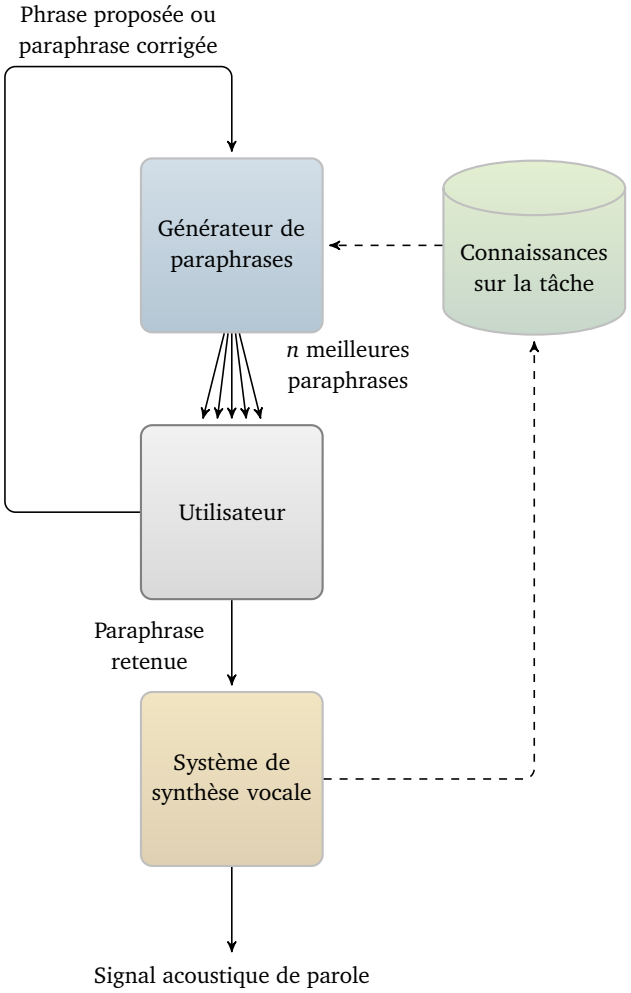


FIGURE 2.2: Application de la production de paraphrases pour l'aide à la conception de systèmes vocaux. Un utilisateur propose des messages pour le système et corrige les éventuels défauts des paraphrases proposées.

Générateur de paraphrases

Les différentes thématiques des textes à produire par un système vocal peuvent être très variées : du message humoristique pour les voix de personnages célèbres à un message technique pour les systèmes de dialogue d'aide à l'installation de matériel de télécommunication. De ce fait, nous ne souhaitons pas un imposer *a priori* un sous-langage pour les phrases d'entrée. Afin de simplifier l'adaptation du générateur de paraphrases au type d'application, l'utilisation d'une méthode par apprentissage nous semble préférable. Nous traiterons donc principalement des méthodes de production de paraphrases par apprentissage.

Format de l'entrée

Premièrement, dans notre cadre applicatif, l'entrée du système est constituée d'une phrase textuelle. Nous aurions pu choisir de partir d'une représentation sémantique – comme celle proposée par la théorie sens-texte par exemple [Mel'čuk, 1998]. En effet, pour certains systèmes de dialogue humain-machine, c'est ce type de représentation qui est transmis par le gestionnaire de dialogue au générateur en langue naturelle, comme expliqué dans la section 1.2. Dans nos travaux, nous souhaitons conserver une entrée sous forme textuelle. En effet, dans le système de dialogue industriel DISSERTO, présenté dans la section 1.1, dans lequel se place nos travaux, les messages sont justement conçus manuellement. De plus, dans un outil d'aide à la conception de messages vocaux comme SPEECH ONLINE, présenté dans la section 1.1.2, l'utilisateur manipule un texte et non une représentation sémantique.

Deuxièmement, dans notre étude, le texte d'entrée est limité à une phrase. En effet, les messages vocaux en dialogue ou en synthèse sont souvent constitués d'une seule phrase. De plus, il existe peu de travaux sur la reformulation d'objets plus grands – comme des paragraphes – si ce n'est pour fusionner plusieurs phrases ou découper une phrase en plusieurs phrases [Jing et McKeown, 2000]. Notons que pour ce type d'opérations, il est incorrect de parler de paraphrase puisqu'elle se restreint à une phrase par définition (voir la section 2.1.1). Il serait probablement plus correct de parler de *reformulation*, comme extension de la relation de paraphrase à un grain supérieur. De même, nous ne chercherons pas à prendre en compte le paragraphe qui pourrait contenir la phrase à traiter ou la partie de dialogue précédente, en particulier parce que cette information n'est pas disponible dans un outil d'aide à la conception de messages vocaux. Plus généralement, en dehors des informations liées à la tâche – qualité de la synthèse vocale, homogénéité entre les messages, etc. –, nous supposerons que nous ne disposons pas d'information de contexte.

Modifications introduites par le générateur de paraphrases

Dans notre problème de production de paraphrases, les mots à modifier ne sont pas définis *a priori*. L'outil de production est donc libre de modifier tout ou partie de la phrase d'origine pour améliorer le système global. Notons qu'il existe d'autres problèmes pour la paraphrase où la zone à modifier est imposée, au moins partiellement. C'est le cas dans des applications comme l'aide à l'écriture [Max, 2008] ou dans certains travaux pour l'aide à la traduction automatique [Bannard et Callison-Burch, 2005] par exemple. Dans le cas des systèmes vocaux humain-machine, il est aussi possible d'envisager un outil d'aide à la conception où l'utilisateur précise les mots à

modifier. Nous ne retenons pas ce cadre, bien que plus simple, car il est incompatible avec une application entièrement automatique. À l'inverse, même sans information sur les mots à modifier, les paraphrases peuvent être utilisées dans un outil d'aide à la conception de messages vocaux.

Enfin, il est à noter que la phrase d'origine peut être plus adaptée au contexte applicatif que l'ensemble des paraphrases que le générateur est en mesure de proposer. Malgré tout, afin de pouvoir étudier l'apport d'un outil de production de paraphrases, nous excluons la phrase d'origine des solutions possibles pour nos travaux d'études. Bien évidemment, dans une application industrielle, cette limitation n'est pas pertinente. Cette interdiction se retrouve dans la grande majorité des travaux sur la production de paraphrases, comme dans Barzilay et McKeown [2001]; Bannard et Callison-Burch [2005] ou Zhao et coll. [2009] par exemple.

2.2.2 *Paraphrases pour systèmes vocaux*

Peu de travaux s'intéressent à l'utilisation des paraphrases pour améliorer les systèmes vocaux. Cette section détaille trois travaux traitant de cette thématique.

D'un texte « écrit » vers un texte « oral »

À notre connaissance, Kaji et coll. [2004] est l'un des premiers travaux s'intéressant aux paraphrases dans un contexte de synthèse vocale. Ils cherchent à adapter un texte, en japonais, à la modalité vocale. Les auteurs constatent qu'il existe des expressions utilisées à l'écrit qui ne conviennent pas à l'oral. Ils proposent donc d'utiliser un générateur de paraphrases pour améliorer la « naturalité » d'un texte à synthétiser. Le système de Kaji et coll. [2004] est décomposable en trois étapes :

1. constitution d'un corpus de texte écrit et un corpus de texte « oral » en classant des documents d'un corpus de pages internet. Le système utilise une méthode de classification par règles, construites manuellement, qui reposent sur l'absence ou la présence d'expressions « interpersonnelles » c'est-à-dire des formes d'expressions de politesse ou honorifiques facilement détectables en japonais ;

2. extraction des paraphrases par une méthode automatique. Les paraphrases considérées ici sont des couples de segments sous-phrastiques et non pas des couples de phrases entières. La méthode utilisée est fondée sur une extraction de paraphrases depuis un dictionnaire [Kaji et coll., 2002] ;

3. conservation, parmi l'ensemble des paraphrases disponibles, de celles qui associent une expression « inadaptée à l'oral » à une expression « adaptée à l'oral ». Ce filtrage est réalisé à partir d'un classifieur supervisé – ici un séparateur à vaste marge (SVM) [Vapnik, 1998] – s'appuyant sur les distributions des mots des paraphrases dans les deux corpus de la première étape.

Les performances de l'outil de classification sont bonnes mais le système repose sur une heuristique propre à une langue, le japonais, pour l'extraction des paraphrases. Enfin, ces travaux n'expliquent pas comment utiliser ces paraphrases et ne mesurent pas l'impact de ces paraphrases sur la « naturalité » acoustique des textes synthétisés.

Choisir la meilleure paraphrase pour un système de synthèse vocale

À l'inverse, Nakatsu et White [2006] supposent que des paraphrases sont disponibles. Ces paraphrases sont obtenues à l'aide d'un outil de production en langue naturelle intégré dans un système de dialogue en langue naturelle (voir le chapitre 1). Les différences entre paraphrases correspondent à la substitution de quelques mots par des synonymes ou par des variations dans l'ordre des mots. Dans Nakatsu et White [2006], un ordonnanceur supervisé, ici, SVM, a pour mission de classer les paraphrases d'une même réalisation sémantique en fonction de leur qualité acoustique après synthèse. Les traits utilisés lors de l'apprentissage sont les coûts du synthétiseur vocal ou des modèles de langue appris sur le corpus de synthèse.

L'expérience réalisée dans Nakatsu et White [2006] montre qu'un classifieur est en mesure de choisir avec une bonne fiabilité les paraphrases jugées acoustiquement meilleures – dans 77,3 % des cas. Mais, lorsque l'on compare la qualité acoustique perçue par des auditeurs, les différences entre le système sélectionnant automatiquement la meilleure paraphrase et un système aléatoire ne sont pas statistiquement significatives. Ces travaux ce sont heurtés au manque de variabilité des paraphrases disponibles.

Production statistique de paraphrases et synthèse vocale

Cahill et coll. [2009] ont été les premiers à coupler un système de production de paraphrases avec un système de synthèse de la parole. Le système de production utilise des outils provenant de la traduction statistique, sur laquelle nous nous étendrons au chapitre 4. En l'absence des corpus d'apprentissage, les auteurs construisent manuellement un petit corpus de paraphrases de 250 paires de phrases. À partir d'une phrase source, le système produit plusieurs paraphrases et conserve celle ayant le coût de synthèse le plus faible. Les auteurs observent une diminution du coût de synthèse moyen sur un petit corpus d'évaluation de 30 phrases. En revanche, ils ne mesurent pas si l'amélioration en terme de qualité acoustique est perçue lors d'évaluations humaines. De plus, ils ne vérifient pas si le sens de la phrase est bien conservé en sortie du générateur.

2.3 PROBLÉMATIQUES LIÉES AUX PARAPHRASES

Dans les sections précédentes, nous avons explicité ce qu'est une paraphrase, situé notre problématique applicative et présenté quelques travaux s'en approchant. Nous allons maintenant élargir notre tour d'horizon pour traiter de la production automatique de paraphrases en général. Nous identifions trois problématiques principales :

- la section 2.3.1 expliquera comment produire des paraphrases;
- la section 2.3.2 expliquera comment utiliser les paraphrases pour accomplir une tâche ;
- la section 2.3.3 expliquera comment évaluer la qualité des paraphrases.

Nous détaillerons et critiquerons quelques travaux significatifs traitant de ces problèmes.

2.3.1 *Production*

La production de paraphrases est un problème ardu : il consiste à produire du texte naturel alors que la grammaire d'une langue est difficile à formaliser ; ce texte doit conserver le sens de la phrase d'origine alors que la notion de sens pose aussi des problèmes de définition. De nombreux travaux cherchent à résoudre ce problème grâce à différents paradigmes.

Suite à l'observation de plusieurs travaux, nous proposons une taxinomie des approches en quatre types de ressources et trois types de production. En effet, nous avons constaté que les travaux sur la production de paraphrases associent généralement un type de production avec un type de ressources particulier. En fait, les méthodes de production sont souvent très dépendantes du type de ressources retenu.

Dans notre taxonomie, les ressources peuvent être du type : bases de connaissances linguistiques, bi-corpus monolingue aligné, bi-corpus monolingue comparable et bi-corpus bilingue aligné. Les trois types de production sont : la production par schémas de transformation ; la production grâce à une représentation sémantique intermédiaire ; la production par des méthodes venant du domaine de la traduction automatique. Nous présentons ces différents types dans les paragraphes suivants.

Type de ressources : les bases de connaissances linguistiques

Ce type de ressources regroupe les bases de règles construites manuellement [McKeown, 1983] et les dictionnaires utilisés pour l'apprentissage des systèmes. Ces derniers peuvent être des dictionnaires généralistes [Kaji et coll., 2002; Fujita et coll., 2005], ou plus spécifiques comme des dictionnaires de structures lexicales conceptuelles [Fujita et coll., 2005]. Les réseaux sémantiques comme WordNet [Miller, 1995], utilisé par Hassan et coll. [2007] entrent aussi dans cette catégorie. Ces ressources sont généralement de très grande qualité mais demandent un travail de construction important, ce qui les rend rares et limitées, en particulier pour certaines langues.

Type de ressources : les bi-corpus monolingues alignés

Ces corpus sont constitués de couples de paraphrases qui servent d'exemples pour les outils d'apprentissage. Ils sont relativement rares et difficiles à construire. Comme nous l'avons signalé dans la section 2.1.2, l'acquisition de tels corpus fait l'objet de recherches importantes dans de nombreux travaux.

Type de ressources : les bi-corpus monolingues comparables

Ces corpus sont formés de deux ensembles de textes, dans la même langue, où la présence de paraphrases entre les deux corpus est quasi-certaine bien qu'elles ne soient pas spécifiquement identifiées. Il peut s'agir d'articles de journaux traitant des mêmes évènements, comme le proposent Barzilay et Lee [2003]. Ce type de ressources peut être facile à construire automatiquement pour certains domaines. En revanche, l'absence d'informations précises sur l'emplacement des paraphrases rend l'apprentissage automatique plus complexe ;

Type de ressources : les bi-corpus bilingues alignés

Ces corpus sont constitués d'un texte et de sa traduction. Les deux textes sont alignés phrase à phrase. Son utilisation pour la paraphrase vient du fait que l'opération de traduction est censée conserver le sens de la phrase d'origine. Ce type de ressources peut être simple à obtenir en grande quantité. En particulier, il constitue l'élément de base des systèmes de traduction statistiques.

Type de production : par schémas de transformation

Ces systèmes fonctionnent sur le principe suivant : le système dispose d'une base de schémas de transformation. Pour produire une paraphrase, le système sélectionne puis applique le schéma le plus adapté. Les schémas peuvent être des grammaires de transformations [McKeown, 1983], des règles de substitution [Hassan et coll., 2007; Max, 2008] ou des treillis de production [Barzilay et Lee, 2003]. Ces schémas sont l'équivalent de règles plus ou moins complexes, censés conserver le sens des phrases sur lesquelles ils sont appliqués.

Le choix du schéma à appliquer pour produire une paraphrase est réalisé à l'aide d'une mesure [Kaji et coll., 2002; Barzilay et Lee, 2003] ou d'un classifieur [Hassan et coll., 2007]. Ces outils doivent définir si le schéma est adapté à la phrase à transformer.

Par exemple, Barzilay et Lee [2003] apprennent les schémas à partir de groupes de phrases « similaires ». Des phrases sont dites « similaires » si elles traitent d'évènement similaire – du bilan humain d'un attentat dans l'exemple donné par les auteurs – et si elles ont la même structure – dans ces travaux, en utilisant une mesure fondée sur le nombre de mots et de séquences de mots en commun. À partir d'un groupe de phrases similaires, un algorithme d'alignement est utilisé pour construire un treillis – dans ces travaux, AMS [3] [Durbin et coll., 1998]. Les treillis sont ensuite généralisés. Pour ce faire, seuls les nœuds présents dans au moins 50 % des phrases utilisées pour construire le treillis sont conservés. Les nœuds supprimés sont remplacés par des emplacements libres et correspondent donc aux variabilités permises par le schéma. Les schémas sont ensuite appariés. Deux schémas sont considérés comme paraphrases lorsque les valeurs des emplacements libres sont similaires dans l'ensemble du corpus d'apprentissage. Ainsi, le corpus d'apprentissage doit non seulement contenir des phrases « similaires » mais aussi des paraphrases. Pour une phrase donnée, la production de paraphrases consiste à chercher le schéma lui correspondant le mieux, au sens d'AMS, et à utiliser le treillis « paraphrase » associé pour produire une paraphrase.

Ces approches n'utilisent qu'un seul schéma à la fois pour produire une paraphrase. Celui-ci peut couvrir une très grande partie de la phrase [Lin et Pantel, 2001; Barzilay et Lee, 2003], mais dans ce cas il faut un nombre exponentiel de schémas pour pouvoir traiter une grande variété de phrases ; ils peuvent aussi n'affecter que quelques mots [Max, 2008; Zhao et coll., 2008b], mais dans ce cas le type de transformation est très restreint.

Type de production : par représentation sémantique intermédiaire

Ce type de production fonctionne en deux temps :

3. Alignement Séquentiel Multiple

1. réalisation d'une analyse sémantique de la phrase d'origine pour produire une représentation sémantique ;

2. production d'une forme de surface, différente de la phrase d'origine, à partir de cette représentation sémantique.

Une étape intermédiaire peut être ajoutée pour modifier la représentation sémantique, en passant d'une voix active à la voix passive par exemple [Fujita et coll., 2005]. Il existe un grand nombre de représentations possibles comme les structures lexicales conceptuelles [Fujita et coll., 2005], la théorie sens-texte [Nasr, 1996], etc.

Cette décomposition reproduit un processus qui semble naturel de compréhension et de restitution du message. D'un autre côté, chacun de ces problèmes est très complexe et nécessite souvent des ressources linguistiques expertes difficiles à obtenir. Notons que la problématique de compréhension et de production de texte en langue naturelle se retrouve dans le modèle du dialogue humain-machine que nous avons présenté dans la section 1.2.

Type de production : par des méthodes venant du domaine de la traduction automatique

Pour ce type de production, la production de paraphrases est considérée comme un problème de traduction automatique monolingue : la langue d'origine est la même que celle dans laquelle on souhaite traduire. Ces travaux utilisent donc des outils provenant du domaine de la traduction bilingue comme des systèmes par analogie [Lepage et Denoual, 2005] ou des traducteurs statistiques [Dolan et Brockett, 2005; Zhao et coll., 2009]. La principale adaptation consiste souvent à obtenir des corpus d'apprentissage monolingues à la place des bilingues. Cette approche a l'avantage de bénéficier des importantes avancées en traduction automatique. La production statistique de paraphrases, version monolingue de la traduction statistique, semble prometteuse.

Synthèse des types de production et de ressources

Le tableau 2.1 classe certains travaux sur la production de paraphrases dans notre taxonomie. De plus en plus de travaux sur la paraphrase utilisent des corpus bilingues, en grande partie à cause de leur disponibilité. Enfin, suite aux importantes avancées en traduction automatique, les travaux sur les paraphrases se tournent de plus en plus vers ce paradigme.

2.3.2 *Utilisation*

Comme le montre la section 2.1.2, la production de paraphrases n'est jamais utilisée de façon isolée et est toujours associée à une tâche. Il faut donc être capable d'utiliser un générateur de paraphrases pour améliorer la tâche. Nous identifions trois stratégies dans la littérature : le pré-filtrage ; la post-sélection ; l'optimisation conjointe.

Le pré-filtrage consiste à conserver uniquement les connaissances du générateur de paraphrases qui peuvent être bénéfiques pour la tâche. Par exemple, pour la compression de texte, Zhao et coll. [2009] ne conservent dans leur table de pa-

		Type de production		
		Schémas de transformations	Représentation sémantique	Traduction automatique
Type de ressources	Bases de connaissances linguistiques	[McKeown, 1983] ; [Kaji et coll., 2002] ; [Hassan et coll., 2007]	[Nasr, 1996] ; [Fujita et coll., 2005] ; [Power et Scott, 2005]	
	Bi-corpus monolingue aligné	[Lin et Pantel, 2001]		[Quirk et coll., 2004] ; [Lepage et Denoual, 2005]
	Bi-corpus monolingue comparable	[Barzilay et Lee, 2003]		
	Bi-corpus bilingue aligné	[Max, 2008] ;[Zhao et coll., 2008b]		[Bannard et Callison-Burch, 2005] ; [Zhao et coll., 2009] ; [Max, 2009]

TABLEAU 2.1: Synthèse des travaux sur la production de paraphrases en fonction du type de production utilisé et des ressources utilisées.

raphrases[4] que les entrées où le résultat de la transformation est plus court, en nombre de caractères. Le pré-filtrage a l'avantage d'être simple à mettre en œuvre ; il ne nécessite généralement pas de modification des algorithmes de production. En revanche, il faut que la mesure de la tâche soit décomposable localement. De plus, cette approche ne tient pas compte des possibles synergies entre transformations : il peut être éventuellement avantageux d'utiliser une transformation qui agrandit la longueur de la phrase si celle-ci autorise par la suite des transformations plus efficaces globalement.

L'approche duale consiste à sélectionner *a posteriori* la meilleure paraphrase, au sens de la tâche, parmi une liste de candidates produites par le générateur. Cette sélection se fait traditionnellement en réordonnant les sorties du générateur afin de combiner les scores provenant du générateur à ceux de la tâche [Nakatsu et White, 2006; Cahill et coll., 2009]. Par contre, cette solution ne permet pas au générateur de prendre en compte la tâche, puisque cette dernière est faite après production des paraphrases. Ainsi, il se peut que les paraphrases proposées par le générateur soient toutes inadaptées.

La dernière solution consiste à optimiser conjointement la tâche et le modèle de production de paraphrases [Callison-Burch et coll., 2006; Onishi et coll., 2010]. Cette approche ne souffre pas des inconvénients des deux solutions précédentes.

4. Une table de paraphrases contient l'ensemble des transformations que le système est capable de réaliser sur la phrase d'origine pour produire une paraphrase. Son fonctionnement sera décrit dans le chapitre 4.

En revanche, en fonction des caractéristiques de la tâche, elle peut impliquer des modifications importantes dans le système de production. De plus, il faut pouvoir définir un modèle joint entre la production de paraphrases et la tâche.

2.3.3 *Évaluation*

Comme nous venons de le voir, la production de paraphrases est un problème difficile, en grande partie parce que les objets manipulés, des phrases et leur sens, ne sont pas aisément formalisables. Cette difficulté se retrouve naturellement pour l'évaluation des paraphrases produites.

Le domaine de la production de paraphrases manque d'une méthodologie d'évaluation automatique des paraphrases. À notre connaissance, la proposition de Callison-Burch et coll. [2008] constitue la seule mesure automatique spécifique pour étudier la qualité des paraphrases. Cette mesure fait l'hypothèse que le générateur de paraphrases fournit un alignement entre la phrase d'origine et la paraphrase proposée. Un alignement est une fonction qui, à chaque élément d'une partition de la phrase d'origine – au sens mathématique du terme, avec les mots comme éléments atomiques – associe un élément d'une partition de la paraphrase proposée. La mesure consiste à comparer des alignements de référence, réalisés manuellement à partir d'un corpus d'évaluation, à ceux d'un générateur de paraphrases. En comptant le nombre de paires communes aux alignements de référence et aux alignements des paraphrases proposées, il est possible de calculer un rappel et une précision. Cette approche est donc fortement contrainte par la taille et l'exhaustivité du corpus d'évaluation.

Compte tenu des difficultés autour de l'évaluation automatique, les différentes études sur la paraphrase réalisent en général des évaluations manuelles. Nous décrivons ci-dessous le protocole présenté dans quatre travaux que nous jugeons significatifs.

Le premier est celui de Barzilay et Lee [2003], l'un des premiers travaux sur la production automatique de paraphrases par apprentissage. L'évaluation est réalisée par deux juges à qui il est demandé si le sens est préservé entre deux phrases. Les juges ne savent pas quelle est la phrase d'origine et quelle est la paraphrase produite automatiquement. Notons que les auteurs fournissent les évaluations de chaque évaluateur ainsi que le coefficient d'accord. En revanche, ils ne proposent pas de « score » regroupant les évaluations des deux juges.

Le second protocole d'évaluation est celui proposé par Dolan et Brockett [2005]. Dans ces travaux, deux juges indiquent si deux phrases sont « sémantiquement équivalentes » ou non. Les juges ne savent pas quelle est la phrase d'origine et quelle est la paraphrase produite automatiquement. En cas de désaccord, un troisième juge prend la décision finale, ce qui permet de calculer un taux de paraphrases correctes.

Le troisième travail est celui de Callison-Burch [2007]. Dans ce cas, la méthodologie utilisée reprend celle proposée en traduction automatique, elle-même inspirée des recommandations du rapport ALPAC [1966]. Pour chaque couple de phrases, deux questions sont posées :

– Quelle part du sens exprimé dans la phrase d'origine est exprimée dans la paraphrase ? [5]

5. [*How much of the meaning of the original phrase is expressed in the paraphrase?*]

– Quelle est la naturalité de la phrase ? [6]

Deux évaluateurs doivent répondre à ces questions sur une échelle allant de 1 à 5. L'échelle pour la première question est [7] :

5 – tout ;

4 – la majorité ;

3 – une grande partie ;

2 – un peu ;

1 – rien.

L'échelle pour la seconde question est [8] :

5 – anglais parfait ;

4 – bon anglais ;

3 – anglais « étranger » ;

2 – anglais incorrect ;

1 – incompréhensible.

Notons que les juges savent quelle est la phrase d'origine. Une paraphrase est jugée correcte lorsqu'elle n'a aucune note strictement inférieure à 3.

Le dernier travail que nous présenterons est celui de Zhao et coll. [2009]. Ici, deux juges notent les paraphrases sur deux échelles allant chacune de 1 à 3. La première échelle, portant sur le sens de la paraphrase, est la suivante [9] :

3 – le sens est complètement préservé ;

2 – le sens est grossièrement préservé ;

1 – le sens est clairement changé.

La seconde échelle, portant sur la naturalité de la paraphrase, est la suivante [10] :

3 – **t** est une phrase parfaite ;

2 – **t** est compréhensible ;

1 – la paraphrase **t** est incompréhensible.

L'article ne mentionne pas si les évaluateurs savent quelle phrase est l'originale lors de la question sur la sémantique. Les auteurs ne définissent pas ce qu'est une paraphrase correcte et ne réalisent donc pas de synthèse des résultats ni pour chaque juge ni sur tous les juges.

Nous proposons le tableau 2.2 comme synthèse des différents protocoles d'évaluation. Nous constatons qu'il n'existe pas de consensus sur le protocole d'évaluation des paraphrases. Ce genre de différences se retrouve dans de nombreux autres travaux. Pour ajouter à la confusion, certains chercheurs changent même de protocole entre deux travaux : ainsi, dans Zhao et coll. [2009], l'évaluation est réalisée sur une échelle à trois niveaux alors que dans Zhao et coll. [2010], l'échelle de

6. [*How do you judge the fluency of the sentence?*]
7. [*5 – All; 4 – Most; 3 – Much; 2 – Little; 1 – None.*]
8. [*5 – Flawless English; 4 – Good English; 3 – Non-native English; 2 – Disfluent English; 1 – Incomprehensible.*]
9. [*3 – The meaning is completely preserved; 2 – The meaning is generally preserved; 1 – The meaning is evidently changed.*]
10. [*3 – **t** is a flawless sentence; 2 – **t** is comprehensible; 1 – the paraphrase **t** is incomprehensible.*]

	Nombre de juges	Nombre de questions	Taille de l'échelle	Originale identifiée	Synthèse des évaluations
Barzilay et Lee [2003]	2	1	2	Non	Non
Dolan et Brockett [2005]	2 + 1	1	2	Non	Oui
Callison-Burch [2007]	2	2	5	Oui	Oui
Zhao et coll. [2009]	2	2	3	?	Non

TABLEAU 2.2: Comparaison de quatre protocoles d'évaluation. La première colonne indique le nombre de juges qui évaluent chaque paraphrase ; la seconde, le nombre de questions posées pour chaque paraphrase à chaque juge ; la troisième, le nombre de réponses possibles pour chaque question ; la quatrième, si le juge sait quelle est la phrase d'origine ; la dernière colonne, si les auteurs définissent ce qu'est une paraphrase correcte et s'ils synthétisent les différentes évaluations. Ce tableau montre clairement qu'il n'existe pas de consensus sur la façon d'évaluer les paraphrases.

notation comporte cinq niveaux. Non seulement la comparaison de deux évaluations humaines faites avec le même protocole est délicate, en raison des variations entre juges, mais elle devient impossible en raison de la prolifération des protocoles.

Lorsqu'un générateur de paraphrases est intégré dans un système de TAL et que les paraphrases produites sont une étape de calcul intermédiaire, alors il est possible d'utiliser les protocoles d'évaluation spécifiques au système global. Par exemple, Callison-Burch [2007] évalue l'apport du générateur de paraphrases dans un système de traduction statistique en utilisant les mesures du domaine de la traduction automatique – comme BLEU [Papineni et coll., 2002]. Ces méthodes permettent de mesurer automatiquement l'apport d'un système de paraphrases mais ne permettent pas de connaître la qualité intrinsèque des paraphrases produites. De plus, lorsque ces phrases sont les sorties du système global, comme pour la synthèse vocale, une évaluation extrinsèque n'est pas suffisante. Par exemple, une paraphrase peut améliorer la qualité acoustique d'un système de synthèse vocale, mais si elle ne conserve pas le sens de la phrase d'origine, alors on ne peut pas dire que le système est amélioré. Dans ce cas, l'évaluation intrinsèque de la qualité de la paraphrase est nécessaire pour évaluer le système global.

2.4 CONCLUSION

Ce chapitre a présenté un tour d'horizon et un état de l'art de la production de paraphrases en TAL. Nous avons présenté les applications pour les paraphrases. Nous avons spécifié notre cadre d'utilisation des paraphrases pour les systèmes vocaux humain-machine. Nous avons présenté les travaux antérieurs s'approchant de ce cadre. Enfin, nous avons élargi notre tour d'horizon des travaux sur la production de paraphrases en présentant la façon dont sont abordées dans la littérature les

trois problématique principales : la production, l'évaluation et l'utilisation des paraphrases.

Nous n'avons que peu abordé le problème de la constitution des ressources, car il est souvent trop dépendant de la méthode de production choisie, comme nous l'avons montré dans la section 2.3.1. Malgré tout, la constitution des ressources reste un problème crucial qui attire d'ailleurs la majorité des efforts de la communauté travaillant sur les paraphrases [Quirk et coll., 2004; Sekine, 2005; Bouamor et coll., 2007]. Nous traiterons de ce problème dans la section 4.3, après avoir retenu une approche pour la production de paraphrases.

Ce chapitre a donc constitué notre point de départ quant à l'étude de la production de paraphrases. La réalisation d'expériences sur la production de paraphrases pour les systèmes vocaux humain machine nécessite :

- un protocole d'évaluation que nous proposerons dans le chapitre 3 ;
- un système de production que nous présenterons dans le chapitre 4 ;
- l'intégration dans un système vocal que nous réaliserons dans le chapitre 5.

Deuxième partie

ÉTUDIER : UN CADRE DE LA PRODUCTION DE PARAPHRASES

3 UN PROTOCOLE D'ÉVALUATION

La conception d'un outil de production de paraphrases doit être accompagnée d'une méthode d'évaluation afin de mesurer son efficacité. Nous avons montré dans la section 2.3.3 que l'évaluation des générateurs de paraphrases est un problème difficile que beaucoup de travaux abordent par des méthodes différentes et incompatibles entre elles. Ce chapitre se propose donc de présenter un protocole d'évaluation, inspiré des travaux du domaine, afin d'étudier la production de paraphrases sur une base commune à toutes les expériences.

La section 3.1 présentera une réflexion sur la forme d'une évaluation des paraphrases. Une plateforme d'évaluation sera présentée dans la section 3.2. Nous définirons notre formalisme de présentation des résultats dans la section 3.3. Enfin, la section 3.4 décrira les différents corpus utilisés pour nos expériences d'évaluation.

3.1 FORME DE L'ÉVALUATION

Deux aspects sont à prendre en compte lors de la conception de tout protocole d'évaluation :

- le critère à mesurer ;
- l'évaluation : peut-elle être automatisée ou bien doit-elle être réalisée avec l'assistance de juges humains ?

Interrogeons nous d'abord sur ce qu'il faut mesurer pour déterminer si une phrase est une paraphrase.

Les différents usages des paraphrases présentés dans la section 2.1.1 ont montré qu'un générateur de paraphrases est souvent un composant d'un système plus complexe. Il serait donc légitime de réaliser l'évaluation du système complet, à l'aide d'un critère global, afin de mesurer l'impact du générateur de paraphrases. Le problème est que pour les applications où la paraphrase est présentée au juge final, il n'est pas possible de se contenter d'une évaluation de la tâche, comme dans Cahill et coll. [2009]. Par exemple, un générateur de paraphrases peut être utilisé conjointement à un système de synthèse vocale. Si celui-ci introduit une négation, il se peut que la synthèse vocale soit améliorée, mais le générateur dénature complètement le message d'origine. Un tel message ne peut être considéré comme paraphrase. De plus, une évaluation comportant uniquement un critère lié à une tâche ne permet pas de généraliser les résultats pour d'autres problèmes. C'est pourquoi il existe peu de protocoles d'évaluation orientés tâche [Callison-Burch et coll., 2008].

Comme nous l'avons montré dans la section 2.3.3, lorsqu'ils ne sont pas évalués en fonction des performances globales d'un système, les générateurs sont évalués selon le critère fondamental de la définition des paraphrases : la notion de conservation du sens [Barzilay et Lee, 2003; Quirk et coll., 2004; Bannard et Callison-Burch, 2005; Max, 2008].

À cause des difficultés de formalisation inhérentes à la notion de *sens*, en l'état actuel des connaissances, il est difficile d'envisager des mesures automatiques d'évaluation des paraphrases. Deux principaux points empêchent de réaliser des évaluations automatiques sur le principe de celles utilisées en traduction.

Tout d'abord, il est difficile de développer une liste de paraphrases pour une phrase donnée. Lorsqu'il est demandé à des personnes de produire une telle liste, elles oublient des paraphrases que peuvent proposer un système automatique de production [Lin et Pantel, 2001]. L'évaluation est donc limitée par l'absence de références suffisamment exhaustives.

Ensuite, dans le domaine de la traduction, l'hypothèse qui est faite est que l'objectif n'est pas forcément de fournir une bonne traduction, mais de fournir une traduction proche de celle que ferait un traducteur humain [Papineni et coll., 2002]. Il n'est donc pas nécessaire de comparer une traduction à toutes celles possibles. Il suffit de la comparer à un ensemble de traductions humaines. Le problème est que la paraphrase ne correspond pas à une activité humaine autonome et archivée. De plus, si de telles références existent, elles ne sont pas nécessairement adaptées à la tâche pour laquelle les paraphrases sont produites. L'évaluation ne dispose donc pas de paraphrases de référence à atteindre.

Beaucoup de travaux évaluent donc leurs paraphrases en faisant appel à des jugements humains [Barzilay et McKeown, 2001; Quirk et coll., 2004; Bannard et Callison-Burch, 2005]. Malheureusement, il n'y a pas de consensus sur un protocole d'évaluation des paraphrases comme nous l'avons montré dans la section 2.3.3.

3.2 PLATEFORME D'ÉVALUATION

Cette section décrit une plateforme d'évaluation des paraphrases, et un ensemble de réflexions qui a accompagné sa réalisation. Compte tenu des remarques de la section précédente, cette plateforme s'appuie sur le jugement humain afin de mesurer si une paraphrase conserve bien le sens de la phrase pour laquelle elle a été produite.

Pour réaliser cette plateforme, il faut pouvoir répondre aux questions suivantes :
- quelle question poser ?
- comment poser cette question et à qui ?
- comment les juges vont-ils pouvoir répondre ?

3.2.1 *Questions posées*

Un point primordial dans l'évaluation subjective est la forme de la question posée aux juges. Barzilay et McKeown [2001] fournissent aux juges un ensemble de directives définissant les paraphrases comme *une équivalence conceptuelle approximative*. De façon similaire, Dolan et Brockett [2005] demandent aux juges si les deux phrases présentées *signifient la même chose*. De leur côté, Bannard et Callison-Burch [2005] décomposent l'évaluation en deux étapes en s'inspirant des protocoles du domaine de la traduction automatique. L'une de ces deux étapes cherche à évaluer la syntaxe : l'objectif est de savoir si la paraphrase est grammaticale. L'autre étape évalue la sémantique ; elle est conçue pour déterminer si le sens de la phrase d'origine se retrouve dans la paraphrase. Cette décomposition en deux étapes semble

adaptée pour les paraphrases. En effet, leur définition en elle-même fait ressortir ces deux aspects : une paraphrase est une phrase, donc syntaxiquement correcte, qui préserve le sens de la phrase d'origine.

Un des objectifs du protocole proposé ici est de simplifier le plus possible la tâche d'évaluation afin de travailler sur de grands corpus à évaluer. Dans ce but, les cinq ou neuf réponses possibles pour le juge, proposées dans le domaine de traduction automatique, semblent trop complexes. Nous réduisons le nombre de réponses au minimum, c'est-à-dire à un choix binaire « Oui », « Non ». Malgré tout, nous ne souhaitons pas forcer le juge à répondre lorsqu'il ne s'estime pas en mesure de le faire. Nous introduisons donc une troisième réponse possible : « Ne sais pas ». Nous ne savons pas, *a priori*, quels sont les cas concernés par cette réponse.

Les questions posées se doivent aussi d'être les plus simples possibles sans introduire de vocabulaire technique. La question posée aux juges pour l'étape syntaxique est :

La phrase suivante est-elle en bon français ?

La question posée aux juges pour l'étape sémantique est :

Les deux phrases suivantes veulent-elles dire la même chose ?

La simplicité de ces questions peut laisser des ambiguïtés sur la réponse attendue. Dans le but d'aider les juges, nous reprenons l'initiative de Barzilay et McKeown [2001] en fournissant aux juges des exemples de réponses commentées. Ces exemples sont présentés dans les tableaux 3.1 et 3.2.

Les premiers essais de notre interface ont causé une fatigue importante des juges lors de la tâche d'évaluation sémantique. Il semble relativement pénible de lire deux textes en parallèle afin de déterminer si leurs sens sont identiques. Aussi, afin de réduire la charge cognitive des juges, les mots différents entre la paraphrase et la phrase originale sont colorés. De la même façon, lors de la tâche syntaxique, les écarts avec la phrase d'origine sont colorés même si celle-ci n'est pas présentée. Ces différences sont simplement calculées à l'aide du programme *diff* [Hunt et McIlroy, 1976]. En effet, les changements introduits dans la paraphrase sont des espaces privilégiés pour l'introduction d'une erreur syntaxique. Les retours des premiers juges ont montré que l'ajout de cette fonctionnalité est grandement apprécié. Cela simplifie le travail et accélère l'évaluation, en particulier pour de longues phrases.

3.2.2 *Protocole d'évaluation*

Puisque le processus d'évaluation est constitué de deux tâches d'évaluation différentes, il est nécessaire de définir la façon de présenter les différentes questions aux juges et en particulier leur enchaînement.

L'évaluation ne doit pas être trop monotone pour le juge. De plus, il est préférable que chaque tâche, sémantique et syntaxique, avance à peu près à la même vitesse afin d'avoir des résultats partiels exploitables. C'est pourquoi la plateforme alterne régulièrement les deux tâches. D'un autre côté, le juge doit acquérir une certaine habitude dans une tâche. Comme le but est de ne pas lui demander un effort de réadaptation trop fréquent, il est préférable qu'un juge réalise un certain nombre d'évaluations d'une même tâche avant d'en changer. Il faut donc trouver un juste milieu. Des séries de dix questions du même type nous semblaient être un bon

Phrases		Consigne	Explication
Cette question est extrêmement importante !	Ce point est d'une importance capitale !	Oui	Les deux phrases disent la même chose.
Je rentre tard.	Je rentre tôt.	Non	Les deux phrases ne disent pas la même chose.
C'est un joli petit chat.	C'est un joli petit chat noir.	Non	La seconde phrase est beaucoup plus précise car avec la première, on ne peut pas savoir que le chat est noir.
Je la trouve mignonne.	Je la trouve jolie.	Oui	Même si les deux phrases ne sont pas identiques, elles disent *à peu près* la même chose.
Il verse de l'argent sur son compte.	Il remplit de l'argent sur son compte.	Non	Dans ce contexte, *remplit* n'est pas synonyme de *verse*.
Il me remplit un verre de vin.	Il me verse un verre de vin.	Oui	Les phrases disent la même chose.
J'aime la soupe à la tomate.	Soupe moi tomate.	Non	La seconde phrase ne veut rien dire.
Je veux de la soupe à la tomate.	Moi veux soupe tomate.	Oui	Même si la seconde phrase n'est pas en bon français, elle sera comprise par la plupart des personnes de la même façon que la première phrase.

TABLEAU 3.1: Exemples fournis lors des évaluations sémantiques.

Phrases	Consigne	Explication
La vie est belle !	Oui	Cette phrase est en bon français.
La vie est beaux.	Non	Problème d'accord entre "vie" et "beaux".
Le ciel est beau. Et bleu	Non	Le point est mal placé ou alors il manque un second point à la fin.
La vie) est belle " !!!	Non	Il y a des symboles de ponctuations en trop.
Moi aime tomate.	Non	La phrase n'est pas en bon français.
En conséquence ,j' aime le bleu .	Oui	Les règles de positionnement des espaces (en particulier pour la ponctuation) ne sont pas à prendre en compte.

TABLEAU 3.2: Exemples fournis lors des évaluations syntaxiques.

compromis entre ces deux aspects. Après plusieurs séries d'évaluations, les juges nous ont demandé de réduire ce paramètre à cinq afin d'avoir des enchaînements plus fréquents et de réduire la monotonie de l'évaluation.

Le processus d'évaluation se présente ainsi : lorsqu'un juge se connecte et s'identifie, le programme lui propose de réaliser une séquence de cinq évaluations d'une même tâche, sémantique ou syntaxique. À la fin de cette série, l'interface suggère une nouvelle série de cinq évaluations dans l'autre tâche et ainsi de suite. Le juge peut interrompre son travail à tout moment. Lors d'une connexion ultérieure, il reprendra la série à l'endroit où il s'était arrêté.

La paraphrase à évaluer est choisie au hasard. Néanmoins, ce choix est biaisé afin de favoriser les phrases qui ont le moins d'évaluations, tout en privilégiant les phrases qui n'ont pas le même nombre d'évaluations dans les deux tâches. Ainsi, même si tout le corpus n'a pas été complètement évalué, on est assuré d'avoir un sous-ensemble qui possède suffisamment d'évaluations afin d'effectuer des analyses préliminaires des résultats.

3.2.3 *Interface d'évaluation*

Les différents protocoles rencontrés dans la littérature reposent principalement sur une évaluation à deux juges [Barzilay et McKeown, 2001; Quirk et coll., 2004; Bannard et Callison-Burch, 2005]. Mais comme la définition des paraphrases reste imprécise et que l'évaluation n'est pas réalisée par des experts, c'est en fait une forme de consensus social qui est recherché. Il n'est donc pas évident que ce consensus puisse être capturé avec seulement deux juges. Le coefficient kappa d'accord entre les juges est généralement proche de 0,6 ce qui est traditionnellement interprété entre un accord *modéré* et un accord *bon* [Barzilay et Lee, 2003; Bannard et Callison-Burch, 2005], voir la section 3.3.2. Afin de rechercher ce consensus social, notre plateforme, de par sa conception, n'impose pas un nombre fixe de juges pour chaque phrase. Nous verrons ci-dessous que nous n'avons pas utilisé cette fonctionnalité que nous estimons pourtant désirable.

L'interface d'évaluation que nous avons développée prend la forme d'une application web. Le système a été réalisé en PHP et utilise une base de données MySQL. L'avantage d'un outil en ligne est qu'il laisse le choix aux juges du lieu, de l'heure et de la fréquence de leurs connexions. De plus, il est très simple d'ajouter un nouveau juge pour une expérience d'évaluation déjà en cours.

Les figures 3.1 et 3.2 présentent des captures d'écran de notre interface.

3.3 PRÉSENTATION DES RÉSULTATS

Maintenant que nous disposons d'une interface d'évaluation, il nous faut exploiter les résultats. Dans cette section, nous traiterons de la présentation et de l'interprétation des résultats d'évaluation. Nous expliquerons dans la section 3.3.1 notre façon de présenter les résultats d'évaluation. De plus, puisque les évaluations cherchent à capturer une forme de consensus social sur la conservation du sens, nous expliquerons dans la section 3.3.2 comment nous mesurerons l'accord entre les juges. Enfin, nous décrirons dans la section 3.3.3 notre formalisme quant à la présentation des paraphrases.

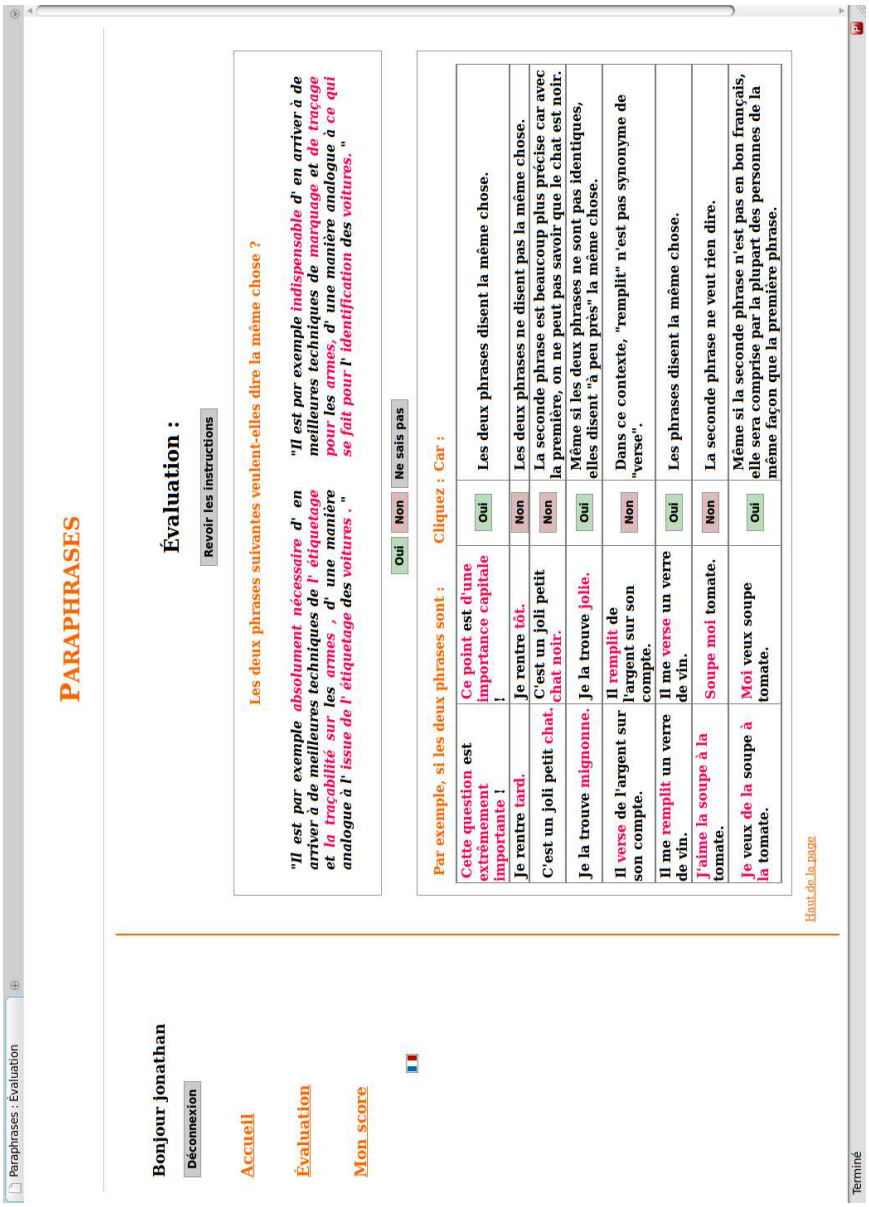


FIGURE 3.1: Captures d'écran d'une partie de l'interface d'évaluation lors de la phase d'évaluation sémantique. Deux phrases sont présentées au juge et il doit indiquer si elles disent la même chose. Afin de simplifier la tâche, un ensemble d'exemples est fourni et les différences entre les deux phrases sont mises en couleur.

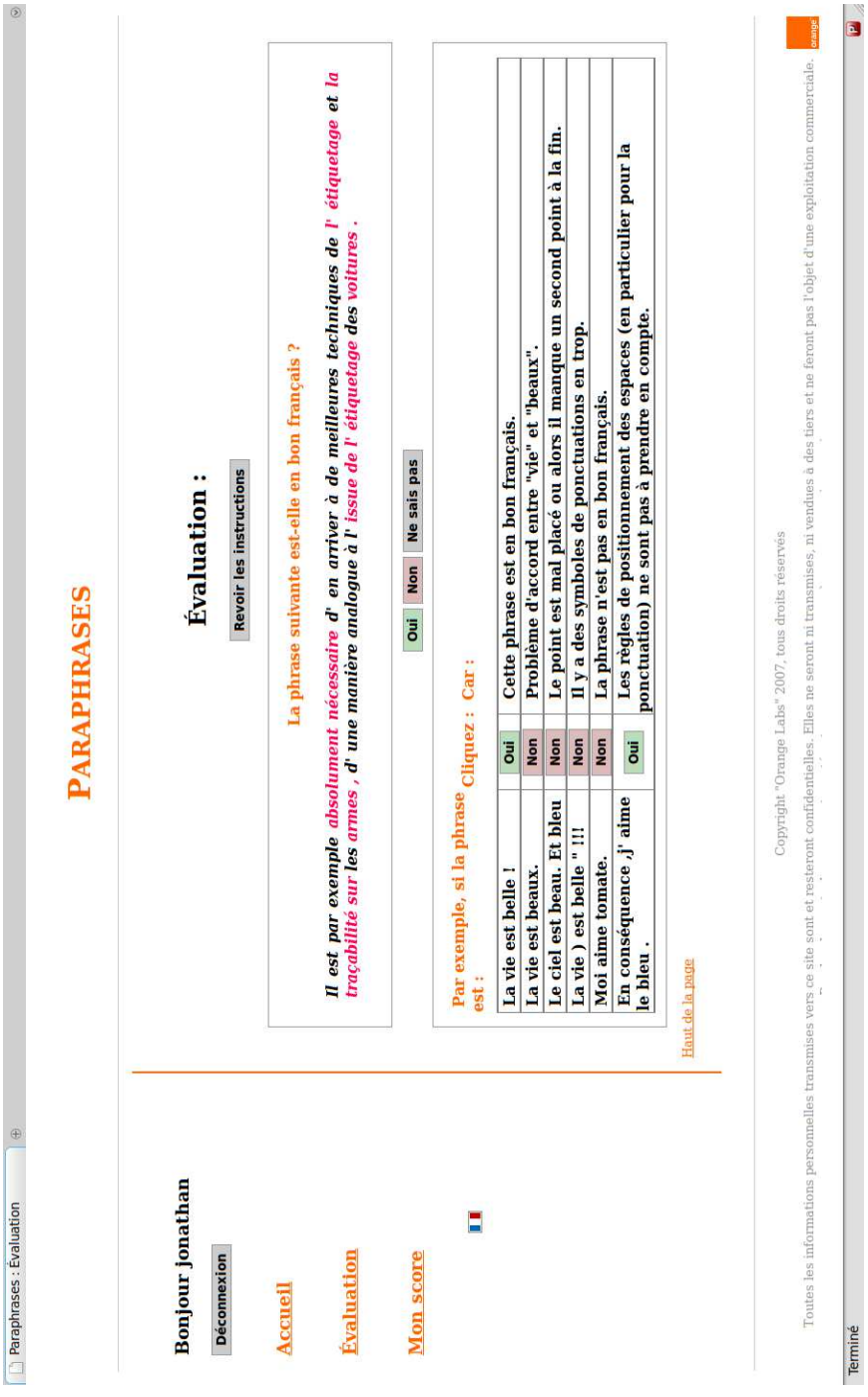


FIGURE 3.2: Captures d'écran d'une partie de l'interface d'évaluation lors de la phase d'évaluation syntaxique. Le juge doit indiquer si la paraphrase est en bon français. Les différences avec la phrase d'origine sont mises en couleur afin de mettre en évidence les endroits où il risque d'y avoir des erreurs.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	49	8	48	10	66	11
oui	3	40	10	32	2	**22**
Kappa	**0,69** (p-valeur $< 10^{-3}$) **Accord substantiel**					

TABLEAU 3.3: Exemple de résultats d'une évaluation d'un générateur de paraphrases par deux juges. Les colonnes correspondent aux réponses d'un juge et les lignes à celles de l'autre. Pour être correcte, une paraphrase doit être jugée correcte syntaxiquement et sémantiquement.

3.3.1 *Évaluation d'un générateur de paraphrases*

L'ensemble des expériences réalisées dans ce document a été évalué par deux juges. Les résultats d'une évaluation d'un générateur de paraphrases par ces deux juges seront présentés comme dans le tableau 3.3.

Dans ce type de tableau, sauf indication contraire, les lignes correspondent aux évaluations du premier juge et les colonnes à celles du second. Cette présentation, sous forme d'une matrice de diffusion, permet de comparer le comportement des deux juges.

Si un juge évalue une paraphrase syntaxiquement ou sémantiquement incorrecte – une réponse non – alors la paraphrase est jugée incorrecte. Pour qu'une paraphrase soit jugée correcte, il faut que les deux évaluateurs la jugent correcte, c'est-à-dire qu'elle n'a aucune réponse « non ».

3.3.2 *Accord entre les juges*

Afin d'évaluer la pertinence des résultats, il est important de pouvoir mesurer l'accord entre les juges. Une approche simple pour mesurer l'accord consiste à calculer la proportion d'accords observée, comme dans Quirk et coll. [2004]. Nous disposons d'une matrice de diffusion des résultats R, telle que présentée dans la formule 3.1.

$$R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \tag{3.1}$$

Dans notre cadre, r_{11} correspond au nombre de paraphrases jugé correcte par les deux évaluateurs ; r_{22} au nombre de paraphrases jugé incorrecte par les deux évaluateurs ; r_{12} le nombre de paraphrases jugé correcte par le premier évaluateur et incorrecte par le second ; r_{21} le nombre de paraphrases jugé incorrecte par le premier évaluateur et correcte par le second.

Alors, la proportion d'accords observée entre les juges est donnée par la formule 3.4.

$$\begin{aligned} P_O &= P(J_1 = J_2) && (3.2)\\ &= P(J_1 = \text{Oui} \cap J_2 = \text{Oui}) + P(J_1 = \text{Non} \cap J_2 = \text{Non}) && (3.3)\\ &= \frac{r_{11} + r_{22}}{r_{11} + r_{12} + r_{21} + r_{22}} && (3.4) \end{aligned}$$

avec J_i le jugement du i^{e} évaluateur.

Ce qui donne, par exemple, pour les données du tableau 3.3 :

$$P_O = \frac{49 + 40 + 48 + 32}{100 + 100} = 0{,}845 \tag{3.5}$$

L'interprétation est que les juges donnent donc la même évaluation dans 84,5% des cas. Notons qu'il est difficile de calculer un écart-type non biaisé pour la moyenne car dans le cas présent nous avons seulement deux juges.

Une meilleure manière de faire consiste à calculer l'accord entre les juges grâce au coefficient Kappa [Cohen, 1960]. Pour celui-ci l'accord observé entre des jugements résulte d'une composante « aléatoire » et d'une composante d'accord « véritable ».

Le coefficient Kappa s'écrit :

$$\kappa = \frac{P_O - P_A}{1 - P_A} \tag{3.6}$$

où P_O est la proportion d'accords observée et P_A la proportion d'accords aléatoire attendue sous l'hypothèse d'indépendance des jugements. Dans le cas d'une évaluation bipartite par deux juges, P_O s'écrit avec la même formule que 3.4 et P_A comme suit :

$$\begin{aligned} P_A &= P(J_1 = \text{Oui} \cap J_2 = \text{Oui}) + P(J_1 = \text{Non} \cap J_2 = \text{Non}) && (3.7)\\ &= P(J_1 = \text{Oui}) \times P(J_2 = \text{Oui}) + P(J_1 = \text{Non}) \times P(J_2 = \text{Non}) && (3.8)\\ &= \frac{\big((r_{11} + r_{12}) \times (r_{11} + r_{21})\big) + \big((r_{21} + r_{22}) \times (r_{12} + r_{22})\big)}{(r_{11} + r_{12} + r_{21} + r_{22})^2} && (3.9) \end{aligned}$$

Notons que la décomposition des probabilités lors du passage de l'équation 3.7 à l'équation 3.8 n'est possible que sous hypothèse d'indépendance statistique des évaluations, ce qui est le cas si l'accord entre les juges est aléatoire.

Le coefficient Kappa est un réel entre -1 et 1 :

- l'accord est maximal lorsque $\kappa = 1$;
- l'accord est accidentel lorsque $\kappa = 0$;
- le désaccord est maximal lorsque $\kappa = -1$.

Le tableau 3.4 reprend l'interprétation de l'accord, en fonction de la valeur du Kappa, proposée par Landis et Koch [1977]. En réalité, les frontières entre les catégories sont arbitraires et restent dépendantes de la tâche. Elles permettent surtout d'avoir des points de repères pour interpréter un κ observé.

La valeur κ est calculée selon la formule 3.6, à partir d'observations. La vraie valeur du coefficient Kappa est donc une variable aléatoire qui suit une loi normale de moyenne κ et d'écart-type σ_κ. Un test statistique est nécessaire pour s'assurer

Kappa	Accord
]0,80; 1]	parfait
]0,60; 0,80]	substantiel
]0,40; 0,60]	modéré
]0,20; 0,40]	médiocre
[0,00; 0,20]	mauvais
[−1; 0[	très mauvais

TABLEAU 3.4: Interprétation traditionnelle des valeurs du coefficient Kappa.

que la valeur de l'estimateur κ n'est pas accidentelle. Il faut donc tester l'hypothèse nulle $H_0 : \kappa = 0$ contre l'hypothèse $H_1 : \kappa > 0$. Si la p-valeur est inférieure à 5% alors l'accord observé n'est pas accidentel.

Pour notre exemple du tableau 3.3, le Kappa est de 0,69 ce qui est traditionnellement interprété comme un accord substantiel. Comme la p-valeur est inférieure à 10^{-3} cette observation n'est pas accidentelle.

3.3.3 *Présentation des paraphrases*

Afin d'analyser plus finement les résultats d'évaluation, nous allons devoir illustrer les capacités des algorithmes par des exemples de paraphrases produites. Dans la suite de ce document, par convention d'écriture, lorsque nous illustrerons les résultats par des paraphrases, elles seront présentées selon le modèle suivant :

Paraphrase 3.1 – *Exemple :*
O : La ***phrase d'origine****.*
P : La ***paraphrase****.*

La phrase d'origine est appelée « O » et la paraphrase « P ». Une paraphrase jugée incorrecte sera précédée d'un astérisque. Les différences entre les deux phrases sont signalées en couleur et en gras : **différence**. Notons que cette mise en avant ne correspond pas nécessairement aux segments de la table de paraphrases utilisés pour produire la paraphrase – voir la section 4.3 – mais uniquement aux mots – et symboles – différents calculés par *diff* [Hunt et McIlroy, 1976].

3.4 CORPUS D'EXPÉRIMENTATION

Nous sommes désormais en mesure de produire des évaluations et de les discuter. Afin d'éviter les biais d'évaluation, encore faut-il que tous les systèmes que nous évaluerons travaillent sur les mêmes données. Nous traitons dans cette section de la constitution des corpus d'évaluation nécessaires à la réalisation d'expériences sur les paraphrases.

Les générateurs statistiques de paraphrases, que nous décrirons au chapitre 4, utilisent un corpus d'entraînement bilingue aligné. La relation de proximité sémantique que ces systèmes arrivent à capturer est conditionnée par ce corpus d'apprentissage. Pour s'assurer que les corpus de tests puissent être traités par les générateurs de

	Entraînement français	Entraînement anglais	TEST 1	TEST 2
Nombre de phrases	1 576 997	1 576 997	100	100
Nombre de mots	42 157 697	38 449 414	2 637	2 631
Longueur moyenne des phrases	26,7±12,9	24,4±11,7	26,4±11,7	26,3±11,8
Taille du vocabulaire	130 342	113 769	969	995

TABLEAU 3.5: Statistiques des corpus d'entraînement et de test. Les phrases sont longues en moyenne.

paraphrases à évaluer, ceux-ci doivent appartenir au même domaine que le corpus d'apprentissage.

Le corpus EUROPARL [Koehn, 2005] est l'une des plus importantes ressources de corpus bilingues alignés disponibles. C'est la cinquième édition du corpus Europarl [2010] que nous utilisons. Ce corpus est composé des transcriptions des débats du parlement européen pour les années comprises entre 1996 et 2009 et traduits en onze langues. Pour les différentes expériences que nous avons effectuées, la langue des paraphrases est le français. La langue pivot utilisée est l'anglais. La version français-anglais d'EUROPARL est constituée de 1 723 705 phrases représentant 51 708 806 mots pour le français et pour 47 915 991 mots pour l'anglais.

Nous extrayons deux jeux de test du corpus EUROPARL. Le premier jeu sera dédié aux évaluations dites subjectives réalisées sur la plateforme d'évaluation que nous venons de présenter à la section 3.2. Le second servira aux mesures d'optimisation décrites dans la troisième partie de ce document au chapitre 8. Nous utilisons ce second jeu afin d'éviter des biais de sur-apprentissage lors des évaluations subjectives.

Chaque jeu de test est constitué de 100 phrases en français. Pour les construire, 210 phrases sont extraites aléatoirement du corpus complet et affectées aléatoirement à l'un des deux corpus. Le corpus EUROPARL pouvant contenir des erreurs, nous choisissons d'extraire plus de phrases que nécessaire afin de pouvoir supprimer des phrases mal formées. De plus, nous réalisons une correction manuelle des phrases restantes. Les phrases qui n'ont pas été extraites aléatoirement pour former les corpus de test sont utilisées comme corpus d'entraînement. Les caractéristiques des corpus sont présentées dans le tableau 3.5.

3.5 CONCLUSION

La plateforme d'évaluation ainsi que les corpus décrits précédemment permettent d'évaluer des systèmes de production de paraphrases selon les critères du domaine : la conservation du sens et la naturalité des paraphrases. L'ensemble des réflexions

sur les méthodologies d'évaluation des paraphrases nous ont permis de mettre en place une plateforme la plus simple possible pour les juges.

Suite à l'observation de plusieurs séries d'évaluations, nous estimons qu'un juge sans entraînement met approximativement 1h15 pour évaluer les 100 paraphrases du jeu TEST 1. Ceci correspond à 200 questions (100 questions sur la naturalité et 100 questions sur la conservation du sens) soit une moyenne de 45 secondes par question. Ce temps paraît raisonnable mais il semble difficilement améliorable compte tenu de la difficulté de la tâche. Il n'en reste donc pas moins que l'évaluation d'un système de production de paraphrases est une opération coûteuse. En l'absence de mesures automatiques, ceci reste un facteur limitant fortement l'expérimentation des générateurs de paraphrases.

La difficulté de la tâche d'évaluation et le volume nécessaire pour apprécier la performance d'un système font qu'il est difficile de motiver un juge sur une évaluation complète. L'introduction d'un caractère ludique, comme pour l'acquisition de relations sémantiques [Lafourcade, 2007], permettrait probablement d'obtenir des évaluations de juges spontanément volontaires. La transformation de l'évaluation des paraphrases sous la forme d'un jeu reste une perspective de recherche importante mais compliquée [Bouamor et coll., 2007].

À l'heure actuelle, notre plateforme se contente de proposer un score aux juges. Celui-ci correspond au pourcentage de paraphrases évaluées multiplié par l'écart type entre les réponses fournies par le juge et la moyenne des réponses des autres juges (en attribuant des poids de *-1* à une réponse *Non*, *0* à un *Ne sais pas* et *1* à un *Oui*). Ce score qui est principalement informatif pour les juges reste insuffisant pour attirer des joueurs dans une compétition.

En l'absence de mesure automatique de la qualité des paraphrases, la recherche d'une méthode d'évaluation ludique nous semble la plus prometteuse pour évaluer des systèmes à grande échelle.

4 UN GÉNÉRATEUR DE PARAPHRASES PAR LANGUE PIVOT

En l'absence de système de production de paraphrases de référence disponible et compatible avec la problématique présentée dans la section 2.2, nous allons d'abord réaliser un premier générateur. Ce système nous servira de référence, nous permettra d'analyser les points forts et faibles de l'état de l'art et servira de point de départ à notre réflexion sur la paraphrase.

La réalisation de ce système de référence s'inspirera de l'état de l'art et utilisera au maximum les outils fournis par la communauté. Ce générateur restera relativement simple tout en permettant de passer à l'échelle en terme de taille du vocabulaire et du nombre de paraphrases produites.

Pour ces raisons, parmi l'ensemble des paradigmes que nous avons présenté dans le chapitre précédent, nous retenons celui de la production statistique de paraphrases. Cette approche a déjà fait l'objet de nombreux travaux [Quirk et coll., 2004; Bannard et Callison-Burch, 2005; Max, 2008; Zhao et coll., 2009]. La production statistique de paraphrases est fondée sur l'apprentissage à partir de grands corpus textuels ce qui lui permet de couvrir beaucoup de phénomènes et de s'adapter facilement à un domaine – à partir du moment où le corpus est disponible ou constructible à un coût raisonnable. Ce paradigme est directement inspiré de la traduction statistique qui est le courant dominant en traduction automatique et bénéficie donc de ses avancées.

Dans ce chapitre, nous présenterons d'abord le modèle théorique de la production statistique de paraphrases dans la section 4.1. Dans les sections 4.2, 4.3 et 4.4 nous présenterons certaines approches possibles et détaillerons nos choix pour la réalisation des divers composants nécessaires à la réalisation d'un générateur statistique de paraphrases. Nous aborderons dans la section 4.5 le problème de l'optimisation du modèle. Enfin nous récapitulerons les caractéristiques de notre système de référence dans la section 4.6.

4.1 MODÈLE DE PRODUCTION STATISTIQUE

La production statistique de paraphrases est une version monolingue de la traduction statistique par segments [Quirk et coll., 2004]. La langue vers laquelle le système « traduit » est la même que la langue source.

La traduction statistique par segments [Koehn et coll., 2003] repose sur le modèle du canal bruité introduit par Shannon et Weaver [1949]. Dans ce modèle, la phrase source [1] s est le message reçu lorsque la traduction c a été transmise au travers d'un canal bruité $B_{L_c \to L_s}$ entre la langue cible L_c et la langue source L_s. La traduction est alors une opération de décodage au sens de la théorie de l'information. L'objectif est de trouver une séquence de mots qui produit la phrase source avec la

1. Dans ce chapitre, nous parlons de « phrase source », conformément au formalisme du modèle du canal bruité, pour ce que nous nommions précédemment « phrase d'origine ». Dans ce document, ces deux appellations sont équivalentes.

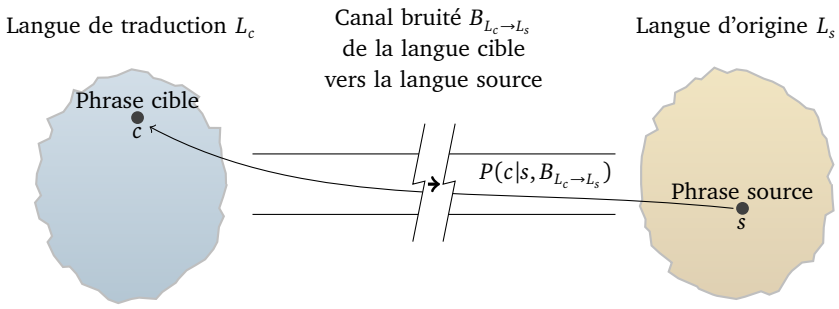


FIGURE 4.1: Modèle du canal bruité. La phrase source est vue comme le message reçu lorsque la traduction a été transmise au travers d'un canal bruité. Le décodage consiste à trouver la traduction la plus probable sachant la phrase source.

plus forte probabilité sachant une modélisation du canal bruité. Plus formellement, la traduction consiste à trouver :

$$c^* = \arg\max_c P(c|s, B_{L_c \to L_s}) \tag{4.1}$$

La figure 4.1 illustre le modèle du canal bruité.

Afin de le rendre calculable, le modèle est décomposé en plusieurs facteurs. La formule de Bayes est d'abord utilisée :

$$c^* = \arg\max_c \frac{P(c|B_{L_c \to L_s}) \times P(s|c, B_{L_c \to L_s})}{P(s|B_{L_c \to L_s})} \tag{4.2}$$

D'après le modèle, la traduction c est indépendante du canal bruité puisqu'il s'agit du message de départ. Ceci permet de simplifier $P(c|B_{L_c \to L_s})$ ainsi que d'éliminer le facteur de normalisation $P(s|B_{L_c \to L_s})$:

$$c^* = \arg\max_c P(c) \times P(s|c, B_{L_c \to L_s}) \tag{4.3}$$

La formule 4.3 introduit le facteur $P(c)$ qui représente un modèle de la langue vers laquelle la traduction est faite. Un des objectifs de ce facteur est de garantir la « naturalité » de la traduction. En revanche, le terme $P(s|c, B_{L_c \to L_s})$ n'est pas plus facilement calculable que $P(c|s, B_{L_c \to L_s})$.

La phrase c est alors découpée en segments supposés statistiquement indépendants. Chaque segmentation est contiguë, par hypothèse. L'utilisation de segments discontigus reste un problème ouvert en raison des nombreuses problématiques nouvelles que cela introduit [Chiang, 2005]. Soit I une partition continue des mots de c en segments indépendants, avec c_i^I la séquence de mots de c constituant le i^{e} segment de I. Ainsi :

$$c^* = \arg\max_c P(c) \times \sum_I \left(P(I|B_{L_c \to L_s}) \prod_{i \in I} P(s_i^I|c_i^I, B_{L_c \to L_s}, I) \right) \tag{4.4}$$

Deux hypothèses simplificatrices sont faites à cette étape. La première consiste à attribuer une distribution uniforme aux $P(I|B_{L_c \to L_s})$. En effet, il semble difficile

de modéliser autrement cette distribution. Afin de simplifier le problème, on ignore ce terme. La seconde hypothèse est plus critiquable. Elle consiste à ignorer la somme sur l'ensemble des découpages en approximant $\arg\max_c P(c|s, B_{L_c \to L_s})$ par $\arg\max_c \max_I P(c|s, B_{L_c \to L_s}, I)$. Cette hypothèse est réalisée afin de pouvoir appliquer des heuristiques de programmation dynamique lors de la production comme l'explique la section 4.4.

Le modèle de la traduction statistique par segments devient donc :

$$c^* = \arg\max_c P(c|s, B_{L_c \to L_s}) \approx \arg\max_c P(c) \times \max_I \prod_{i \in I} P(s_i^I | c_i^I, B_{L_c \to L_s}, I) \tag{4.5}$$

Notons que l'information qu'apporte le découpage I pour déterminer le segment s_i^I est incluse dans celle du segment c_i^I. C'est pourquoi $P(s_i^I | c_i^I, B_{L_c \to L_s}, I)$ est égale à $P(s_i^I | c_i^I, B_{L_c \to L_s})$.

Traditionnellement, un modèle de distorsion est introduit à cette étape pour pouvoir changer l'ordre des segments. Son objectif est de modéliser les différences d'ordre des mots entre la langue source et celle visée par la traduction. Cela permet, par exemple, de traiter l'inversion nom-adjectif fréquente lors du passage de l'anglais vers le français. Il est admis que pour le problème de la paraphrase, le modèle de distorsion n'est pas nécessaire puisque qu'il n'y a qu'une seule langue [Quirk et coll., 2004; Bannard et Callison-Burch, 2005; Zhao et coll., 2009]. Nous conserverons nous aussi cette hypothèse pour nos travaux.

Un générateur statistique de paraphrases a donc besoin de trois composants. Ces trois composants se retrouvent dans le modèle de la traduction statistique par segments :

$$c^* \approx \underbrace{\arg\max_c}_{\text{Décodeur}} \quad \underbrace{P(c)}_{\text{Modèle de langue}} \quad \max_I \prod_{i \in I} \quad \underbrace{P(s_i^I | c_i^I, B_{L_c \to L_s})}_{\text{Table de paraphrases}} \tag{4.6}$$

- un modèle de langue pour pouvoir estimer la naturalité de la paraphrase ;
- une table de paraphrases qui, pour un segment d'un ou plusieurs mots, propose des segments de paraphrases probabilisés. Cette table est la version monolingue de la table de traduction ;
- un décodeur qui permet de trouver une paraphrase le plus probable possible.

Les prochaines sections détaillent la sélection, les choix et les réalisations relatifs à ces trois composants. Nous nous attarderons sur la réalisation de la table de paraphrases car c'est l'aspect le plus différent par rapport au problème de la traduction statistique.

4.2 MODÈLE DE LANGUE

Un modèle de langue a pour objectif d'évaluer la naturalité d'une phrase. Plus formellement, dans le modèle de la paraphrase, il doit estimer la probabilité $P(c)$ que la phrase c apparaisse dans la langue.

Traditionnellement, la traduction statistique [Koehn et coll., 2007] ainsi que les générateurs de paraphrases [Quirk et coll., 2004] utilisent un modèle de langue n-grammes avec les heuristiques de lissage et de retrait. Nous présentons ici les principes d'un tel modèle de langue.

Un modèle de langue n-grammes définit la probabilité d'une phrase c comme le produit des probabilités conditionnelles des mots $m_1 m_2 m_3 \dots m_n$ qui la composent :

$$\begin{aligned} P(c) = \quad & P(m_1) \times P(m_2|m_1) \times P(m_3|m_1 m_2) \times \dots \\ & \times P(m_n|m_1 m_2 m_3 \dots m_{n-1}) \end{aligned} \tag{4.7}$$

où m_i est le i^e mot de la phrase c.

Afin de rendre cette quantité effectivement mesurable, la mémoire du système est bornée. L'hypothèse est que l'apparition d'un mot n'est que marginalement conditionnée par les mots de la phrase très éloignés. Par exemple, en limitant la mémoire à un seul mot, la formule devient :

$$P(c) = P(m_1) \times P(m_2|m_1) \times P(m_3|m_2) \times \dots \times P(m_n|m_{n-1}) \tag{4.8}$$

Ce modèle peut être estimé simplement à partir de l'observation des occurrences des séquences de n mots dans un corpus de texte. On parle d'un modèle n-grammes pour un tel modèle de langue avec une mémoire de $n-1$ mots. Notons qu'un modèle n-grammes est une chaîne de Markov d'ordre $n-1$. Une première approche pour apprendre un tel modèle consiste à estimer chaque probabilité par l'estimation du maximum de vraisemblance lors de l'observation d'un corpus d'apprentissage textuel. Ainsi :

$$P(m_i|m_{i-n+1} \dots m_{i-1}) \approx \tilde{P}(m_i|m_{i-n+1} \dots m_{i-1}) = \frac{C(m_{i-n+1} \dots m_{i-1} m_i)}{C(m_{i-n+1} \dots m_{i-1})} \tag{4.9}$$

où $C(m)$ est le nombre d'apparitions de la séquence m dans le corpus d'apprentissage.

La fréquence d'apparition des n-grammes dans un texte suit approximativement une loi de Zipf [Zipf, 1935], c'est-à-dire que la fréquence d'un n-gramme est approximativement inversement proportionnelle à son rang dans la table des fréquences. Ainsi, même avec un corpus d'apprentissage important, un grand nombre de n-grammes n'apparaissent qu'une fois seulement. Un nombre encore plus important de n-grammes sont absents du corpus, même si ce sont des séquences possibles. De ce fait, le maximum de vraisemblance n'est pas adéquat pour estimer la probabilité de ces n-grammes rares. C'est pourquoi il est nécessaire d'utiliser un modèle plus complexe, en introduisant par exemple un lissage.

Une méthode répandue est celle du modèle de retrait avec lissage proposée par Katz [1987]. L'objectif de ce modèle est double :

- par le lissage : redistribuer une partie de la masse de probabilité des séquences très fréquentes aux n-grammes rares. Katz [1987] utilisent le lissage de Good-Turing [Good, 1953]
- par le retrait : définir la probabilité $P(m_i|m_{i-n+1} \dots m_{i-1})$ d'un n-gramme $m_{i-n+1} \dots m_i$ absent du corpus, en fonction du $n-1$-gramme $m_{i-n} \dots m_i$. L'idée est de se permettre, en retirant le dernier mot du n-gramme manquant, de réduire la taille de la mémoire du modèle en échange d'une pénalité, provenant de la masse des probabilités récupérées par le lissage.

L'avantage des modèles de langue n-grammes est qu'ils sont simples à calculer, rapides et ne nécessitent qu'un corpus textuel pour l'apprentissage pour fournir des performances correctes. Enfin, le modèle est incrémental et requiert un historique

	⋮	
-3.838601	merci	-2.072343
	⋮	
-2.046351	merci au	-0.8129876
-2.069833	merci aussi	-0.8884156
-2.480007	merci aux	-0.1550518
-0.4556777	merci beaucoup	-1.724925
	⋮	
-2.734632	merci beaucoup au	-0.2044758
-2.232605	merci beaucoup d'	-1.107634
-1.841399	merci beaucoup de	-0.798173
-2.232605	merci beaucoup et	-0.2287284
-3.016443	merci beaucoup madame	0.03154036
-2.346549	merci beaucoup monsieur	0.07918119
-1.118662	merci beaucoup pour	-0.739033
	⋮	

TABLEAU 4.1: Extrait d'un modèle de langue selon le format de SRILM. La première colonne correspond au logarithme de la probabilité du n-gramme, la seconde colonne contient le n-gramme et la dernière colonne correspond à la pénalité de retrait en cas d'utilisation de ce n-gramme lors du calcul d'un n-gramme plus grand mais absent de la table.

borné : à partir de la probabilité d'une sous-phrase, des $n-1$ derniers mots qui la composent et d'un nouveau mot, il est possible de calculer la probabilité d'une sous-phrase plus grande. Ceci permet l'utilisation d'algorithmes de décodage fondés sur la programmation dynamique comme détaillé dans la section 4.4. En revanche, leur simplicité fait aussi leur plus grande lacune. En effet, l'hypothèse fondamentale de limitation de la mémoire fait que ces modèles ne sont pas en mesure de capturer les dépendances entre les mots éloignés dans une phrase.

Il existe d'autres modèles de langue plus complexes. Par exemple, le domaine de la traduction statistique utilise aussi des modèles fondés sur des arbres syntaxiques [Charniak et coll., 2003; Chiang, 2005]. Cependant ces modèles nécessitent généralement, pour l'apprentissage, des ressources annotées plus difficile à construire.

SRILM [Stolcke, 2002] est un modèle de langue par segments performant et librement disponible proposant par défaut le lissage et le retrait de Katz [1987]. Le générateur de paraphrases présenté dans ce chapitre utilise un modèle de langue 5-grammes appris sur le corpus français d'entraînement de la section 3.4 au moyen de SRILM. Un extrait du modèle de langue est présenté dans le tableau 4.1.

⋮
merci ||| miséricorde ||| 0.571429
merci ||| la miséricorde ||| 0.571429
merci ||| eleison ||| 0.571429
merci ||| remerciements que j' ai ||| 0.333333
merci ||| remerciements du ||| 0.333333
merci ||| et grâce à vous ||| 0.333333
⋮

TABLEAU 4.2: Extrait d'une table de paraphrases selon le format de MOSES. Les lignes sont de la forme « s_i^I ||| c_i^I ||| $P(s_i^I|c_i^I, B_{L_s \to L_c})$ ».

4.3 TABLE DE PARAPHRASES

La table de paraphrases associe à des segments de phrases un ensemble de segments probabilisés. Elle peut être vue comme un modèle de l'éloignement sémantique entre les segments de phrases. C'est le contenu de cette table qui fait la principale différence entre un système de traduction statistique par segments et un générateur statistique de paraphrases. Le tableau 4.2 présente un extrait d'une table de paraphrases formaté pour le décodeur MOSES [Koehn et coll., 2007].

La table de paraphrases est une version monolingue de la table de traduction. Les tables de traduction sont apprises à partir de corpus bilingues où les phrases sont alignées entre les deux langues. Le tableau 4.3 présente un extrait d'un tel corpus. Un aligneur sous-phrastique réalise dessus des alignements entre segments et leur comptage permet de construire la table probabilisée comme nous le présentons dans la section 4.3.1.

Nous allons maintenant discuter de la constitution de ces tables dans le cadre de la production de paraphrases. Dans un premier temps nous détaillerons le choix de l'aligneur dans la section 4.3.1. Nous traiterons dans la section 4.3.2 du problème du corpus d'apprentissage. Enfin, nous présenterons et explorerons une méthode de construction de table de paraphrases par langue pivot dans la section 4.3.3.

4.3.1 *Aligneur sous-phrastique*

L'aligneur sous-phrastique est l'un des piliers de la traduction statistique. Sa fonction est de produire la table de traduction à partir d'un corpus bilingue aligné au niveau de la phrase. C'est à partir d'une étude statistique des co-occurrences de segments dans le corpus que l'aligneur est capable de modéliser, par des relations probabilisées, les liens sémantiques entre les deux langues.

Un alignement est une fonction qui, pour chaque mot d'une phrase, associe un ou plusieurs mots de sa traduction. Le principe de l'alignement sous-phrastique statistique consiste à produire les alignements dans chaque couple de phrases du corpus d'apprentissage. Ensuite, à partir de ces alignements, il est possible de

⋮	
Reprise de la session Je déclare reprise la session du Parlement européen qui avait été interrompue le vendredi 17 décembre dernier et je vous renouvelle tous mes vux en espérant que vous avez passé de bonnes vacances. Comme vous avez pu le constater, le grand "bogue de l'an 2000" ne s'est pas produit. En revanche, les citoyens d'un certain nombre de nos pays ont été victimes de catastrophes naturelles qui ont vraiment été terribles. Vous avez souhaité un débat à ce sujet dans les prochains jours, au cours de cette période de session. En attendant, je souhaiterais, comme un certain nombre de collègues me l'ont demandé, que nous observions une minute de silence pour toutes les victimes, des tempêtes notamment, dans les différents pays de l'Union européenne qui ont été touchés. Je vous invite à vous lever pour cette minute de silence. (Le Parlement, debout, observe une minute de silence)	Resumption of the session I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period. Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful. You have requested a debate on this subject in the course of the next few days, during this part-session. In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union. Please rise, then, for this minute' s silence. (The House rose and observed a minute' s silence)
⋮	

TABLEAU 4.3: Extrait du corpus bilingue aligné EUROPARL. On peut y constater des erreurs : par exemple, l'absence du « œ » dans « je vous renouvelle tous mes vux ».

calculer la probabilité d'association de deux segments, c_i^I et s_i^I, par l'estimation du maximum de vraisemblance :

$$P(c_i^I | s_i^I, B_{L_s \to L_c}) = \frac{C(a(s_i^I) = c_i^I)}{C(s_i^I)} \tag{4.10}$$

où $C(a(s_i^I) = c_i^I)$ est le nombre de fois où le segment source s_i^I est aligné avec le segment cible c_i^I et $C(s_i^I)$ le nombre d'apparitions du segment s dans le corpus. Le problème consiste donc à obtenir les alignements nécessaires pour estimer les paramètres du modèle de production.

Une approche consiste à apprécier les alignements et les paramètres du modèle conjointement et à améliorer les estimations itérativement. En effet, connaître les alignements permet d'estimer les probabilités du modèle de production. Mais, à l'inverse, à partir d'un modèle de production, il est possible de calculer les alignements les plus probables.

Ce problème est résolu à l'aide d'un algorithme de type estimation-maximisation (EM). L'algorithme commence avec un modèle de production, comme celui présenté à la section 4.1 par exemple, où les probabilités de la table de traduction sont initialisées – avec une distribution uniforme par exemple. Pour chaque couple de phrases du corpus d'apprentissage, l'algorithme cherche l'alignement qui maximise le modèle de production. Ensuite, il modifie les estimations des probabilités de la table de traduction – par estimation du maximum de vraisemblance – en comptant les alignements sous-phrastiques. L'algorithme itère ces deux dernières étapes jusqu'à convergence des probabilités. Cette approche et certains de ses développements sont plus amplement décrits dans les travaux de Birch et Koehn [2010]; Och et Ney [2003] et de Cromières [2010]. Le programme GIZA++ [Och et Ney, 2003] est un des aligneurs de référence reposant sur ces principes. Il est librement disponible.

D'autres types d'aligneurs sous-phrastiques sont accessibles, comme par exemple l'aligneur basse fréquence ANYMALIGN proposé par Lardilleux et Lepage [2008]. Cet aligneur repose sur un principe très simple : si à chaque fois qu'un segment donné est présent dans une phrase, un autre segment est lui aussi présent systématiquement dans la traduction et uniquement dans ces phrases, alors les deux segments sont certainement des traductions l'un de l'autre. La probabilité de trouver un alignement parfait est presque nulle pour un corpus suffisamment grand et varié. C'est pourquoi ANYMALIGN réalise des échantillonnages aléatoires du corpus et recherche dans ces sous-corpus des alignements parfaits. À chaque itération, l'algorithme acquière des alignements et, sachant le nombre de fois où ces alignements ont été trouvés parmi toutes les itérations, le programme estime leur probabilité.

Nous avons montré dans Lardilleux et coll. [2009] qu'ANYMALIGN produit des alignements de 1-gramme plus nombreux et de meilleure qualité que GIZA++. En revanche, l'algorithme est moins apte à produire des alignements pour des segments plus longs. En effet, il est beaucoup plus difficile de trouver un alignement parfait alors que les mots qui composent un segment ont des fréquences différentes. Or, la qualité de la traduction – et donc de la paraphrase – s'améliore lorsque les segments les plus longs possibles sont utilisés. Ainsi, à l'extrême, lorsqu'il existe un segment dans la table qui correspond à la phrase entière, la traduction est très probablement correcte.

Les tables de paraphrases pour le système décrit dans ce chapitre seront produites à l'aide de GIZA++, avec les paramètres par défaut.

4.3.2 *Corpus d'apprentissage*

Comme nous l'avons vu dans la section 4.3.1, l'aligneur sous-phrastique produit la table de traduction à partir d'un corpus bilingue aligné au niveau des phrases.

Dans le domaine de la traduction statistique par segments, ce sont des corpus de plusieurs millions de mots qui sont utilisés, comme EUROPARL [Koehn, 2005]. Dans ce domaine, la taille du corpus influe sur les performances du traducteur. Il faudrait alors des corpus de plusieurs centaines de milliers de couples de paraphrases pour utiliser les outils de la traduction statistique par segments.

La traduction est une activité humaine fréquente et fortement enregistrée. Il est ainsi relativement aisé de disposer de corpus de traduction pour de nombreuses paires de langues et en grande quantité. Par exemple, le corpus EUROPARL est constitué de débats au parlement européen. Ces débats sont traduits systématiquement dans les multiples langues de la communauté européenne.

À l'inverse, la paraphrase n'est pas une activité aussi institutionnalisée et dans les domaines où elle apparaît fréquemment – comme le journalisme par exemple – elle n'est pas enregistrée ou pas directement disponible. La constitution de corpus de paraphrases est un problème difficile qui limite celui de la production de paraphrases. De fait, il existe peu de corpus de paraphrases et l'acquisition automatique ou semi-automatique de tels corpus reste un domaine de recherche ouvert.

Pour construire de telles ressources, Barzilay et McKeown [2001] proposent d'utiliser plusieurs traductions d'un même ouvrage effectuées par des traducteurs différents. Il s'agit fréquemment d'ouvrages « classiques » qui font l'objet de réédition au fil du temps ou des traductions pour différents pays anglophones. Ils utilisent, entre autres, plusieurs éditions de *Madame Bovary* de Flaubert, des contes d'Andersen et *Vingt Mille Lieues sous les mers* de Verne comme corpus de paraphrases quasi-alignés au niveau des phrases. Malheureusement, les auteurs constatent que l'interprétation personnelle des traducteurs fait que les multiples traductions d'un texte entier ne permettent pas de systématiquement recréer un corpus parallèle aligné au niveau des phrases En fait, ce type de ressources correspond plus à une collection de textes comparables qu'à un corpus monolingue aligné au niveau des phrases. De telles ressources ne sont pas directement exploitables par les outils classiques d'alignement et de traduction automatique statistique.

Dans le même but, Dolan et Brockett [2005] utilisent des articles de journaux relatifs au même évènement et publiés à la même date pour en extraire automatiquement des paraphrases. Ils utilisent un outil de classification automatique pour extraire les paraphrases candidates. À partir de leurs travaux, un corpus de près de 6 000 paraphrases, vérifié manuellement a été mis à la disposition de la communauté de recherche.

D'autres méthodes reposent sur une heuristique bien connue qui consiste à dire que si deux phrases ont la même traduction, alors ce sont des paraphrases l'une de l'autre. Ainsi, Lepage et Denoual [2005] construisent un corpus d'approximativement 160 000 paraphrases à partir d'un corpus d'évaluation de traduction

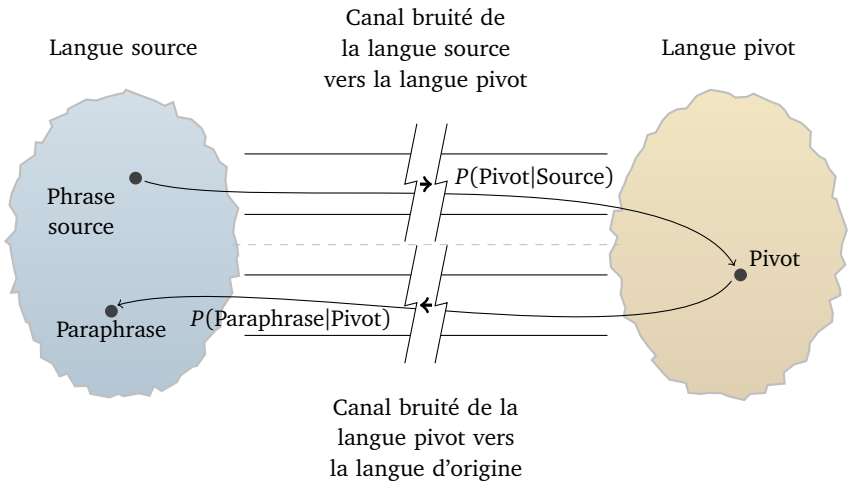


FIGURE 4.2: La production de paraphrases par langue pivot.

automatique. Ce corpus est agrandi par l'ajout de 15 % de nouvelles paraphrases produites par analogie. Ce corpus n'est pas librement disponible.

À partir de la même heuristique, Bouamor et coll. [2010] demandent à plusieurs personnes de traduire vers la langue de paraphrases, plusieurs phrases dans d'autres langues. Ils ont ainsi construit un corpus d'un millier de paraphrases. Ce corpus est librement disponible.

À notre connaissance, il n'existe pas à ce jour de corpus de paraphrases disponible de taille suffisamment importante pour permettre l'utilisation d'un aligneur statistique.

4.3.3 *Table de paraphrases par langue pivot*

Parallèlement à cette recherche de corpus, d'autres méthodes de constitution de table de paraphrases ont été proposées. C'est le cas, par exemple, de la méthode par langue pivot proposée par Bannard et Callison-Burch [2005]. Nous allons maintenant détailler cette méthode qui utilise un corpus bilingue aligné, ressource plus facilement disponible, pour produire une table de paraphrases.

Dans cette approche, la relation de paraphrase, conformément au modèle du canal bruité, est vue au travers d'une langue pivot. La figure 4.2 illustre ce modèle. Notons que le canal bruité entre la langue source et la langue pivot n'est pas nécessairement symétrique. Nous le décomposons donc en deux canaux distincts $B_{L_s \to L_p}$ et $B_{L_p \to L_s}$.

Suivant ce modèle, les probabilités de transformation de segments de la table de paraphrases sont égales à la probabilité de produire le segment paraphrase, sachant le segment source et sachant que l'on passe par une langue pivot. Cette probabilité se décompose en une somme de tous les « chemins » partant du segment source, passant par un pivot et retournant au segment paraphrase. Plus formellement, la

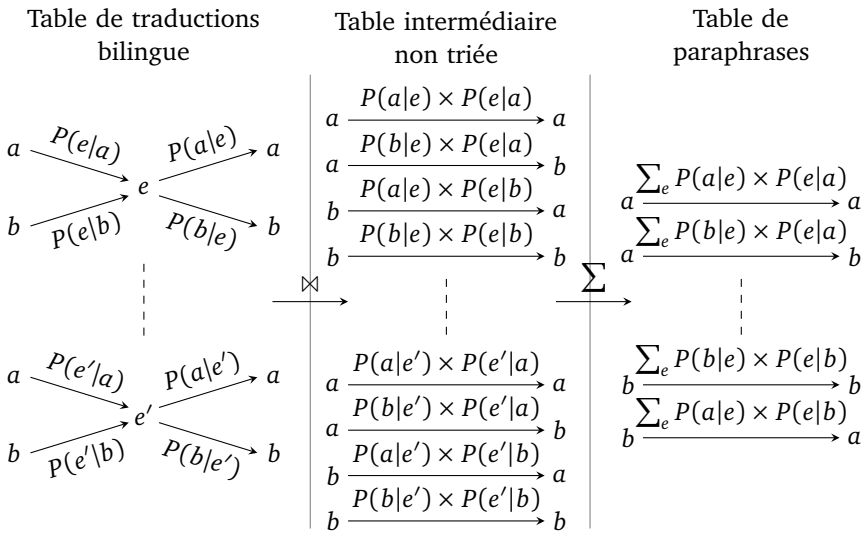


FIGURE 4.3: Production d'une table de paraphrases par auto-jointure d'une table de traduction bilingue. Pour simplifier le schéma, les canaux bruités $B_{L_s \to L_p}$ et $B_{L_p \to L_s}$ sont volontairement omis des formules de probabilité.

probabilité de « traduction » d'un segment c_i^I en un segment s_i^I, utilisée dans le modèle décrit section 4.1, est donc décomposée ainsi :

$$\begin{aligned} P(s_i^I | c_i^I, B_{L_s \to L_s}) &\equiv P(s_i^I | c_i^I, B_{L_s \to L_p}, B_{L_p \to L_s}) \\ &= \sum_{e \in L_p} P(s_i^I | e, B_{L_p \to L_s}) \times P(e | c_i^I, B_{L_s \to L_p}) \end{aligned} \tag{4.11}$$

Le calcul de cette table de paraphrases peut donc être réalisé à partir de deux tables modélisant les canaux $B_{L_p \to L_s}$ et $B_{L_s \to L_p}$. Contrairement à une double traduction, cette méthode n'utilise pas de modèle de langue pour la langue pivot. Le choix d'un segment pour une phrase donnée n'est réalisé qu'une seule fois et le système est donc moins pénalisé par d'éventuelles erreurs de traduction dans la langue pivot.

La construction de la table de paraphrases à partir de cette méthode est très simple. Comme les aligneurs sous-phrastiques calculent déjà les deux modèles simultanément, il suffit d'une table de traduction bilingue. La table de traduction permet d'associer deux probabilités à chaque segment a pour chaque pivot e : $P(a|e, B_{L_p \to L_s})$ et $P(e|a, B_{L_s \to L_p})$. En triant cette table en fonction des pivots, l'algorithme n'a plus qu'à parcourir les ensembles de segments associés au même pivot. Pour chaque couple a, b de segments d'un de ces ensembles, le programme produit une entrée dans une table intermédiaire avec comme probabilité associée $P(a|e, B_{L_p \to L_s}) \times P(e|b, B_{L_s \to L_p})$. Ces opérations réalisent en fait une *auto-jointure* de la table de paraphrases sur les pivots. Notons que pour un ensemble de n segments liés à un même pivot, l'auto-jointure ajoute n^2 entrées dans la table intermédiaire. Pour produire la table de paraphrases, le programme n'a plus qu'à faire la somme de toutes les entrées de la table intermédiaire qui sont composées des mêmes segments. L'ensemble de ce processus est illustré dans la figure 4.3.

pivot	nombre de segments liés au pivot
is	7 251
it is	4 692
to	4 495
are	4 336
have	4 206
be	3 895
that	3 789
it	3 755
will	3 656
the	3 640

TABLEAU 4.4: Les 10 pivots ayant le plus de liens dans la table de traduction anglais-français.

En utilisant cette méthode, la construction d'une table de paraphrases à partir d'un corpus bilingue aligné de grande taille devient problématique. En effet, à partir du corpus français-anglais présenté dans la section 3.4, les outils d'alignements sous-phrastiques de la section 4.3.1 produisent une table de traduction bilingue de plus de 53 millions d'entrées qui avoisine les 7 Go. L'auto-jointure d'une telle table produirait une table intermédiaire de 1,6 milliards d'entrées pour une taille finale estimée de 900 Go environ. Non seulement une table d'une telle taille est difficilement manipulable mais toute l'information qu'elle contient n'est certainement pas pertinente. En particulier, certains alignements proviennent d'erreurs introduites par l'outil d'alignement sous-phrastique ou d'erreurs dans le corpus d'apprentissage.

Afin de réduire la taille de la table, plusieurs heuristiques sont envisageables. Une première heuristique consiste à supprimer les entrées de la table intermédiaire qui ont une probabilité inférieure à un seuil donné ϵ. L'objectif est de supprimer les entrées qui ne contribueront pas beaucoup à la probabilité finale dans la table de paraphrases. Pour l'ensemble de nos expériences, ϵ est fixé empiriquement à 10^{-5}. Cette première étape reste toutefois insuffisante pour réduire suffisamment la table intermédiaire.

Puisqu'un ensemble de n segments liés à un même pivot ajoute n^2 entrées dans la table intermédiaire, c'est la taille de ces ensembles qui détermine la taille de la table intermédiaire. La distribution des ensembles de segments liés à un même pivot est présentée à la figure 4.4. Notons le lien inversement exponentiel entre le nombre d'ensembles de même taille et leur taille. Par exemple, l'ensemble composé de 7 251 segments, mentionné dans le tableau 4.4, produira plus de 52 millions d'entrées dans la table intermédiaire. Le pivot associé à cet ensemble est *is* [2]. Comme le montre le tableau 4.4 les plus grands ensembles sont liés à des mots « outils » très fréquents. De la même façon, l'ensemble lié au pivot « , » se trouve en 20^{e} position avec 2 691 segments.

2. Le verbe «être» en anglais, conjugué à la troisième personne du singulier.

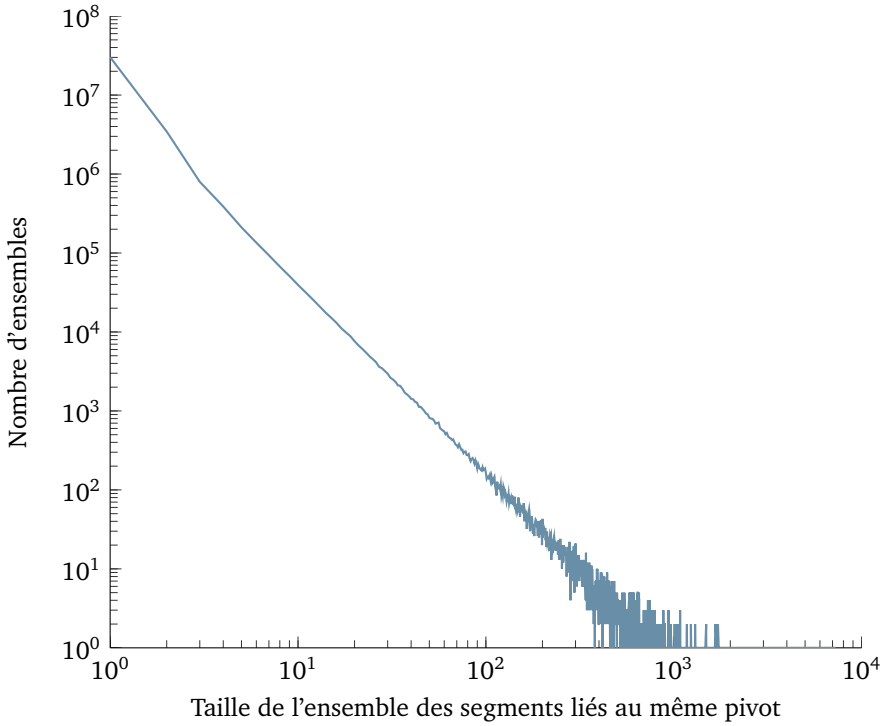


FIGURE 4.4: Le nombre d'ensembles de même taille est inversement exponentiel à la taille des ensembles des segments associés à un même pivot.

Évidemment, *is* ne peut pas avoir 7 251 traductions correctes en français or cet ensemble ajoute 52 577 001 entrées dans la table intermédiaire. Cela est encore plus vrai pour la ponctuation. Malheureusement, les aligneurs sous-phrastiques sont très sensibles à la fréquence d'un mot et réalisent souvent des erreurs d'alignement pour les mots très fréquents. Cette analyse nous conduit à introduire une seconde heuristique qui consiste à supprimer les ensembles composés d'un nombre de segments supérieur à un seuil donné τ. L'objectif est d'éviter de perturber les probabilités issues d'ensembles beaucoup plus petits qui ont plus de chance d'être correctes. Le seuil τ est fixé empiriquement à 200.

Enfin, la table de paraphrases finale peut contenir un très grand nombre de paraphrases pour un même segment source. Nous ne conservons dans la table finale que les η paraphrases les plus probables. Le paramètre η est fixé empiriquement à 20. Avec cette troisième heuristique, la table apprise sur EUROPARL est réduite de 1,6 milliards d'entrées à 116 millions.

La paraphrase, telle que définie dans la section 2.2, peut conserver des mots inchangés mais doit comporter au minimum une modification. Cet aspect diffère de travaux précédents sur la production de paraphrases [Bannard et Callison-Burch, 2005; Max, 2008] qui imposent le lieu des modifications. La reformulation d'un segment en lui-même, appelé *paraphrase identité*, est une « transformation » localement valide. Les paraphrases identités ne sont donc pas supprimées et représentent tout de même 33% de la table finale.

Grâce à cette approche, nous sommes donc en mesure de produire une table de paraphrases qui contient un grand nombre d'alternatives et qui couvre un important vocabulaire. Il est évident que cette table comporte énormément de bruit – beaucoup de segments n'ont pas vingt paraphrases valides. Mais avant de continuer le filtrage de cette table, il nous semble préférable d'étudier les sorties d'un générateur de paraphrases afin de guider les éventuelles corrections à réaliser.

4.4 DÉCODEUR

Conformément au modèle décrit dans la section 4.1, nous disposons maintenant de deux des trois éléments nécessaires pour construire un générateur statistique de paraphrases : le modèle de langue et la table de paraphrases. D'après le modèle du canal bruité, le dernier élément est un décodeur qui va rechercher et construire la meilleure paraphrase. Nous présentons dans cette section les principes de fonctionnement d'un décodeur fondé sur l'algorithme de Viterbi [1967] couplé avec une recherche par faisceau [Tillmann et Ney, 2003]. Ce type de décodeur est très utilisé en traduction automatique mais aussi en production de paraphrases [Quirk et coll., 2004; Zhao et coll., 2009; Cahill et coll., 2009].

Pour une phrase donnée, la table de paraphrases fournit plusieurs entrées correspondantes qui sont appelées *options de traduction*. L'ordre sur les mots de la phrase source réalise un ordre partiel sur les options de traduction. Le décodage est donc réalisé dans un ordre particulier, le plus souvent de gauche à droite. L'ensemble des options de traduction et l'ordre partiel associé forme un treillis de paraphrases. L'objectif du décodeur est de trouver le chemin dans ce treillis qui minimise la fonction de score du modèle de la paraphrase.

Suite aux hypothèses présentées dans la section 4.1, la fonction de score est incrémentale. La notion d'incrémentalité signifie qu'à partir du score de n'importe quelle « traduction » partielle il est possible de calculer le score d'une « traduction » plus complète où un segment de plus a été « traduit ». De plus, l'historique nécessaire pour calculer le score d'une « traduction » plus complète à partir d'une « traduction » partielle est limité par la taille de la plus grande entrée de la table de paraphrases– au sens du nombre de mots couverts – et par l'ordre du modèle de langue.

Grâce à ces propriétés d'incrémentalité et de limitation de l'historique, le problème de décodage revient à chercher l'ensemble d'états le plus probable *a posteriori* sachant une suite d'observations dans un modèle de Markov. L'ordre k du modèle de Markov est le maximum entre l'ordre du modèle de langue et la longueur du plus long support parmi les options de traduction. Ce problème peut être résolu grâce à un algorithme de programmation dynamique comme celui de Viterbi.

L'algorithme consiste à explorer le treillis en entier mais il utilise les propriétés de la fonction de calcul pour réduire le temps de calcul et l'occupation mémoire. Nous allons maintenant détailler une itération de l'algorithme.

Supposons que le programme ait déjà calculé $n-1$ itérations, c'est-à-dire qu'il a réalisé toutes les « traductions » possibles jusqu'au $n-1^{e}$ mot. Le programme dispose de la liste de toutes les suites de k mots possibles qui soient suffixes de la partie « traduite » dans les « traductions » de longueur $n-1$ – où k est l'ordre du modèle de Markov. Pour chaque élément de cette liste, le chemin de score maximal – et le score associé – menant à cet historique est conservé.

Pour calculer l'itération n, le programme va tester l'ensemble des options de traduction qui couvre le n^{e} mot de la phrase source. Chaque option conduit à un nouvel historique où il est possible de calculer un nouveau score grâce à l'état de départ, l'historique associé à cet état, l'option de traduction choisie et les modifications qu'a entraînées l'option de traduction. Notons que c'est la propriété d'incrémentalité et de limitation de l'historique qui permet ce calcul à partir de connaissances réduites. Si deux options différentes conduisent au même historique, seule celle avec le score le plus élevé est conservée. Lorsqu'un historique de l'itération $n-1$ a été utilisé pour l'itération n, il n'est plus nécessaire de le conserver. Ce sont ces deux heuristiques qui permettent de limiter la taille des états conservés en mémoire.

Une fois que l'algorithme a été itéré sur tous les mots de la phrase source, le résultat est la solution associée à l'état de score maximal parmi les derniers historiques.

Grâce à la dépendance locale et l'incrémentalité de la fonction de score, le problème est polynomial en la longueur de la phrase. Notons que c'est grâce à l'absence de modèle de réordonnancement que, contrairement à la traduction statistique, le problème de production de paraphrases n'est pas NP-difficile [Knight, 1999].

Malgré la faible complexité algorithmique, l'importante taille de la table de paraphrases fait que le décodage n'est pas toujours réalisable dans des conditions de temps et d'occupation mémoire raisonnables. Afin de simplifier le décodage, l'heuristique de la recherche par faisceau est souvent utilisée. Elle consiste à supprimer de la liste des historiques les états qui ont le moins de chances de conduire à la solution optimale. Il existe classiquement deux façons de réduire la taille de la liste des historiques :

– supprimer les états avec un score inférieur à ϵ fois le score de la meilleure solution ;
– borner la taille de la liste et ne conserver que les meilleurs états.

Contrairement aux heuristiques de fusion d'états, ces heuristiques ne garantissent plus l'obtention d'une solution optimale. En revanche, elles permettent de réduire très fortement le temps de calcul et l'occupation mémoire nécessaires.

Il est possible que la meilleure paraphrase soit la phrase source. Pour éviter ce cas de figure, on peut chercher le meilleur chemin différent de la phrase source, ce qui se fait en produisant les deux meilleures paraphrases et en conservant celle qui est différente de la phrase source. Pour ce faire, il suffit de conditionner la fusion d'états pour toujours conserver les deux meilleures alternatives permettant d'atteindre un état appartenant au meilleur chemin.

Le décodage fondé sur l'algorithme de Viterbi permet d'obtenir une solution à faible coût calculatoire. En revanche, il force à faire certaines hypothèses sur le modèle de la paraphrase afin d'avoir une fonction incrémentale. Nous utilisons un décodeur reposant sur ces principes, librement disponible et qui est une référence dans le domaine de la traduction statistique : MOSES [Koehn et coll., 2007].

4.5 MISE AU POINT DES PARAMÈTRES

La mise au point des paramètres permet d'améliorer significativement les performances des traducteurs statistiques [Och, 2003]. Cette procédure consiste à pondérer les différentes composantes du modèle présenté dans la section 4.1 – le modèle de langue et la table de paraphrases – et à optimiser ces poids pour minimiser l'erreur du système par rapport à une mesure externe. Le modèle s'écrit donc ainsi :

$$c^* \approx \arg\max_c P(c)^{\alpha_1} \left(\max_I \prod_{i \in I} P(s_i^I | c_i^I, B_{L_c \to L_s}) \right)^{\alpha_2} \tag{4.12}$$

Afin de simplifier le problème d'optimisation, l'écriture linéarisée par la fonction logarithme est souvent utilisée :

$$c^* \approx \arg\max_c \max_I \left(\alpha'_1 \log(P(c)) + \alpha'_2 \left(\sum_{i \in I} \log \left(P(s_i^I | c_i^I, B_{L_c \to L_s}) \right) \right) \right) \tag{4.13}$$

Dans la pratique, un corpus de développement est construit à partir d'un ensemble de phrases à traduire. Un ensemble de traductions de référence est associé à chacune de ces phrases. La mise au point consiste à tenter de traduire l'ensemble des phrases du corpus de développement et à mesurer l'erreur entre la traduction proposée par le système et les références à l'aide d'une mesure externe automatique de performance, comme BLEU [Papineni et coll., 2002] ou NIST [Zhang et coll., 2004]. Le système cherche ensuite à optimiser les différentes pondérations pour réduire la mesure de l'erreur, grâce à une méthode d'optimisation comme la descente de gradient par exemple [Press et coll., 1993].

Voyons maintenant si ce protocole reste valable pour la production de paraphrases, ainsi que les alternatives possibles.

Comme nous l'avons écrit dans la section 4.3.2, il existe peu de corpus de paraphrases. De plus, les travaux de Lin et Pantel [2001] ont montré qu'il est

difficile de construire un corpus de paraphrases de référence. En effet, un système de production est capable de proposer des paraphrases valides très différentes de ce qui est présent dans les corpus de référence construits manuellement.

La seule référence simple à obtenir est la phrase source. Or, en utilisant uniquement cette dernière comme référence, les mesures comme BLEU ou NIST sont corrélées avec la distance d'édition. Par exemple, nous utilisons le système présenté dans ce chapitre, sans pondérer le modèle, pour produire des paraphrases du jeu TEST 2 – présenté dans la section 3.4. En comparant les paraphrases aux phrases sources à l'aide des mesures BLEU, NIST et la distance d'édition, nous obtenons les distributions présentées à la figure 4.5 – après avoir centré et réduit les scores. Nous constatons que la corrélation entre BLEU et NIST est de 0,95 (p-valeur $< 10^{-3}$). Les mesures BLEU et NIST sont anti-corrélées avec la distance d'édition respectivement de $-0{,}67$ et $-0{,}71$ (p-valeur $< 10^{-3}$).

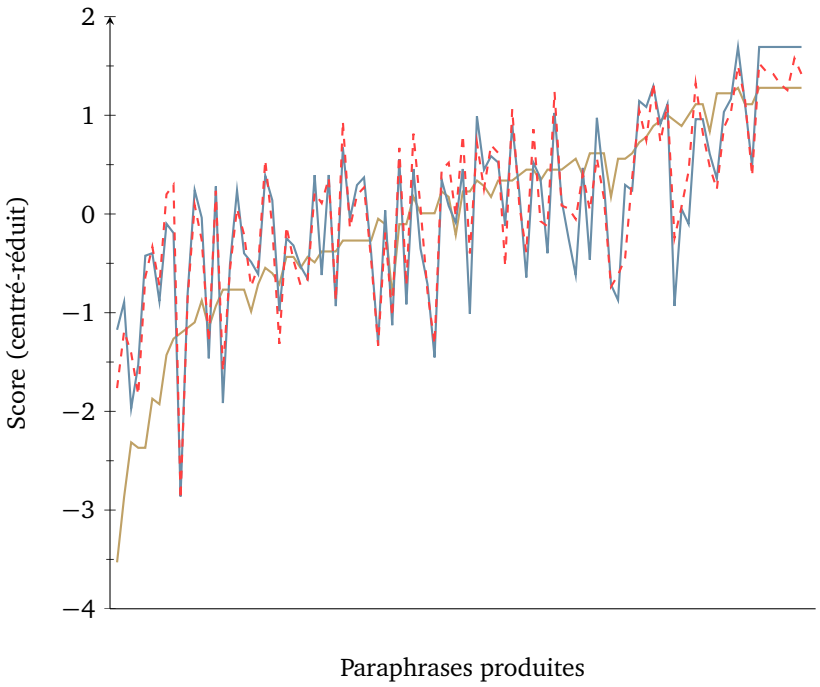


FIGURE 4.5: Comparaison entre les paraphrases et les phrases sources du jeu TEST 2. En beige, l'opposé de la distance d'édition ; la mesure BLEU en bleu ; la mesure NIST en rouge. Les trois mesures suivent la même tendance.

C'est pourquoi l'utilisation d'un algorithme de mise au point des paramètres conduit à affecter toute la pondération à la table de paraphrases et à réduire au maximum l'impact du modèle de langue dans celui de la paraphrase. En effet, parmi les entrées de la table de paraphrases portant sur le même segment, l'entrée « identité » – qui ne modifie rien – est l'entrée avec la plus forte probabilité. La solution la plus efficace pour minimiser la distance d'édition – et donc maximiser BLEU ou NIST – consiste donc à utiliser uniquement la table de paraphrases.

Comme nous venons de le montrer, l'absence de référence fait que le protocole d'optimisation des paramètres de la traduction statistique n'est pas directement utilisable pour la production de paraphrases.

Zhao et coll. [2008a] proposent une approche innovante pour pouvoir réaliser la mise au point des paramètres. À partir d'un corpus de développement, ils calculent toutes les paraphrases qui peuvent être produites avec une seule option de traduction. Ces paraphrases sont alors évaluées manuellement afin de déterminer les transformations valides et les transformations invalides en contexte – au sens de la syntaxe et de la conservation du sens. La mesure de référence utilisée pour la mise au point est alors :

$$\mathrm{PSER}(c,s) = \frac{\left|\mathrm{PS}_0(c,s)\right|}{|\mathrm{PS}(c,s)|} \tag{4.14}$$

où c est la paraphrase, s la phrase source, $\mathrm{PS}(c,s)$ l'ensemble des segments, autres que l'identité, utilisé pour produire c. Enfin, $\mathrm{PS}_0(c,s)$ est l'ensemble des segments jugés manuellement, et indépendamment les uns des autres, incorrects. L'objectif de l'algorithme d'optimisation est alors d'adapter les paramètres du modèle pour réduire le $\mathrm{PSER}(c,s)$ moyen sur le corpus de développement.

Bien que cette approche semble très prometteuse, l'effort nécessaire à la construction du corpus de développement est extrêmement important, surtout pour une table de paraphrases conséquente. Nous n'avons donc pas retenu cette approche pour régler les paramètres du système.

Face à si peu de solutions, nous avons choisi de garder une fonction de score des paraphrases fidèle au modèle du canal bruité – tous les poids sont à 1. Ce choix correspond à celui qui est fait par Quirk et coll. [2004]; Bannard et Callison-Burch [2005] ou Max [2008]. Il reste que la mise au point d'une méthode simple pour réaliser le réglage des paramètres serait une avancée non négligeable pour améliorer les performances des générateurs statistiques de paraphrases.

4.6 CONCLUSION

Dans ce chapitre, nous avons présenté le modèle classique de la production statistique de paraphrases. À partir d'outils disponibles nous avons construit un générateur de paraphrases fondé sur l'apprentissage à partir de ressources disponibles en grande quantité : les corpus bilingues alignés.

Ce générateur est composé de trois éléments principaux :

- un modèle de langue 5-grammes utilisant les outils de SRILM avec les paramètres par défaut ;
- une table de paraphrases construite à partir d'un corpus bilingue français-anglais EUROPARL. L'alignement est réalisé par GIZA++ avec ses paramètres par défaut. La table de paraphrases est obtenue par auto-jointure de la table bilingue. Les trois paramètres pour les heuristiques de filtrage que nous introduisons sont $\epsilon = 10^{-5}$, $\tau = 200$ et $\eta = 20$ (voir la section 4.3.3 pour plus de détails) ;
- un décodeur fondé sur l'outil MOSES avec ses paramètres par défaut. À chaque composante du modèle nous associons le même poids. Le décodeur produit comme résultat les deux meilleures paraphrases et nous ne conservons que celle qui a le score le plus élevé mais différente de la phrase source.

Compte tenu des réflexions que nous avons menées dans ce chapitre, nous constatons que de nombreuses pistes d'améliorations sont déjà possibles, parmi lesquelles :

- le filtrage de la table de paraphrases ;
- le choix de la langue pivot ;
- la combinaison de plusieurs langues pivots ;
- la mise au point des paramètres du modèle.

Ce système nous semble représentatif de l'avancement actuel des recherches en production automatique de paraphrases. Il nous servira donc de référence dans nos expériences ultérieures. L'étude de son comportement et de ses résultats constitue la prochaine étape de nos travaux sur la paraphrase et est exposée dans le chapitre qui suit.

5

LES LIMITES DE LA PRODUCTION STATISTIQUE DE PARAPHRASES

Suite aux travaux présentés dans les chapitres précédents, nous disposons d'un générateur de paraphrases de référence ainsi que d'un protocole d'évaluation. L'étude des résultats de ce premier système va nous permettre d'analyser les forces et les faiblesses du modèle de la production statistique de paraphrases. Nous détaillerons plus précisément les problématiques liées aux différents aspects du système : l'intégration avec la synthèse vocale, le décodage et l'évaluation.

La section 5.1 présente et discute les performances du système de référence du chapitre 4. La section 5.2 propose et évalue le générateur statistique de paraphrases en association avec un système de synthèse vocale. La section 5.3 présente les limites d'un décodeur fondé sur l'algorithme de Viterbi, comme décrit section 4.4. La section 5.4 démontre par l'absurde les limites du modèle de la paraphrase associé à une évaluation focalisée uniquement sur la conservation du sens.

5.1 LIMITES DES PERFORMANCES

La première expérience que nous proposons pour mettre en évidence les limites des modèles présentés précédemment, consiste à évaluer, grâce à la plateforme décrite dans la section 3.2, le système de production statistique de paraphrases décrit au chapitre 4.

Dans cette expérience, deux évaluateurs francophones ont eu pour tâche d'évaluer les cent paraphrases produites à partir du jeu TEST 1 – voir section 3.4. Les résultats sont donnés dans le tableau 5.1, d'après le formalisme présenté dans la section 3.3.

Avec un coefficient Kappa de 0,69, ce qui est traditionnellement interprété comme un accord substantiel, nous constatons que les deux juges abordent l'évaluation de façon similaire. Ceci semble valider une évaluation à deux juges. Avec

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	49	8	48	10	65	11
oui	3	40	10	32	2	**22**
Kappa	**0,69** (p-valeur $< 10^{-3}$) **Accord substantiel**					

TABLEAU 5.1: Évaluation du générateur de paraphrases statistique de référence sur le jeu TEST 1.

seulement 22 paraphrases sur 100 jugées correctes, les performances du système restent trop faibles pour une utilisation pratique dans un système automatique.

5.1.1 *Analyse des paraphrases*

Nous analysons maintenant plusieurs paraphrases produites lors de l'expérience précédente. Les paraphrases sont dans le format présenté dans la section 3.3.

Analysons d'abord les paraphrases évaluées comme correctes par les juges. L'objectif de cette analyse est de déterminer le type de paraphrases que le système est capable de produire.

Une modification fréquente consiste à remplacer un mot par un synonyme comme pour la paraphrase 5.1 ci-dessous. On retrouve ce type de modification pour la paraphrase 5.2 où c'est un acronyme qui est décomposé. Enfin, la paraphrase 5.3 introduit un anglicisme d'origine latine probablement fréquent au parlement européen. Nous constatons, dans ce dernier exemple, l'influence du corpus d'apprentissage sur les relations d'équivalence sémantiques capturées dans la table de paraphrases.

Paraphrase 5.1 – *Utilisation d'un synonyme :*
O : La Commission soutient activement la mise au point au niveau mondial et de façon ***uniformisée*** *d'un cycle de tests des motocycles.*
P : La Commission soutient activement la mise au point au niveau mondial et de façon ***standardisée*** *d'un cycle de tests des motocycles.*

Paraphrase 5.2 – *Décomposition d'un acronyme :*
O : Madame le Président, tout a déjà été dit sur les ***PME****.*
P : Madame le Président, tout a déjà été dit sur les ***petites et moyennes entreprises****.*

Paraphrase 5.3 – *Introduction d'un anglicisme :*
O : [. . .] dans son ***rectificatif*** *très appréciable.*
P : [. . .] dans son ***corrigendum*** *très appréciable.*

Certaines modifications sont beaucoup plus réduites et ne portent que sur la suppression ou l'ajout d'une virgule comme dans la paraphrase 5.4 après *Mais la Commission*. Compte tenu de la conception de l'expérience, le système aurait pu se contenter de cette modification pour produire une paraphrase valide. Notons que, d'après la définition 2.5, l'ajout d'une virgule est une transformation suffisante pour que la phrase proposée soit considérée comme une paraphrase. Nous reviendrons sur ce problème de l'utilité d'une transformation dans la section 5.4.

Paraphrase 5.4 – *Suppression d'une virgule :*
O : Mais la Commission, non plus, n'a pas joué son rôle et ***cela est*** *grave, Monsieur le Président.*
P : Mais la Commission non plus, n'a pas joué son rôle et ***c'est un problème*** *grave, Monsieur le Président.*

Comme le montre la paraphrase 5.5, l'interversion locale de mots est un autre type de modification rencontré. Ces interversions sont possibles lorsque les mots

ne sont pas trop éloignés et restent dans la portée d'un segment de la table de paraphrases. Le système n'est pas capable de réaliser, de façon fiable, des inversions pour des mots très éloignés.

Paraphrase 5.5 – *Interversion de mots :*
O : Je ***vous remercie tous*** *et j'espère que nous nous reverrons.*
P : Je ***voudrais tous vous remercier*** *et j'espère que nous nous reverrons.*

Enfin, le système est capable de combiner plusieurs modifications. Ces modifications peuvent être séparées les unes des autres, comme pour la paraphrase 5.6. Elles peuvent aussi être les unes à côté des autres, comme pour la paraphrase 5.7, ce qui donne une impression de réécriture importante de la phrase d'origine.

Paraphrase 5.6 – *Plusieurs modifications disjointes :*
O : Nous devons demander aux Américains d'être plus actifs et ***d'essayer*** *de remettre les* ***populations de la région*** *sur* ***le droit chemin****.*
P : Nous devons demander aux Américains d'être plus actifs et ***de tenter*** *de remettre les* ***gens sur place*** *sur* ***la bonne voie****.*

Paraphrase 5.7 – *Plusieurs modifications en continu :*
O : Je ***voudrais également mettre en avant une préoccupation liée*** *à la question de l'immigration.*
P : Je ***souligne d'ailleurs l'inquiétude quant*** *à la question de l'immigration.*

Ces exemples donnent un aperçu des capacités de production d'un générateur statistique de paraphrases. Intéressons-nous maintenant à des paraphrases jugées incorrectes afin de déterminer les lacunes et limites du système.

Des problèmes d'accord ou de conjugaison sont fréquemment observés, comme illustré avec les paraphrases 5.8 et 5.9. Ce type de transformation est possible car la table de paraphrases est construite à partir d'une langue pivot. En passant par l'anglais, nous perdons, dans certains cas, les informations de genre, de nombre ou de temps. Par exemple, le verbe anglais au présent *have* est aligné, en français, avec *avez* – de *vous avez* – mais aussi *avons* – de *nous avons*. Ceci fait que la table de paraphrases, construite par auto-jointure d'une table de traduction, contient des entrées du type de celles présentées dans le tableau 5.2. On peut se demander si le choix de l'anglais comme langue pivot est effectivement judicieux.

Dans tous les cas, pour les paraphrases 5.8 et 5.9, le modèle de langue aurait dû être capable de corriger les erreurs. En effet, « examen » et « sectorielles » ne sont jamais à la suite dans le corpus d'apprentissage du modèle de langue, contrairement à « examen » et « sectoriel ». La pénalité pour cette transformation donnée par le modèle de langue est, semble-t-il, insuffisante. Ce type d'erreurs montre les limites d'un modèle de langue avec retrait, tel que présenté à la section 4.2.

Paraphrase 5.8 – *Problème d'accord en genre et en nombre :*
O : Et, quoi qu'il en soit, cela ***exige*** *un examen* ***sectoriel*** *préalable.*
*P : * Et, quoi qu'il en soit, cela* ***requiert*** *un examen* ***sectorielles*** *préalable.*

Paraphrase 5.9 – *Problème d'accord en genre et en nombre :*
O : À Vienne, la stratégie européenne pour l'emploi a reçu une nouvelle impulsion ***ambitieuse****.*

avez construit ||| avons bâti ||| 0.00816328
avez construit ||| avons construit ||| 0.0085034234694
avez construit ||| avons construite ||| 0.0204082

TABLEAU 5.2: Quelques entrées incorrectes de la table de paraphrases résultant du choix de l'anglais comme langue pivot.

*P : * À Vienne, la stratégie européenne pour l'emploi a reçu une nouvelle impulsion* ***ambitieux****.*

Une autre limite des modèles de langue *n*-grammes est visible dans la paraphrase 5.10. Dans cet exemple, une erreur d'accord est produite en introduisant le pronom « elle » plutôt qu'un « il » comme l'impose le début de la phrase – avec « Il est » – et la fin de la phrase – avec « de lui ». Cette dépendance syntaxique ne peut être vue qu'en regardant dans un voisinage de 7 mots autour du pronom. Un modèle de langue *n*-grammes qui observe uniquement le voisinage proche ne sera jamais capable de corriger une telle erreur qui nécessiterait d'avoir une approche globale de la phrase.

Paraphrase 5.10 – *Problème de dépendance à longue distance :*
O : Il est tombé gravement malade et n'a personne pour prendre soin de lui.
*P : * Il est tombé gravement malade,* ***et elle*** *n'a personne pour prendre soin de lui.*

La table de paraphrases et plus particulièrement les erreurs d'alignements sont aussi responsables de certaines erreurs qui ne relèvent d'aucune classification linguistique. Par exemple, la paraphrase 5.11 introduit un « bien » à cause de la règle « dans son ||| bien dans son ||| 0.261438 ». La présence d'un mot rare ou hors vocabulaire – un nom propre ici – fait que le modèle de langue est incapable de rattraper la faute.

Paraphrase 5.11 – *Problème de table de traduction :*
O : Toutefois, nous devons nous préoccuper de la survie de l'État de droit, des libertés civiles et des droits de l'homme, comme l'a souligné M. Watson dans son ***rectificatif*** *très appréciable.*
*P : * Toutefois, nous devons nous préoccuper de la survie de l'État de droit, des libertés civiles et des droits de l'homme, comme l'a souligné M. Watson* ***bien*** *dans son* ***corrigendum*** *très appréciable.*

Lorsque les paraphrases sont jugées syntaxiquement correctes mais sémantiquement incorrectes, c'est souvent un problème de contexte. Par exemple, la paraphrase 5.12 est valide si les phrases autour définissent le contexte de l'immunité. En l'absence de contexte, les deux juges estiment que les deux phrases n'ont pas le même sens. La paraphrase 5.13 introduit une modification valide au regard du contexte général du corpus d'apprentissage EUROPARL. En effet, « M. Fayot » est effectivement président d'une commission ce qui fait que les deux phrases doivent être jugées valides dans le contexte du parlement européen. C'est pour cela que nous retrouvons, dans la table de paraphrases, l'entrée suivante : « du président

Fayot ||| de M. Fayot ||| 0.0833335 ». Nous voyons que la relation de proximité sémantique d'un système par apprentissage est dépendante du contexte. Elle s'inscrit dans le temps et l'espace du corpus d'apprentissage.

Paraphrase 5.12 – *Problème de conservation du sens :*
O : Notre recommandation préconise de ne pas lever ***son*** *immunité.*
*P : * Notre recommandation préconise de ne pas lever* ***l'****immunité.*

Paraphrase 5.13 – *Problème de contexte inconnu :*
O : Pour des raisons pratiques, je considère donc que la proposition ***de M.*** *Fayot est bonne.*
*P : * Pour des raisons pratiques, je considère donc que la proposition* ***du président*** *Fayot est bonne.*

À la suite de Eco [2007], nous considérons que l'équivalence sémantique stricte n'est pas possible. C'est bien une proximité sémantique que les juges doivent évaluer. La frontière entre ce qui « dit la même chose » et ce qui « ne le dit pas » est laissée à leur appréciation. Ainsi, la paraphrase 5.14 est jugée incorrecte car ils jugent qu'une « contribution » apporte plus qu'une « intervention ». *A posteriori*, les évaluateurs estiment qu'ils ont jugé sévèrement cette paraphrase. En revanche, ils ont jugé correcte la paraphrase 5.15. Or, on peut estimer à bon droit que la paraphrase ne dit pas qu'il n'y a pas plus que 130 espèces de requin. Nous voyons bien que l'évaluation des paraphrases reste un problème difficile, même pour les humains.

Paraphrase 5.14 – *Jugement sévère :*
O : ***Merci beaucoup pour*** *votre* ***contribution*** *ici ce soir.*
*P : * **Je vous remercie de** *votre* ***intervention*** *ici ce soir.*

Paraphrase 5.15 – *Un jugement clément :*
O : Il est insensé de continuer à agir comme on le fait aujourd'hui, alors que ***pratiquement*** *50%* ***des*** *130 espèces de requin sont* ***aujourd'hui menacées****.*
P : Il est insensé de continuer à agir comme on le fait aujourd'hui, alors que ***près de*** *50%* ***de*** *130 espèces de requin sont* ***à présent mises en péril****.*

5.1.2 *Stabilité des résultats*

Nous nous intéressons maintenant à la stabilité des évaluations.

Afin d'estimer la dépendance des résultats envers le corpus de test, nous avons refait l'expérience décrite en début de section mais sur le jeu TEST 2. Les juges sont les mêmes et plus de trois mois se sont écoulés entre les deux évaluations. Les résultats sont présentés dans le tableau 5.3.

Nous constatons une amélioration significative des performances mais avec un taux d'accords entre les juges moins important. Un corpus de 100 phrases extrait aléatoirement ne semble pas suffisamment important pour avoir une estimation précise des performances d'un système. En revanche, ces corpus permettent tout de même de comparer différents systèmes entre eux.

Nous avons aussi demandé à un de nos juges de refaire l'évaluation sur le jeu TEST 1 deux mois après la première évaluation. En considérant ces deux évaluations comme deux juges différents, les résultats sont présentés dans le tableau 5.4.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	29	7	30	9	40	11
oui	12	52	12	48	12	**37**
Kappa	**0,57** (p-valeur $< 10^{-3}$) **Accord modéré**					

TABLEAU 5.3: Évaluation du générateur de paraphrases statistique de référence sur le jeu TEST 2.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	54	3	51	7	72	4
oui	4	39	12	30	4	**20**
Kappa	**0,73** (p-valeur $< 10^{-3}$) **Accord substantiel**					

TABLEAU 5.4: Stabilité de l'évaluation dans le temps : les deux juges sont la même personne à deux mois d'intervalle.

Cette expérience n'étant réalisée que pour un seul évaluateur, il n'est pas possible de généraliser les résultats, mais il donne tout de même des tendances. Nous constatons un taux d'accords important – 0,73 – ce qui montre que le juge est resté relativement stable dans ses évaluations. Le juge a apprécié les systèmes de la même façon puisque les performances restent similaires : le décompte de bonne syntaxe passe de 40 à 39, de bonne sémantique de 32 à 30 et de bonnes paraphrases de 22 à 20.

5.1.3 *Comparaison avec d'autres travaux*

Afin de savoir comment le système du chapitre 4 se situe par rapport à l'état de l'art, nous le comparons ici avec ceux d'autres travaux de la littérature. Comme précisé dans la section 2.3.3, il n'existe pas de consensus quant à la méthode d'évaluation des paraphrases. La comparaison est donc difficile à mettre en œuvre. De plus, comme le montre l'expérience de la section précédente, les résultats semblent particulièrement sensibles au corpus de test utilisé.

Dans les travaux de Max [2008], le corpus de test est constitué de 82 phrases. La tâche consiste à produire des paraphrases en modifiant pour chaque phrase, et en une fois, un unique segment sélectionné manuellement. Les paraphrases sont évaluées par deux juges selon trois critères : la grammaticalité, la conservation du sens et l'intérêt pour l'aide à l'écriture. Elles sont notées sur une échelle allant de 1 à 5 – 1 pour mauvais jusqu'à 5 pour parfait. Un système a pour score la moyenne

arithmétique des notes des paraphrases qu'il sélectionne. Le système jugé le meilleur a un score de grammaticalité de $4{,}68 \pm 0{,}59$. Son score en conservation du sens est de $4{,}09 \pm 0{,}7$. Ce système utilise une table de paraphrases par langue pivot – l'anglais – pour produire des paraphrases en français. Le modèle de score utilise un modèle de langue 5-grammes, une distance sur les lemmes des mots, un modèle de préservation des dépendances lexicales et les scores de la table de paraphrases. Une évaluation sur cinq échelons semble difficile pour les évaluateurs au regard des écarts-types. Ce type dévaluation ne permet aucunement de comparer les résultats aux nôtres. En effet, dans notre modèle, une paraphrase notée 4 ou moins serait probablement évaluée comme fausse.

Dans le système de Bannard et Callison-Burch [2005] l'évaluation ne se porte que sur les segments d'une table de paraphrases construite par langue pivot, l'allemand, pour produire des paraphrases en anglais. 46 segments sont utilisés pour l'évaluation. Pour chaque segment, entre 2 et 10 phrases sont retenues afin de former un corpus de test de 283 phrases. Pour cette expérience, la place des modifications est donc fixe. L'évaluation s'effectue selon des critères très similaires à ceux décrits dans le chapitre 3 : deux juges, une évaluation syntaxique et une évaluation sémantique en « oui ; non ». Pour être valide, une paraphrase ne doit jamais être jugée « non ». Le taux d'accords mesuré entre les juges est de 0,6 ce qui est comparable avec nos résultats. Les performances du système varient entre 55,3 – en utilisant uniquement le score de la table de paraphrases et un modèle de langue – à 61,9 – en utilisant plusieurs langues pivots et en ajoutant un module de contrôle du sens. Ces résultats semblent montrer qu'il est plus simple de produire une paraphrase correcte en ne modifiant qu'une petite partie restreinte de la phrase d'origine plutôt qu'en s'autorisant à modifier plusieurs morceaux de phrases.

Dans Quirk et coll. [2004] deux systèmes sont comparés. Dans ces deux systèmes la place des modifications est libre. Un système est l'un des premiers générateurs de paraphrases par corpus d'apprentissage, proposé par Barzilay et Lee [2003]. Comme le décrit la section 2.3.1, cette approche repose sur un algorithme d'alignement de séquences. Le second système, proposé par Quirk et coll. [2004], est un décodeur statistique fondé sur le même principe que celui présenté dans la section 4.4. Il est demandé à deux juges de déterminer directement si les phrases produites sont des paraphrases acceptables des phrases d'origine. Notons que dans ce protocole, les juges connaissent la phrase d'origine. Après une première évaluation du corpus, les paraphrases pour lesquelles les juges sont en désaccord sont présentées à nouveau afin qu'ils puissent modifier leur évaluation, si nécessaire. Cette deuxième étape a pour effet d'augmenter fortement l'accord entre les juges : le pourcentage d'accord passe de 84,0% à 96,9% – le coefficient kappa n'est pas précisé.

Le système de Barzilay et Lee [2003] produit 78,0% de paraphrases jugées correctes avec cette évaluation. Le corpus d'évaluation est constitué de 59 phrases provenant d'articles courts de journaux. Le système de Quirk et coll. [2004] atteint lui 89,5%. Le corpus d'évaluation est constitué des 59 phrases précédentes auxquelles 141 phrases sont ajoutées. Cette évaluation utilise une définition beaucoup plus souple de la relation de paraphrase que celle utilisée dans nos travaux. En effet, pour le système de Barzilay et Lee [2003], les juges estiment que :

- 73% des phrases générées perdent des informations ;
- 32% des phrases générées ajoutent des informations.

Pour le système de Quirk et coll. [2004], au moins :

– 31% des phrases générées perdent des informations ;
– 6% des phrases générées ajoutent des informations.

Ces différences fondamentales dans la définition et dans l'approche de l'évaluation rendent les résultats difficilement comparables.

La principale différence entre ces deux travaux et notre approche est le corpus d'apprentissage. Les systèmes de Barzilay et Lee [2003] et de Quirk et coll. [2004] utilisent des corpus de paraphrases extraites d'articles de journaux. Barzilay et Lee [2003] utilisent un corpus de 9Mo d'articles de journaux sur des attentats pour en extraire 6 534 schémas de transformations. Quirk et coll. [2004] utilisent un corpus de 138 000 paraphrases très proches des phrases d'origine – distance d'édition moyenne de 5,17 pour une longueur moyenne de 18,6 mots. Compte tenu des différences d'échelle, la table de paraphrases produite doit être beaucoup plus petite que notre approche utilisant EUROPARL. Dans ces deux travaux, le nombre de phrases pour lequel une paraphrase peut être produite est vraisemblablement beaucoup plus faible que pour une approche par langue pivot – même si notre table contient probablement plus de transformations incorrectes.

Zhao et coll. [2009] proposent une méthode de production de paraphrases qui s'appuie sur un décodeur statistique mais qui mélange des tables de paraphrases provenant de différentes sources. Ces tables peuvent être produites grâce à une langue pivot, des corpus de paraphrases ou des collocations extraites de corpus comparables. Un corpus de test de 500 phrases est évalué par deux juges, sur une échelle de 1 à 3, sur deux critères : la conservation du sens et la naturalité. Pour la conservation du sens, l'échelle est la suivante [1] :

3 – le sens est complètement préservé ;

2 – le sens est grossièrement préservé ;

1 – le sens est clairement changé.

Pour la syntaxe, l'échelle est la suivante [2] :

3 – **t** est une phrase parfaite ;

2 – **t** est compréhensible ;

1 – la paraphrase **t** est incompréhensible.

Les résultats du système sont présentés dans le tableau 5.5. Au travers de ces résultats, on constate qu'une grande partie des évaluations est regroupée dans le jugement intermédiaire.

Les taux d'accords entre juges sont comparables avec ceux des expériences de la section 5.1. De même, en considérant que dans notre protocole d'évaluation, une évaluation « Oui » correspond à une note « 3 » dans l'évaluation de Zhao et coll. [2009], les performances sont du même ordre.

5.1.4 *Discussions*

Nous avons montré les points forts et les faiblesses du générateur de paraphrases décrit dans le chapitre 4. Celui-ci est capable de produire des paraphrases variées

1. [*3 – The meaning is completely preserved; 2 – The meaning is generally preserved; 1 – The meaning is evidently changed.*]
2. [*3 – **t** is a flawless sentence; 2 – **t** is comprehensible; 1 – the paraphrase **t** is incomprehensible.*]

	sémantique			syntaxe		
	1	2	3	1	2	3
juge 1	29,45	52,76	17,79	25,15	52,76	22,09
juge 2	33,95	46,01	20,04	27,61	48,06	24,34
Kappa	0,66 Accord substantiel			0,65 Accord substantiel		

TABLEAU 5.5: Performances du système de référence de Zhao et coll. [2009]. Les notes vont de 1 à 3 – pour incorrecte à parfait. Les résultats sont du même ordre que pour notre générateur statistique.

comprenant un certain nombre de transformations non triviales. Nous avons mis en évidence l'influence de la méthode de construction de la table de paraphrases sur les transformations opérées mais aussi sur les erreurs introduites. Nous avons surtout montré les limites de modèle de langue n-grammes pour corriger la syntaxe des paraphrases. Compte tenu des résultats, nous pensons que c'est ce composant qu'il faut améliorer pour améliorer la production de paraphrases. Malheureusement, une éventuelle modification du modèle de syntaxe reste contrainte par les limites du décodeur, comme nous le verrons dans la section 5.3.

Pour ce qui est de l'évaluation, nous avons montré que le jugement des évaluateurs est parfois inattendu et que la conservation du sens un concept ambigu. D'un autre côté, les forts taux d'accords entre les juges sur les différentes expériences ainsi que les résultats sur la stabilité valident notre plateforme d'évaluation.

Enfin, nous avons montré qu'il est extrêmement difficile de comparer les résultats de plusieurs travaux. Ceux-ci utilisent des protocoles d'évaluation très différents, sur des jeux de test tout aussi différents. Malgré cela, le système décrit dans le chapitre 4 que nous avons construit « semble » avoir un niveau de performance du même ordre que ceux de l'état de l'art. Ce système nous servira donc de référence pour le reste de ce document.

5.2 LIMITES LORS DE L'INTÉGRATION À UN SYSTÈME DE SYNTHÈSE VOCALE

Après avoir réalisé une évaluation intrinsèque du générateur de référence, nous nous tournons vers notre problématique pour la production de paraphrases. Le but premier de nos travaux est d'étudier l'apport des paraphrases sur un système de synthèse vocale.

Cette section décrit une première tentative d'association des deux outils. Le système transmet la phrase d'origine à un module de production statistique de paraphrases. Celui-ci utilise une table de paraphrases et un modèle de langue afin de produire n paraphrases. Ces paraphrases sont transmises à un second module de sélection qui choisit la meilleure paraphrase en fonction de la voix utilisée par la synthèse vocale. La meilleure paraphrase est transmise à un module de synthèse vocale qui produit le signal acoustique. L'architecture globale du système est donnée dans la figure 5.1.

Le générateur de paraphrases est constitué du système présenté dans le chapitre 4. Le système de synthèse vocale utilisé est BARATINOO, développé par Orange

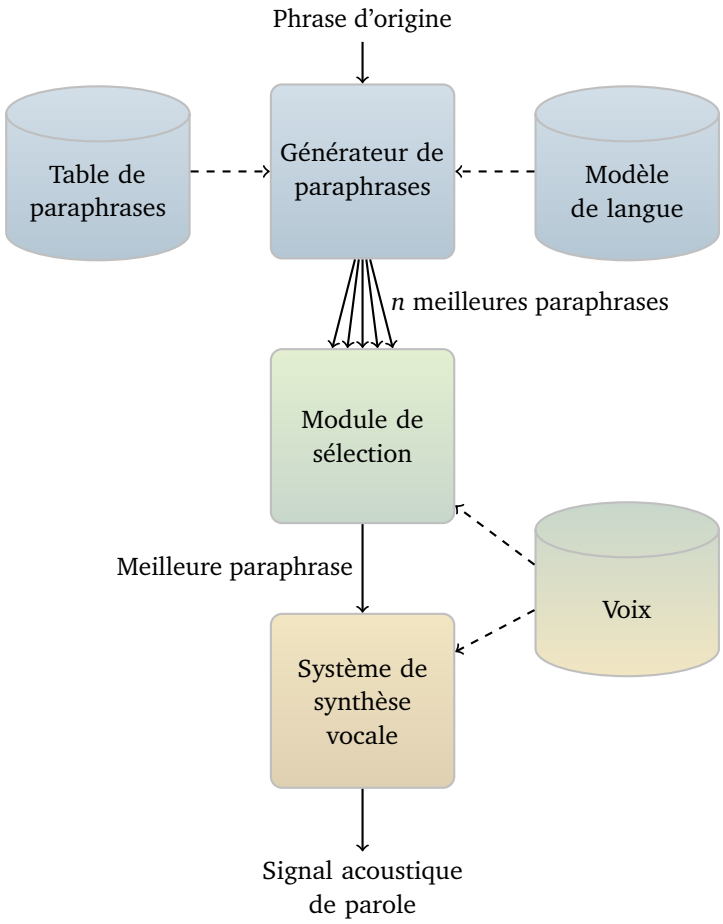


FIGURE 5.1: Architecture d'un système de synthèse vocale enrichi d'un générateur de paraphrases.

Labs que nous avons présenté au chapitre 1. Dans un premier temps, le module de sélection est conçu pour être le plus simple possible. Il utilise uniquement la fonction de coût du module de synthèse vocale pour évaluer les paraphrases et conserver la meilleure. Nous avons montré dans [Boidin et coll., 2009] que le classement des paraphrases par des humains est bien corrélé avec l'ordre défini par cette fonction. En pratique, le module de sélection soumet chaque paraphrase à la synthèse vocale qui lui retourne le « coût de synthèse ». La sélection consiste alors à retenir la paraphrase qui a le coût le plus faible.

D'après le modèle des paraphrases du chapitre 4, plus le nombre n de paraphrases transmises est faible et plus les dernières de la liste ont des chances d'avoir un score élevé d'après le modèle de la paraphrase – et donc d'être des paraphrases correctes. D'un autre côté, plus n est faible et moins les paraphrases sont différentes de la phrase d'origine. Le module de sélection observera alors moins de variabilité dans les scores de synthèse vocale. Les chances d'améliorer la synthèse vocale seront donc

plus faibles. La valeur de n permet de régler le compromis entre amélioration de la synthèse et correction linguistique des paraphrases.

5.2.1 *Paramètres de l'expérience*

Pour cette expérience, le nombre n de paraphrases en sortie du générateur est fixé empiriquement à dix. Le module de sélection et le module de synthèse utilisent une voix féminine nommée *Julie*, composée d'approximativement cinq heures de parole enregistrée.

Le système est testé sur l'ensemble du jeu TEST 1. Les sorties du système sont évaluées selon trois critères. Deux des critères sont ceux définis pour l'évaluation des paraphrases : la conservation du sens et la naturalité. Ils sont évalués grâce à la plateforme décrite dans la section 3.2.

Le troisième critère est la mesure de la qualité acoustique des paraphrases synthétisées. Les systèmes de synthèse de la parole sont traditionnellement évalués à l'aide d'un test d'écoute. Cette évaluation est réalisée à l'aide d'un score d'opinion moyenne (SMO) [UIT, 1996]. Pour notre expérience, sept évaluateurs francophones ont noté la qualité acoustique générale des phrases sur une échelle allant de 1 à 5 – 1 pour mauvaise et 5 pour excellente. Le SMO d'un système est la moyenne arithmétique des notes données pour chaque phrase qu'il a produite.

Il semble difficile d'évaluer la qualité acoustique de phrases qui sont syntaxiquement incorrectes ou qui n'ont pas de sens. C'est pourquoi, seules les paraphrases évaluées comme correctes sur les deux critères linguistiques sont présentées lors de l'évaluation acoustique.

5.2.2 *Résultats*

Les résultats de l'évaluation linguistique des paraphrases sont donnés dans le tableau 5.6.

De façon étonnante, les performances du générateur seul – données dans le tableau 5.1 – sont inférieures à celles en sortie du module de sélection : alors que seulement 22 paraphrases étaient jugées correctes, après l'utilisation du module de sélection 30 paraphrases sont jugées correctes. Le score de la synthèse vocale est lié au texte présent dans le corpus de synthèse. Le module de sélection a peut-être permis de corriger certaines erreurs en choisissant une phrase plus adaptée au corpus de parole. Toutefois, il faut noter que les deux évaluations n'ont pas été réalisées en même temps. Cela peut avoir introduit un biais expliquant la majorité des différences.

Par rapport aux phrases d'origine du jeu TEST 1, les paraphrases en sortie du module de sélection ont un coût acoustique – donné par le synthétiseur vocal – inférieur. En considérant les 100 phrases, ce qui inclut les paraphrases incorrectes, les coûts diminuent de 21±29 % en moyenne. En considérant seulement les 30 paraphrases évaluées correctes, le coût est amélioré de 15±20 % en moyenne.

L'évaluation humaine de l'acoustique donne les résultats présentés à la figure 5.2. Bien que le système incluant un générateur de paraphrases soit meilleur que le système sans générateur de 2,9 % – soit 0,1 point – la différence n'est pas significative compte tenu des intervalles de confiance à 95 %.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	60	1	39	4	62	1
oui	3	36	16	41	7	**30**
kappa	**0,76** (p-value $< 10^{-3}$) **Accord substantiel**					

TABLEAU 5.6: Évaluation des performances linguistiques du système de synthèse vocale couplé avec un générateur de paraphrases.

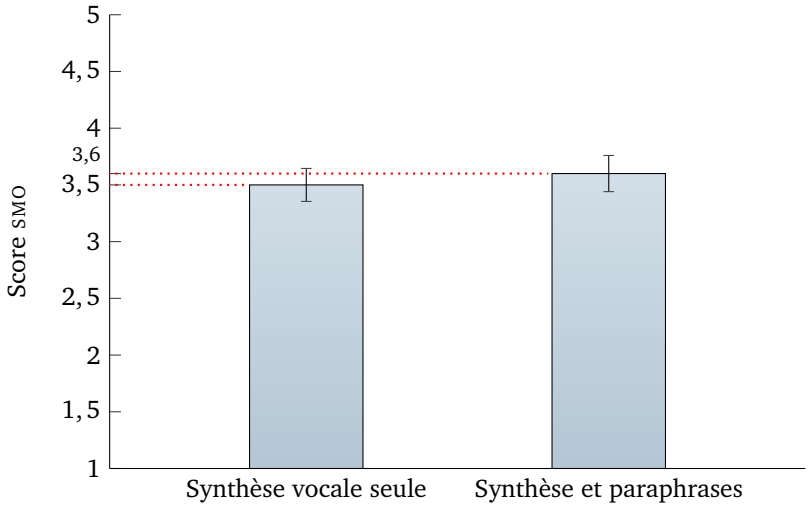


FIGURE 5.2: Résultats du test SMO et intervalles de confiance à 95% : le score d'un système est la moyenne des évaluations acoustiques des 30 paraphrases jugées correctes par 7 juges. Les deux systèmes ne peuvent être distingués.

5.2.3 *Discussions*

Parmi les paraphrases incorrectes, quelques unes ne sont que « légèrement » fausses syntaxiquement et ces erreurs ne seraient plus perçues une fois la paraphrase synthétisée – comme pour la paraphrase 5.16. Ces paraphrases pourraient être considérées « auditivement » valides. Il pourrait être pertinent d'adapter l'évaluation linguistique à la modalité vocale. Plus encore, on peut se demander s'il est judicieux de faire travailler le générateur de paraphrases sur des données textuelles plutôt que sur une représentation acoustique des phrases, comme leur phonétisation par exemple.

Paraphrase 5.16 – *Une faute à l'écrit n'entraîne pas toujours une erreur à l'oral :*
*O : Et, quoi qu'il en soit, cela **exige** un examen **sectoriel** préalable.*
*P : * Et, quoi qu'il en soit, cela **demande** un examen **sectorielles** préalable.*

L'amélioration des coûts de synthèse laissait espérer une amélioration significative de la qualité acoustique perçue lors de l'évaluation humaine. Malheureusement, il n'est pas possible de distinguer le système de synthèse vocale associé au générateur de paraphrases du système seul. Au moins trois points peuvent expliquer ce résultat.

Premièrement, il semblerait que dans cette expérience, le corpus de phrases soit plus adapté au système de synthèse qu'escompté : la couverture des unités acoustiques est suffisamment satisfaisante pour les besoins des phrases d'origine. Une expérience ultérieure devrait être conduite avec une voix plus limitée ou des phrases d'autres domaines.

Deuxièmement, nous constatons que le module de sélection n'a pas réduit les performances linguistiques des paraphrases. Ceci montre que le générateur de paraphrases n'introduit en fait que peu de variabilité. Le taux d'erreurs en caractères (TEC) moyen entre sortie du système et phrase d'origine est de 0,18±0,13. Ceci peut être interprété de la façon suivante : approximativement 18% des caractères des phrases d'origines sont modifiés. Ainsi, la meilleure paraphrase reste probablement trop « proche » de la phrase d'origine pour que cela ait un impact sur la qualité acoustique perçue.

Troisièmement, les modules de sélection et le générateur de paraphrases sont complètement décorrélés : l'endroit où les disfluences acoustiques éventuelles apparaissent dans la phrase d'origine n'a pas de raison d'être choisi par le générateur pour introduire une modification. Pour contourner cette difficulté, il faudrait considérer une optimisation conjointe des modèles linguistiques et acoustiques.

5.3 LIMITES DU DÉCODEUR

Beaucoup de travaux se concentrent sur la table de paraphrases et l'acquisition de ressources nécessaires à sa construction [Sekine, 2005; Fujita et Inui, 2005; Dolan et Brockett, 2005; Barzilay et McKeown, 2001]. Un second élément potentiellement problématique, qui attire moins l'attention, est le générateur lui-même. La plupart des systèmes de production utilisent un outil standard de décodage développé pour la traduction statistique – comme présenté dans la section 4.4. Ces outils n'ont pas été conçus pour le problème de la paraphrase et ne prennent pas en compte ses spécificités. De plus, comme nous l'avons expliqué dans la section 4.4, à cause de

la contrainte d'incrémentalité et de la limitation de l'historique des fonctions de score, ces décodeurs retournent non seulement une paraphrase sous-optimale mais en plus, le score qui lui est associé est aussi sous-optimal.

Cette section s'intéresse aux limites du décodeur statistique fondé sur l'algorithme de Viterbi. La section 5.3.1 décrit un algorithme pour calculer le découpage optimal et le score véritable des paraphrases *a posteriori*. La section 5.3.2 présente une expérience afin d'évaluer l'impact de cette approximation des scores par les décodeurs statistiques. Enfin, la section 5.3.3 analyse les limites rencontrées.

5.3.1 *Score véritable des paraphrases et découpage optimal*

Comme nous l'avons expliqué dans la section 4.4, à cause de la complexité du problème de décodage, les outils utilisent des algorithmes sous-optimaux – comme la recherche par faisceau – pour rechercher la meilleure paraphrase c^*. Le modèle de la production statistique de paraphrases découpe la phrase d'origine s en segments. Le score d'une paraphrase potentielle c sachant un découpage I donné peut s'écrire avec le formalisme de la section 4.1 :

$$Z_s^I(c) = P(c) \prod_{i \in I} P(s_i^I | t_i^I, B_{L_s \to L_s})$$

À cause de l'approximation de la recherche par faisceau, les solutions retournées par le décodeur sont des estimations sur l'ensemble des paraphrases possibles mais aussi sur l'ensemble des découpages. En fait, pour une paraphrase c donnée, seuls quelques scores $Z_s^I(c)$ sont considérés par le décodeur.

Nous définissons le score de découpage optimal d'une paraphrase dans la définition 5.1

Définition 5.1 *Le score de découpage optimal d'une paraphrase potentielle est le maximum des scores sur tout découpage :*

$$Z_s^+(c) = \max_I Z_s^I(c)$$

Il n'y a pas de raison que les scores retournés par le décodeur soient les scores issus du découpage optimal. Ceci pose problème lorsque le score des paraphrases produites est important. Par exemple, le système de production intégré à un système de synthèse vocale peut avoir besoin de produire une liste ordonnée de solutions. Compte tenu des approximations réalisées, nous nous demandons si les scores estimés et les ordres associés calculés par le décodeur sont proches des scores et ordres du découpage optimal.

De plus, une approximation forte est faite dans le modèle présenté dans la section 4.1 et dans l'équation 4.5 : la somme sur tous les découpages est approximée par le découpage maximum. Nous définissons le score véritable d'une paraphrase dans la définition 5.2.

Définition 5.2 *Le score véritable d'une paraphrase potentielle est la somme des scores sur tout découpage :*

$$Z_s^*(c) = \sum_I Z_s^I(c)$$

Notons que le score véritable reste incrémental. Nous le montrons en détaillant une itération du décodeur de façon similaire à ce qui a été fait dans la section 4.4.

Nous supposons que le programme a déjà calculé $n-1$ itérations c'est-à-dire qu'il a réalisé toutes les « traductions » possibles jusqu'au $n-1^{\text{e}}$ mot. Le programme dispose de toutes les « traductions » possibles des préfixes de longueur supérieur à $n-1-k$, où k est le maximum entre l'ordre du modèle de langue et la longueur du plus long support parmi les options de traductions. Le décodeur dispose du score véritable de chacune de ces « traductions ».

Pour calculer l'itération n, le programme va calculer l'ensemble des options de « traduction » qui couvre le n^{e} mot de la phrase d'origine. Le programme multiplie donc le coût de chaque option avec le score véritable de chaque « traduction » de support compatible. Il reste à fusionner les « traductions » qui conduisent à la même forme de surface et faire la somme de leur score.

Même si le score est toujours incrémental, il n'est plus possible de limiter la taille de l'historique nécessaire au calcul de l'itération suivante. En effet, il faut conserver en mémoire une somme par forme de surface produite. La complexité de l'algorithme de Viterbi redevient donc exponentielle selon la longueur de la phrase.

Calculer le score véritable ou le score du découpage optimal d'une paraphrase produite est en fait une tâche algorithmiquement plus simple que celle consistant à produire la meilleure paraphrase : une fois que la phrase « cible » est définie, l'ensemble des découpages possibles est en effet calculable. Ceci est encore plus simple en l'absence de modèle d'ordonnancement. L'algorithme 1 présente une solution « naïve » de calcul *a posteriori* du score véritable d'une paraphrase. Ce calcul est fondé sur un parcours en profondeur de l'ensemble des entrées de la table de paraphrases.

5.3.2 *Score des décodeurs statistiques*

Nous venons de montrer comment calculer simplement le score véritable pour le modèle de la production statistique ou le score de découpage optimal d'une paraphrase donnée. Nous nous intéressons maintenant à la pertinence des scores délivrés par le décodeur de référence de la section 4.4. En particulier, nous souhaitons comparer la stabilité de l'ordre « véritable » d'une liste de paraphrases suite aux heuristiques introduites par un décodeur statistique.

Pour chaque phrase du jeu TEST 1, le décodeur MOSES essaie de produire une séquence constituée des 100 meilleures paraphrases distinctes. Notons que pour ces expériences, la paraphrase identité n'est pas supprimée de la sortie de MOSES. Les scores véritables et les scores de découpages optimaux sont ensuite calculés grâce à l'algorithme 1. Les paraphrases sont réordonnées en conséquence.

L'ordre original et les ordres après réordonnancement peuvent être comparés grâce au coefficient de corrélation des rangs de Kendall (τ_A) [Kendall, 1938]. Ce coefficient mesure la conservation de l'ordre relatif de tout couple de paraphrases. Sa formule est la suivante :

$$\tau_A = \frac{n_p - n_i}{\frac{1}{2}n(n-1)} \tag{5.1}$$

Algorithme 1 : calcul *a posteriori* du score véritable et du score de découpage optimal d'une paraphrase

Entrées :

s la phrase d'origine ;

c la paraphrase à évaluer ;

Ω la table de paraphrases composée d'entrées R de la forme $s_R|||c_R|||P(s_R|c_R, B_{L_s \to L_s})$.

Sorties :

$Z_s^*(c)$ le score véritable de la paraphrase c ;

$Z_s^+(c)$ le score de découpage optimal de la paraphrase c.

Fonction principale :

$Z_s^*(t) \leftarrow 0$;

$Z_s^+(t) \leftarrow -\infty$;

$D \leftarrow$ découpe(s, c) ;

pour tout $d \in D$ faire

$Z_s^*(c) \leftarrow Z_s^*(c) + \prod_{R \in d} P(s_R|c_R, B_{L_s \to L_s})$;

$Z_s^+(t) \leftarrow \max\left(Z_s^+(c), \prod_{R \in d} P(s_R|c_R, B_{L_s \to L_s})\right)$;

$Z_s^*(c) \leftarrow Z_s^*(c) \times P(c)$;

$Z_s^+(c) \leftarrow Z_s^+(c) \times P(c)$;

retourner $Z_s^*(c)$ et $Z_s^+(c)$.

Avec découpe : $(s, c) \to I$ une fonction telle que :

$I \leftarrow \{\emptyset\}$;

pour tout $s_{prefixe}$ tel que $s = s_{prefixe}.s_{suffixe}$ faire

pour tout $R \in \Omega$ tel que $s_R = s_{prefixe}$ et $c = c_R.c_{suffixe}$ faire

$I \leftarrow I \cup (\{R\} \otimes$ découpe$(s_{suffixe}, c_{suffixe}))$;

retourner I.

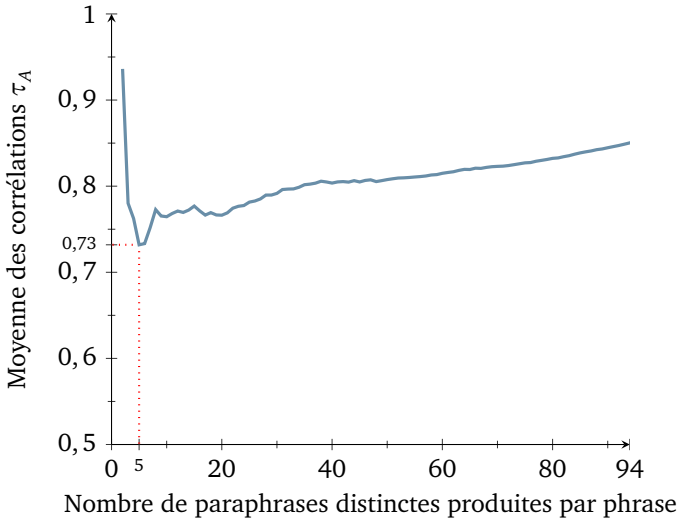


FIGURE 5.3: Évolution du τ_A moyen en fonction du nombre de paraphrases distinctes produites par phrase. Le minimum est de 0,73 pour 5 paraphrases. L'ordre en sortie du générateur est différent mais pas décorrélé de l'ordre du découpage optimal. L'heuristique de la recherche par faisceau a un impact sur les paraphrases produites.

où n_p est le nombre d'ordres préservés, n_d le nombre d'ordres inversés et n le nombre de paraphrases dans la séquence. Le terme de normalisation $\frac{n(n-1)}{2}$ correspond au nombre de couples de paraphrases possibles.

Le coefficient est un score qui peut être interprété comme un coefficient de corrélation entre les deux ordres. Le τ_A est de 1 si les deux ordres sont identiques ; de -1 si les deux ordres sont inversés ; la permutation uniforme d'un ordre entraîne un τ_A de 0 en moyenne.

Afin de comparer des séquences de même longueur, les phrases d'origine pour lesquelles MOSES n'est pas en mesure de produire 100 paraphrases ne sont pas prises en compte. Le corpus de test est ici réduit à 94 phrases.

Pour le score de découpage optimal, l'évolution de la moyenne des τ_A en fonction du nombre de paraphrases distinctes produites par phrase est tracée figure 5.3. Le minimum est atteint pour 5 paraphrases avec une corrélation de 0,73. Ceci montre que les ordres sont clairement différents mais pas décorrélés. Une étude plus fine des résultats montre que parmi les paraphrases produites, 32% ont leur score modifié. L'approximation de recherche par faisceau ignore donc beaucoup de chemins optimaux. De plus, 18% des meilleures paraphrases ne sont plus optimales après le réordonnancement selon le score de découpage optimal. Nous voyons ici que cette heuristique a un impact fort même lorsque l'on ne produit qu'une seule paraphrase.

Le rang moyen des meilleures paraphrases était de 4,4±12,1 avant réordonnancement. Les paraphrases qui ont remplacé d'anciennes meilleures paraphrases provenaient, en moyenne, du rang 21,2±23,5 – avec un minimum de 2 et un maximum de 67. La recherche par faisceau sous-estime beaucoup de paraphrases

qui devraient être les meilleures. La correction de l'heuristique par un réordonnancement *a posteriori* nécessiterait de produire une importante liste de paraphrases.

Enfin, la position moyenne des anciennes meilleures paraphrases est de 2,0±17,7 après réordonnancement – avec un minimum de 1 et un maximum de 40. Si on observe uniquement les meilleures paraphrases qui ne sont plus optimales après le réordonnancement, elles passent du rang 1 au rang 6,8±12,9 en moyenne. L'approximation de la recherche par faisceau favorise certaines paraphrases qui devraient être beaucoup plus bas dans le classement.

Pour le score véritable du modèle de production statistique de paraphrases, l'évolution de la moyenne des τ_A en fonction du nombre de paraphrases distinctes produites par phrase est tracée figure 5.4. Le minimum est atteint pour une production de 7 paraphrases par phrase avec une corrélation de 0,52. Cette fois, les ordres sont beaucoup plus différents. Nous constatons d'ailleurs que 39% des meilleures paraphrases ne sont plus optimales après le réordonnancement dû au score de découpage optimal. L'hypothèse simplificatrice d'approximation de la somme par un maximum modifie donc profondément le modèle et la notion de « meilleure paraphrase ».

Le rang moyen des meilleures paraphrases était de 5,9±13,0 avant réordonnancement. Les paraphrases qui ont remplacé d'anciennes meilleures paraphrases provenaient, en moyenne, du rang 13,5±18,5 – avec un minimum de 2 et un maximum de 67. La paraphrase la meilleurs au sens du modèle originel du canal bruité peut être reléguée très bas dans le classement.

Enfin, la position moyenne des anciennes meilleures paraphrases est de 3,2±9,3 après réordonnancement – avec un minimum de 1 et un maximum de 37. Si on observe uniquement les meilleures paraphrases qui ne sont plus optimales après le réordonnancement, elles passent du rang 1 au rang 6,0±7,6 en moyenne. La meilleure paraphrase au sens du modèle simplifié est une paraphrase jugée beaucoup moins bonne par le modèle complet.

5.3.3 *Discussions*

Les décodeurs statistiques de paraphrases sont généralement les mêmes systèmes que ceux pour la traduction. Compte tenu des faibles performances de la production statistique de paraphrases, il est pertinent de se demander si les paraphrases n'ont pas certaines spécificités qui demandent d'adapter ces outils. Voici plusieurs aspects qui nous semblent justifier cette adaptation des outils.

Premièrement, si le décodage de gauche à droite peut se justifier pour la traduction, il n'est probablement pas nécessaire pour la production de paraphrases. Un générateur de paraphrases a pour entrée une phrase qui peut être proche d'une solution. C'est-à-dire qu'il n'est pas nécessaire de « traduire » toute la phrase.

Deuxièmement, les résultats présentés dans la section 5.1 montrent qu'un des problèmes majeurs du système porte sur la qualité syntaxique des paraphrases. Nous avons montré qu'un modèle de langue n-grammes est trop simple et n'est pas capable de prendre en compte des dépendances syntaxiques au-delà de l'historique qu'il sait traiter. Or, pour envisager des modèles plus complexes, il faut pouvoir évaluer une paraphrase potentielle dans son ensemble.

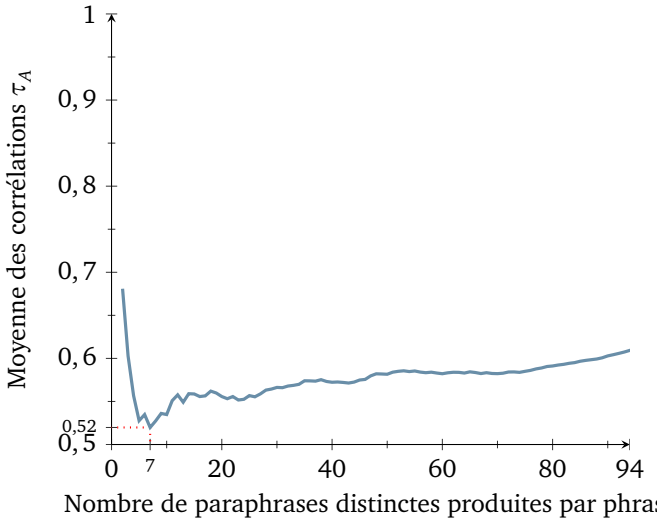


FIGURE 5.4: Évolution du τ_A moyen en fonction du nombre de paraphrases distinctes produites par phrase. Le minimum est de 0,52 atteint pour 7 paraphrases. L'ordre en sortie du générateur est clairement différent de celui donné par le score véritable. L'heuristique simplificatrice du modèle de la production statistique de paraphrases à un impact fort sur les résultats.

Troisièmement, comme nous l'avons expliqué dans la section 4.4, l'algorithme de Viterbi nécessite une fonction d'évaluation incrémentale. Cette contrainte est incompatible avec une évaluation globale des solutions potentielles au fil de la production. Ce problème se retrouve aussi pour la traduction statistique [Langlais et coll., 2008].

Enfin, l'algorithme de Viterbi a besoin de pouvoir travailler avec un historique réduit pour être efficace. Pour résoudre cette contrainte, un certain nombre d'hypothèses sur le modèle de production de paraphrases sont faites, comme nous l'avons montré dans la section 4.1. Nous avons réalisé une estimation de l'impact de ces heuristiques dans la section 5.3.2.

L'expérience sur les scores de découpage optimal démontre que l'heuristique de recherche par faisceau à un impact sur les scores retournés par MOSES. En particulier, l'ordre des n meilleures solutions n'est pas fiable et ne devrait donc pas être pris en compte par un système de réordonnancement – comme nous aurions pu le faire dans la section 5.2.

L'expérience sur les scores véritables démontre que les scores retournés par MOSES ne sont pas cohérents avec le score véritable défini par un modèle général de la paraphrase. L'hypothèse simplificatrice utilisée pour rendre calculable le modèle de la paraphrase par un algorithme de type Viterbi éloigne fortement les résultats de ceux attendus.

La contrainte d'incrémentalité de la fonction de score et de restriction sur l'historique limite les modèles possibles pour la paraphrase. C'est pour nous un des problèmes majeurs empêchant l'amélioration des générateurs de paraphrases. Il

nous semble donc nécessaire de proposer une autre approche et des outils adaptés à la production de paraphrases.

5.4 LIMITES DE L'ÉVALUATION

Comme le montre la section 2.1.1, il existe des définitions variées de la paraphrase dans le domaine de la production automatique de paraphrases. Toutes ces définitions insistent sur une relation de conservation du sens entre la phrase d'origine et la paraphrase produite.

Cette focalisation sur la conservation du sens fait que les systèmes de production et leurs évaluations oublient de prendre en compte la globalité des objectifs de la production automatique de paraphrases : la conservation du sens, la naturalité, mais aussi l'adaptation à la tâche. Par exemple, dans Barzilay et McKeown [2001]; Quirk et coll. [2004], l'évaluation des générateurs porte uniquement sur la conservation du sens. Dans Bannard et Callison-Burch [2005], elle porte aussi sur la naturalité des paraphrases produites, suivant en cela l'évaluation en traduction automatique. Cette évaluation vérifie, entre autres, que les paraphrases sont bien des phrases, c'est-à-dire qu'elles sont syntaxiquement correctes. Plus rarement, les paraphrases sont évaluées uniquement en fonction de l'utilisation qui en sera faite [Cahill et coll., 2009]. Quelques travaux seulement évaluent les paraphrases en fonction des trois critères [Max, 2008; Zhao et coll., 2009].

Dans les sections 5.4.1 et 5.4.2 nous démontrons par l'absurde que l'évaluation d'un système, en ne prenant en compte que la conservation du sens, n'est pas pertinente. Dans la section 5.4.3 nous reprenons des limites de l'évaluation telle qu'elle est généralement posée et montrons que certaines de ces limites sont aussi des limites du modèle de la production de paraphrases.

5.4.1 *Un (trop) bon générateur de paraphrases*

L'objectif de l'expérience suivante est de démontrer qu'une évaluation focalisée sur la conservation du sens n'est pas pertinente. De plus, l'ajout d'un critère d'évaluation lié à la naturalité est certes appréciable mais pas suffisant. Nous proposons de démontrer cela par l'absurde, en comparant deux systèmes uniquement sur des critères de conservation de sens et de naturalité. Nous montrons qu'un système inutile peut, dans une telle évaluation, être jugé meilleur qu'un système au niveau de l'état de l'art.

Le premier système considéré, que nous appelons *Référence* est le générateur de paraphrases statistique décrit dans le chapitre 4.

Le second système est conçu spécialement pour cette expérience. Puisque les générateurs ne seront comparés qu'en fonction du critère de conservation du sens et de la correction syntaxique des paraphrases produites, l'objectif est de concevoir un système prenant le moins de risques possibles. Nous imposons tout de même aux systèmes de proposer des paraphrases différentes de la phrase d'origine. Notre second système *Virgule* est donc capable de supprimer des virgules ou d'en ajouter à la phrase d'origine. Il supprime les virgules de la phrase d'origine s'il y en a. Si la phrase ne comporte pas de virgule, nous utilisons un second sous-système composé lui aussi du décodeur statistique MOSES. La table de paraphrases utilisée est celle du

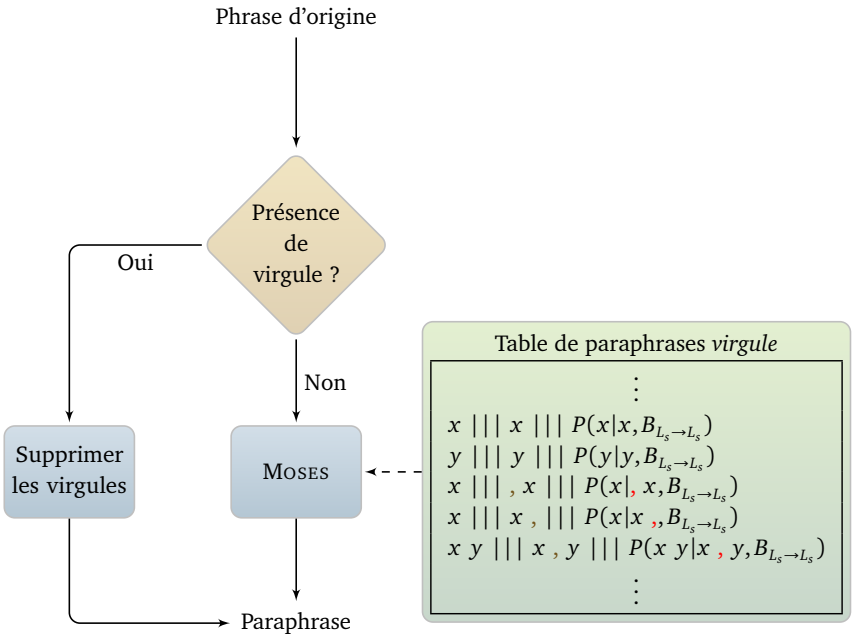


FIGURE 5.5: Principe du générateur de paraphrases *Virgule*. Ce système prend le moins de « risques » possibles en ne manipulant que les virgules pour produire des paraphrases.

système de référence où sont conservées uniquement les entrées qui n'entraînent pas de modification du segment ou qui se contentent d'ajouter une virgule. Le générateur *Virgule* est schématisé à la figure 5.5.

5.4.2 *Résultats*

Les performances détaillées du système *Virgule* sont données dans le tableau 5.7. Contrairement à notre intuition, la présence ou l'absence de virgule semble jouer principalement sur l'aspect syntaxique. Les juges estiment fréquemment que les sens de la phrase d'origine et de sa paraphrase sont les mêmes.

Par exemple, la suppression d'une emphase ne semble pas gêner les évaluateurs. Ainsi, la paraphrase 5.17 est jugée correcte. En revanche, l'ajout de virgules semble une tâche beaucoup plus ardue. En effet, le système est rarement capable d'ajouter des virgules par paire pour créer des emphases, ce qui conduit souvent à une évaluation syntaxique mauvaise, comme pour la paraphrase 5.18.

Paraphrase 5.17 – *La suppression des virgules d'une emphase est jugée correcte :*

O : Agir de la sorte, c'est à vrai dire fouler aux pieds la démocratie, et il me semble tout de même que des gouvernements démocratiquement élus devraient l'éviter autant que possible.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	26	12	0	2	2	11
oui	6	**56**	20	**76**	6	**55**
Kappa	**0,45** (p-value $< 10^{-3}$) **Accord modéré**					

TABLEAU 5.7: Performance du générateur de paraphrases *Virgule* sur le jeu TEST 1

Système	*Référence*	*Virgule*
Sens préservé	40%	**76%**
Syntaxe correcte	32%	56%
Syntaxe correcte et sens préservé	22 %	**55%**
Kappa	0,63 (p-valeur $< 10^{-3}$) Accord substantiel	

TABLEAU 5.8: Comparaison du système de référence avec le système *Virgule*. Le système *Virgule* est meilleur en terme de conservation du sens et aussi lorsque les deux critères sont combinés.

P : Agir de la sorte c'est à vrai dire fouler aux pieds la démocratie et il me semble tout de même que des gouvernements démocratiquement élus devraient l'éviter autant que possible.

Paraphrase 5.18 – *Il manque une seconde virgule après « conséquent » pour créer une emphase :*

O : Il nous faut par conséquent laisser au vestiaire les hésitations et l'ambiguïté.

*P : * Il nous faut, par conséquent laisser au vestiaire les hésitations et l'ambiguïté.*

Les résultats de la production de paraphrases par les deux systèmes sur le jeu TEST 1 sont présentés dans le tableau 5.8. Le coefficient Kappa d'accord entre les juges est de 0,63 (p-valeur $< 10^{-3}$), ce qui est traditionnellement interprété comme « substantiel ».

D'après ces résultats, le système *Virgule* est significativement meilleur que le système *Référence* en terme de conservation du sens. L'ajout du critère de naturalité réduit l'écart de performance des deux systèmes mais le générateur *Virgule* reste meilleur de 142%. Ces résultats ne reflètent évidemment pas l'utilité de ces deux systèmes. Or, en minimisant les risques pris, le système qui manipule uniquement les virgules arrive à produire de meilleurs résultats alors qu'il est certainement inutile pour la majorité des applications réelles de production de paraphrases.

5.4.3 *Discussions*

L'évaluation des paraphrases est un problème difficile qui est très révélateur des problèmes liés à la production de paraphrases et plus généralement à la notion de

sens elle-même. Comme cela a été détaillé dans la section 5.1, l'accord entre les juges lors des évaluations est comparable avec les résultats du domaine. Il reste cependant relativement faible puisque le Kappa reste souvent autour de 0,6. Compte tenu de la définition relativement floue des paraphrases, il est légitime de se demander si une évaluation à seulement deux juges est suffisante pour capter un avis général et si elle est suffisamment stable. L'évaluation des paraphrases par un grand nombre de juges, sur le principe de Callison-Burch [2009], permettrait peut-être de répondre à ces questions.

Notre expérience démontre qu'une évaluation fondée uniquement sur la conservation du sens n'est pas pertinente, même en ajoutant un critère sur la naturalité. Il faut donc revoir ou compléter le protocole d'évaluation.

5.5 CONCLUSION

Dans ce chapitre, nous avons étudié les résultats d'un générateur de paraphrases et mis en évidence certaines lacunes dans le cadre classique de la production statistique de paraphrases.

Premièrement, avec des performances avoisinant les 24% de paraphrases jugées correctes, notre générateur statistique de paraphrases de référence n'est pas suffisamment efficace pour pouvoir être intégré directement dans un système de dialogue humain-machine. Le système est malgré tout capable de produire des alternatives non triviales pour beaucoup de phrases. En revanche, la syntaxe des paraphrases produites est pour nous le point critique à améliorer pour augmenter significativement les performances d'un tel système.

Malgré les difficultés rencontrées lorsque l'on cherche à comparer les résultats de différents travaux, ce système de production de paraphrases semble atteindre des résultats conformes avec l'état de l'art. En comparaison avec d'autres études sur le sujet, les critères utilisés dans nos travaux pour juger une paraphrase valide sont très stricts. Nous justifions cette exigence par la volonté initiale d'intégrer ce type de système dans des outils entièrement automatiques.

Deuxièmement, nous avons montré qu'une utilisation naïve des paraphrases générées dans un système de synthèse vocale ne permet pas d'améliorer significativement les performances. La prise en compte de la synthèse vocale doit être faite directement pendant la phase de production et non pas après.

Troisièmement, nous avons démontré que les contraintes d'incrémentalité du score et de limite de l'historique obligent à poser des hypothèses supplémentaires sur le modèle d'évaluation. Elles limitent les outils utilisables, en particulier pour améliorer la syntaxe des paraphrases et forcent des approximations importantes par rapport au modèle initial. Pour nous, une limite forte de l'approche statistique de la production de paraphrases réside dans ces contraintes ajoutées par le décodeur. Nous estimons que leur suppression est un premier pas nécessaire pour réfléchir au modèle de la paraphrase.

Enfin, les expériences ont permis de valider en partie le protocole d'évaluation que nous proposons et de cerner ses limites. Nous avons en revanche montré l'insuffisance des évaluations fondées uniquement sur la conservation du sens. Il est donc nécessaire d'envisager une correction du protocole d'évaluation des paraphrases mais aussi une modification du modèle de la paraphrase.

L'ensemble des travaux de ce chapitre vont nous servir de point de départ pour proposer des évolutions pour la production de paraphrases.

Troisième partie

PROPOSER : UN AUTRE MODÈLE EN SUPPRIMANT DES CONTRAINTES

6 UN AUTRE CADRE POUR LA PARAPHRASE

Le chapitre 5 a permis de mettre en évidence plusieurs problèmes dans le cadre de la production statistique de paraphrases. Ce cadre est une transposition directe de celui de la traduction statistique au problème de la paraphrase. Nous pensons que cette transposition est responsable de nombreux problèmes car elle ne tient pas compte des spécificités des paraphrases. En d'autres mots : faire une paraphrase, ce n'est pas simplement traduire sans changer de langue.

L'objectif de ce chapitre est de proposer un cadre adapté à la production de paraphrases. Dans un premier temps, nous proposons de définir les objectifs des paraphrases dans la section 6.1. Nous montrons la nécessité de prendre en compte la tâche pour évaluer les outils de production de paraphrases. Nous proposons donc dans la section 6.2 un critère générique pour pouvoir comparer deux générateurs. Enfin, nous proposons dans la section 6.3 une adaptation du cadre de la production de paraphrases en fonction des points mis en avant dans le chapitre 5.

6.1 LES TROIS OBJECTIFS DES PARAPHRASES

Si la conservation du sens et la naturalité seules ne permettent pas d'évaluer un générateur de paraphrases, comme l'a montré la section 5.4.1, c'est que la production de paraphrases doit avoir d'autres buts.

Nous dégageons les objectifs suivants comme les buts de la production de paraphrases :

- la conservation du sens : c'est le but premier de la paraphrase qui la différencie de la production de texte. La phrase d'origine sert de référence en terme de sens à produire;
- la naturalité : il est nécessaire que la paraphrase soit syntaxiquement correcte, afin qu'elle ait un sens;
- l'adéquation à la tâche : la production automatique de paraphrases n'est pas une activité en soi (contrairement à la traduction par exemple). Elle est toujours associée à une tâche et intégrée dans un processus plus vaste. Il est nécessaire que les paraphrases produites soient adaptées à l'usage qu'il en sera fait.

Nous définissons la production de paraphrases comme un compromis entre ces trois objectifs, le réglage de ce compromis dépendant de la tâche. À ce compromis s'ajoute une séparation en deux classes selon que le lieu des modifications est imposé ou non.

La figure 6.1 illustre cette définition du problème de production de paraphrases. La phrase d'origine se trouve quelque part sur l'axe d'inadaptation à la tâche si celle-ci est syntaxiquement correcte. En effet, c'est la phrase la plus proche sémantiquement d'elle-même. La paraphrase idéale se trouve à l'origine de l'espace délimité par ces trois axes : c'est une phrase qui a exactement le même sens que la phrase d'origine, qui est syntaxiquement correcte et qui est parfaitement adaptée à

la tâche. Celle-ci étant probablement impossible à produire, le problème consiste alors à produire la meilleure paraphrase en fonction d'un compromis entre les trois objectifs.

Ces trois objectifs se retrouvent dans les différentes classes d'applications qui utilisent ou pourraient utiliser un générateur de paraphrases. Nous répartissons ces applications en fonction des traitements réalisés, après production, sur les paraphrases proposées. Pour chaque application, nous détaillons le rôle du générateur de paraphrases par rapport au système global.

La première catégorie correspond aux cas où les paraphrases ne sont plus modifiées par le système après production. La paraphrase est une sortie du système global. Parmi les applications fréquemment citées, on trouve :

- la compression de phrase [Knight et Marcu, 2000; Zhao et coll., 2009] : le but est de produire une phrase plus courte en nombre de caractères que la phrase d'origine. Cette application est une version simplifiée du résumé de texte ;
- la synthèse de la parole [Boidin et coll., 2009; Cahill et coll., 2009] : le but est de modifier la phrase d'origine afin que la paraphrase produite, une fois vocalisée, ait une plus grande pertinence. Les travaux actuels se concentrent sur l'amélioration du rendu acoustique. Puisque le synthétiseur vocal ne modifie pas la phrase, nous classons cette application dans cette catégorie.

La seconde catégorie comprend les systèmes où la paraphrase produite peut être modifiée. Mais, la phrase en sortie du système global doit conserver le sens de la phrase d'entrée du générateur de paraphrases. Pour les applications de ce type, c'est souvent un opérateur humain qui réalise les modifications :

- l'aide à l'écriture [Max, 2008] : le but est de fournir des alternatives à un rédacteur lorsque celui-ci n'est pas satisfait d'une partie de la phrase (répétition, terminologie, ...) ;
- l'aide à la conception de messages dans un système de dialogue [Boidin et coll., 2009] : le but est de proposer des alternatives à un concepteur de système de dialogue humain-machine pour les messages du système. Par exemple, ces alternatives peuvent prendre en compte l'ensemble des messages déjà construits pour conserver une certaine homogénéité.

Dans la troisième catégorie, la paraphrase n'est pas directement reliée à la sortie du système global. Elle est une étape de calcul intermédiaire. En général, la paraphrase sert d'entrée à un système de traitement automatique de la langue :

- un système de question-réponse [Duclaye et coll., 2003] : le but est de pouvoir traiter la variabilité linguistique des questions et des réponses faites au système ;
- un système de recherche d'information [Sekine, 2005] : le but est d'améliorer la couverture des schémas que le système sait reconnaître dans un texte en « normalisant » les phrases d'entrée grâce à la relation de paraphrase ;
- un système de traduction automatique [Callison-Burch et coll., 2006] : le but est de remplacer les mots inconnus ou les séquences difficiles à traduire pour le système en séquence de mots facile à traduire.

Chacune de ces applications cherche à conserver le sens de la phrase d'entrée. Les applications de la première catégorie contraignent fortement la qualité syntaxique des paraphrases à produire puisqu'elles sont présentées directement à un utilisateur final. La contrainte sur la conservation du sens et sur la naturalité des paraphrases

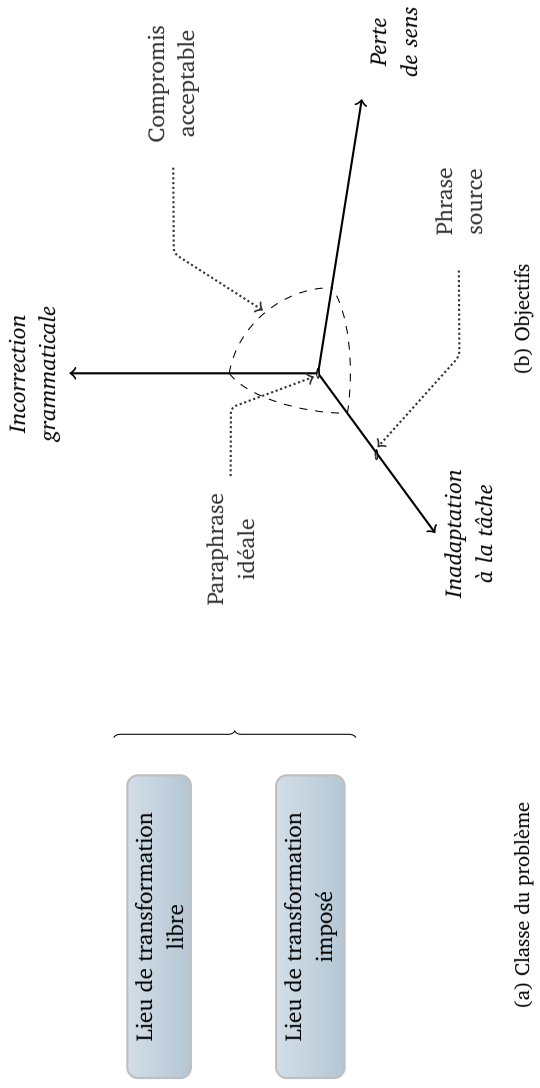


FIGURE 6.1: Modélisation du problème de production de paraphrases

produites est en revanche moins forte pour les applications de la seconde catégorie puisque les paraphrases peuvent être corrigées par la suite. Ceci est encore plus vrai pour la troisième catégorie où les paraphrases ne sont pas des sorties du système. Le réglage du compromis, servant à définir ce qu'est une paraphrase correcte pour une application donnée, est donc dépendant de la tâche.

Comme mentionné plus haut, nous distinguons deux types de problèmes de production. En fonction de la tâche, les mots modifiables peuvent être imposés [Sekine, 2005; Callison-Burch et coll., 2006; Max, 2008]. C'est le cas, par exemple, pour le problème de la traduction automatique où les mots ou expressions hors vocabulaire sont à modifier. D'autres tâches n'imposent pas le lieu de la phrase d'origine à modifier [Barzilay et McKeown, 2001; Zhao et coll., 2009]. C'est le cas de la compression de texte par exemple. Deux types de générateurs potentiellement différents et difficilement comparables entre eux correspondent à ces deux types de problèmes. C'est pourquoi nous ne caractérisons pas cette séparation comme un objectif de la production de paraphrases mais comme deux classes de problèmes distincts.

Dans chacune de ces applications, nous retrouvons bien les trois objectifs et la contrainte proposés pour la production de paraphrases. La prise en compte de ces trois objectifs est donc primordiale pour la conception des modèles de production et leur évaluation comme l'a illustré la section 5.4.

6.2 COMPARER DES GÉNÉRATEURS DE PARAPHRASES

La section précédente montre qu'il n'est pas suffisant de comparer deux générateurs sans prendre en compte la tâche à laquelle ils sont destinés. Il existe peu de protocoles d'évaluation orientés vers la tâche – comme le mentionne Callison-Burch et coll. [2008] – et la comparaison pour toutes les tâches possibles n'est pas envisageable. Or, il est toujours souhaitable de pouvoir comparer deux méthodes de production de façon générale. Dans cette optique, nous proposons une tâche générique pour la comparaison de deux méthodes.

Le critère générique est choisi en fonction des observations suivantes.

Premièrement, le critère de tâche ne peut pas porter sur le sens de la paraphrase produite puisque celui-ci est déjà fortement contraint par la relation de paraphrase : être le plus proche possible du sens de la phrase d'origine. L'objectif lié à la tâche est alors principalement relié à la forme de la paraphrase.

Deuxièmement, la mesure du critère doit donner une indication sur la difficulté de la tâche résolue. Afin de pouvoir comparer finement deux systèmes, le critère générique doit être mesurable sur une échelle avec le plus de gradations significatives possibles. Dans le cas contraire, un critère trop simple comme *La paraphrase est-elle différente de la phrase d'origine ?* et deux réponses possibles – oui, non – ne permettrait pas de différencier la plupart des systèmes.

Troisièmement, dans beaucoup d'applications, le critère lié à la tâche ne peut pas être mesuré simplement à partir de la paraphrase. C'est particulièrement le cas pour les applications d'aide à l'écriture. Afin que le critère soit utilisable dans tous les travaux sur la production de paraphrases, il est important que celui-ci reste simple à calculer et ne nécessite pas de ressources externes ou de connaissances spécifiques.

Compte tenu de ces réflexions, nous proposons la variabilité comme critère de tâche générique. L'objectif des systèmes serait donc de produire les paraphrases les plus variées possibles. Comme souhaité, ce critère ne porte que sur la forme de la phrase. Il est nécessaire de supposer la conjecture 6.1 suivante vraie pour que la mesure de variabilité reflète la difficulté du problème résolu par un système de production de paraphrases.

Conjecture 6.1 *Il est, en général, plus difficile de construire un générateur capable de produire une paraphrase correcte très différente de la phrase d'origine qu'une paraphrase très proche.*

L'expérience de la section 5.4 semble étayer cette conjecture. Le corollaire 6.1 de cette conjecture permet d'assurer le caractère générique du critère de variabilité.

Corollaire 6.1 *Un générateur capable de produire des paraphrases correctes très différentes de la phrase d'origine est tout aussi capable, en général, de produire des paraphrases correctes moins différentes.*

Cette propriété générique n'est pas universelle : de nombreuses applications peuvent préférer de petites transformations. Par exemple, l'ajout d'une virgule peut ajouter une pause dans une phrase synthétisée et améliorer grandement sa qualité acoustique. En revanche, un système performant pour cette tâche générique sera aussi capable de produire des paraphrases moins variées mais plus adaptées à une tâche pratique.

De plus, nous faisons la conjecture suivante :

Conjecture 6.2 *Il existe, en général, plus de paraphrases acceptables pour une phrase longue que pour une phrase courte.*

Par exemple, il nous semble raisonnable que plus une phrase est longue et plus il y a de chance qu'elle contienne des mots possédant des synonymes. Ceci revient à dire que plus la phrase d'origine est longue et plus il est facile de produire une paraphrase avec un nombre donné de transformations élémentaires (insertion, suppression ou remplacement d'un caractère). La mesure de la variabilité doit donc être indépendante de la longueur de la phrase d'origine.

Il reste à définir une mesure pour ce critère de variabilité. La distance d'édition sur les caractères mesure le nombre minimal d'opérations élémentaires – insertion, suppression ou remplacement d'un caractère – à réaliser pour passer de la phrase d'origine à la paraphrase [Levenshtein, 1966]. Maximiser le critère de variabilité consiste à maximiser la distance d'édition. Nous préférons une distance sur les caractères, plutôt que sur les mots, afin de favoriser le remplacement d'un mot en entier sur un simple changement de flexion. Cette distance qui porte sur la forme est calculable de façon naïve en $\mathcal{O}(|Origine| \times |Paraphrase|)$, où $|x|$ est la longueur de la phrase x en caractères. Cette mesure est simple à calculer et ne nécessite aucune autre ressource que la phrase d'origine et la paraphrase.

En revanche, la distance d'édition dépend de la longueur des phrases : pour une paire de phrases donnée, sa valeur maximale est la longueur de la phrase la plus longue. Il est donc difficile d'interpréter une distance d'édition moyenne. Pour que la mesure de variabilité valorise plus une transformation sur une phrase courte

que sur une phrase longue, nous devons utiliser une variante « normalisée » de la distance d'édition.

Il est possible de normaliser la distance d'édition tout en conservant ses propriétés [Yujian et Bo, 2007], mais il semble souhaitable que l'ajout d'un caractère à une phrase d'origine donnée ait le même poids que la suppression ou la modification d'un caractère. De plus, le problème de la production de paraphrases n'est pas symétrique. La phrase d'origine pourrait être considérée comme une référence. C'est pourquoi nous proposons d'utiliser, comme mesure de la variabilité, la distance d'édition normalisée par la longueur de la phrase d'origine. Cette mesure est connue sous le nom de taux d'erreurs en caractères (TEC) :

$$\text{TEC}(\text{Paraphrase}|\text{Origine}) = \frac{\text{distance d'édition}(\text{Paraphrase}, \text{Origine})}{|\text{Origine}|}$$

Ce n'est plus une distance car il n'y a plus de symétrie. En revanche, elle peut être interprétée comme le taux de caractères modifiés dans la phrase d'origine pour produire la paraphrase. Notons qu'elle peut prendre des valeurs plus grandes que 1. Ceci se produit lorsque la paraphrase est plus de deux fois plus longue que la phrase d'origine. Afin d'éviter ces cas, nous bornons cette mesure et introduisons la mesure de la variabilité suivante :

$$\mu_{\text{var}}(\text{Paraphrase} \mid \text{Origine}) = \min\left(\frac{\text{distance d'édition}(\text{Paraphrase}, \text{Origine})}{|\text{Origine}|}, 1\right)$$

Le critère de variabilité et cette mesure satisfont les contraintes posées en début de section. Plus les valeurs moyennes de μ_{var} sont importantes, meilleur est le système de production de paraphrases.

Les performances, en ajoutant μ_{var}, des deux systèmes de la section 5.4 sont synthétisées dans le tableau 6.1. Comme attendu, le système *Test* affiche des performances plus faibles pour le critère de tâche générique, ce qui le rend objectivement moins intéressant pour beaucoup d'applications.

Les trois critères ont une variation entre 0 et 1, où 1 est la meilleure valeur. De plus, ces critères sont opposés d'après la conjecture 6.1. À l'image des mesures du rappel et de la précision en recherche d'information, les capacités respectives de différents systèmes de production de paraphrases peuvent être résumées en prenant la moyenne harmonique (f-mesure) des mesures que nous avons identifiées.

6.3 UNE APPROCHE DIFFÉRENTE

Les expériences du chapitre 5 ont montré les limites du modèle standard de la production de paraphrases par méthodes statistiques. En particulier, la section 5.3 a présenté les limites d'un décodeur classique, fondé sur l'algorithme de Viterbi. Notons que le modèle des paraphrases est fortement contraint par le paradigme de production fondé sur l'exploration d'un treillis. Ce couplage fort entre le modèle des paraphrases – présenté dans le section 4.1 – et le paradigme de production empêchent l'expérimentation de nouveaux outils visant à améliorer les générateurs de paraphrases.

Nous avons montré dans la section 6.1 que la prise en compte simultanée des trois critères est nécessaire à l'évaluation correcte des paraphrases. Elle est

Système	*Référence*	*Virgule*
Sens préservé	32 %	56 %
Syntaxe correcte	40 %	76 %
μ_{var} moyen	**20,8 % ±12,1**	**4,0 % ±3,6**
Moyenne harmonique des trois objectifs	**9,6 %**	**3,7 %**
Syntaxe correcte et sens préservé	22 %	55 %
μ_{var} moyen des paraphrases correctes	**17,8 % ±7,4**	**3,8 % ±2,1**
Moyenne harmonique des paraphrases correctes	**9,8 %**	**3,6 %**
Kappa	0,63 (p-valeur $< 10^{-3}$) Accord substantiel	

TABLEAU 6.1: L'ajout du TEC moyen montre que le système *Virgule* – qui ne travaille que sur les virgules, comme il est décrit dans la section 5.4.1 – est moins intéressant que le système *Référence*.

tout aussi importante lors de la production. En général, les systèmes réalisent une première étape de production selon des critères de conservation du sens et de naturalité. Puis, lors d'une seconde étape, ils réordonnent les paraphrases produites en fonction d'un critère lié à la tâche [Cahill et coll., 2009]. Nous avons montré dans l'expérience de la section 5.2 que cette approche n'est pas satisfaisante au vue de la variabilité des paraphrases proposées par les générateurs de paraphrases. En traduction statistique, il a été montré par [Osborne, site internet] que décoder en utilisant uniquement une table de traduction et utiliser un modèle de langue pour réordonner les traductions candidates fournit de moins bons résultats qu'un décodage utilisant conjointement les deux modèles. Nous pensons qu'il en est de même pour la production de paraphrases et le critère de tâche, et qu'il est nécessaire d'adapter le modèle de production afin d'optimiser conjointement les trois objectifs.

Le but de la présente section est de proposer un paradigme de production complètement découplé du modèle général de la paraphrase. Il sera alors possible de modifier le modèle des paraphrases et de remettre en cause certaines des hypothèses fortes présentées dans la section 4.1.

Dans la vision classique du problème de production de paraphrases, la phrase d'origine et la table de paraphrases sont utilisées pour fabriquer un treillis des paraphrases possibles. Pour chaque suite de mots possibles dans la phrase d'origine, la table de paraphrases fournit un ensemble de « segments paraphrases ». L'ordre des mots de la phrase d'origine permet de construire un ordre partiel sur les « segments paraphrases » ce qui forme le treillis de production. L'objectif du décodage est de trouver le meilleur chemin dans ce treillis. La figure 6.2 illustre cette approche par un exemple.

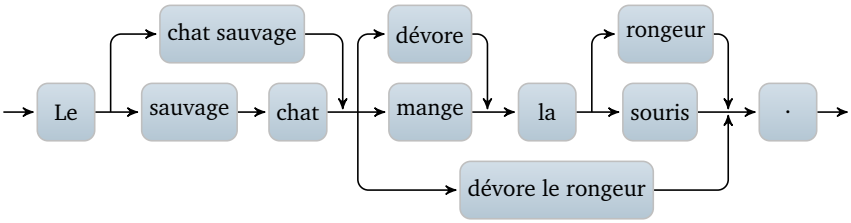
L'approche par exploration d'un treillis pose plusieurs problèmes. Premièrement, la construction du treillis repose sur un ordre partiel sur les mots de la phrase d'origine. C'est cet ordre qui impose un décodage de la gauche vers la droite de la phrase (ou inversement). Si cet ordre peut se justifier pour la traduction automatique – où l'ensemble de la phrase est à modifier – il semble moins pertinent

Le sauvage chat mange la souris.

(a) Phrase d'origine

Le ||| Le ||| P(Le|Le,B)
sauvage ||| sauvage ||| P(sauvage|sauvage,B)
chat ||| chat ||| P(chat|chat,B)
mange ||| mange ||| P(mange|mange,B)
la ||| la ||| P(la|la,B)
souris ||| souris ||| P(souris|souris,B)
. ||| . ||| P(.|.,B)
sauvage chat ||| chat sauvage ||| P(sauvage chat|chat sauvage,B)
mange ||| dévore ||| P(mange|dévore,B)
souris ||| rongeur ||| P(souris|rongeur,B)
mange la souris ||| dévore la souris ||| P(mange la souris|dévore la souris,B)

(b) Table de paraphrases



(c) Treillis de production

FIGURE 6.2: Illustration de la production de paraphrases par parcours de treillis. Le problème consiste à trouver le chemin optimal selon une fonction de score donnée.

pour la production de paraphrases. Ainsi, même si le nombre de mots modifiés dans une paraphrase est faible, le système devra tout de même parcourir les éléments « identité » pour les parties non modifiées de la phrase d'origine. Deuxièmement, les algorithmes efficaces pour chercher le meilleur chemin dans un treillis imposent plusieurs contraintes sur le modèle d'évaluation des paraphrases. Nous en avons déjà mise en lumière certaines dans les sections 4.4 et 5.3.3. Entre autres, la fonction de score doit être incrémentale ce qui empêche le calcul d'un score global pendant le décodage.

Pour proposer un paradigme différent de production de paraphrases, nous partons du constat que la phrase d'origine est proche d'une paraphrase valide : elle reste la meilleure paraphrase en termes de conservation du sens. En revanche, elle n'est souvent pas satisfaisante au regard de la tâche visée – d'où l'usage d'un générateur. Dans notre approche, la production de paraphrases consistera à appliquer à la phrase d'origine des transformations successives afin d'obtenir le meilleur score.

Définition 6.1 *Une règle de transformation est un ensemble de couples support-résultat de supports disjoints. Un support est un intervalle fermé de positions de mots de la phrase d'origine. Un résultat est la séquence de mots remplaçant les mots correspondants au support.*

Nous préférons noter le support par un intervalle plutôt que par la séquence de mots afin d'éviter les ambiguïtés si la séquence est présente plusieurs fois dans la phrase. Notons que le support fait référence aux positions dans la phrase d'origine. Si la phrase a été transformée par une règle supprimant ou ajoutant un mot, les supports font toujours référence aux mêmes séquences de mots.

Cette définition introduit donc une contrainte sur les transformations possibles :

Contrainte 6.1 *Tout mot de la phrase d'origine transformé ne peut plus être transformé.*

Une conséquence de cette contrainte est que les phrases produites sont alors indépendantes de l'ordre d'application des règles. Cette contrainte permet en plus d'assurer la finitude du nombre de paraphrases atteignables à partir d'un ensemble fini de règles.

L'exemple suivant illustre le principe des règles de transformation : pour la phrase d'origine « Le sauvage chat mange la souris. », une entrée de la table de paraphrases comme « sauvage chat ||| chat sauvage ||| P(sauvage chat|chat sauvage,B) » permet de construire une règle de transformation que nous noterons {([2; 3], **chat sauvage**)} où [2; 3] est l'intervalle de la phrase d'origine correspondant à « sauvage chat ». Comme le montre cet exemple, les entrées d'une table de paraphrases peuvent être aisément converties en règles de transformation pour une phrase d'origine donnée.

Ce formalisme permet de décrire tout type de transformation sur une phrase, y compris des transformations avec des supports discontinus. Par exemple, pour la phrase d'origine « Le chat, dans la grange, mange rapidement la souris. », la règle {([1; 2], **La souris**), ([8; 8], **est mangée**), ([10; 11], **par le chat**)} produit la paraphrase « **La souris**, dans la grange, **est mangée** rapidement **par le chat**. ».

Le problème de la production de paraphrases est alors modélisé comme un problème d'exploration dans un graphe orienté. Les nœuds du graphe sont définis ainsi :

Définition 6.2 *Un nœud est constitué d'une phrase et d'un ensemble de règles de transformation applicables à cette phrase.*

Le graphe des paraphrases est construit par explorations successives. La phrase d'origine associée à l'ensemble des règles de transformation possibles pour cette phrase forme le nœud initial. À partir d'un nœud n, on arrive à un autre nœud n' en choisissant une règle r de n, en modifiant la phrase de n conformément à la règle r et en supprimant de la liste des règles applicables toutes celles ayant un support en intersection avec le support de r.

Une règle d'arrêt – dite *stop* – est ajoutée aux nœuds où la phrase transformée est une solution potentielle. L'application de cette règle conduit dans un nœud puits où plus aucune règle n'est jouable. Dans le cadre de la production de paraphrases, cette règle *stop* est ajoutée à tous les nœuds hormis le nœud initial.

Le paradigme de la production de paraphrases par règles de transformation que nous proposons est illustré dans la figure 6.3. Il est plus adapté à la production de paraphrases que l'approche classique de la traduction statistique. En effet :

– un décodage « gauche-droite » n'est plus nécessaire : les transformations peuvent être appliquées n'importe où et dans n'importe quel ordre ;

– il n'y a pas besoin de transformer la phrase d'origine en entier : chaque nœud peut mener à un nœud *stop* ;
– les mots inconnus sont directement gérés de façon paresseuse : aucune règle n'est applicable sur un mot inconnu ;
– les transformations « identité » n'ont pas lieu d'être ;
– les seules « connaissances expertes » requises sont un ensemble de règles de transformation et une fonction de score pour les nœuds *stop*;
– tout type de règle est envisageable, y compris des règles discontinues, directement utilisables dans ce modèle ;
– il est envisageable de mélanger des règles de transformation provenant de plusieurs sources : une table de paraphrases, une mémoire de paraphrases, etc.

6.4 CONCLUSION

Dans ce chapitre, nous avons proposé des alternatives pour plusieurs aspects du cadre de la production statistique de paraphrases.

Nous avons mis en évidence trois objectifs communs aux problèmes de production de paraphrases qui se retrouvent dans toutes les applications. Nous avons pu définir la production de paraphrases comme la production d'une phrase réalisant un compromis entre trois critères :

– la conservation du sens ;
– la naturalité ;
– l'adaptation à une tâche.

Le réglage du compromis dépend de la tâche et le lieu des modifications peut être imposé ou libre.

Afin de pouvoir comparer différents systèmes, nous avons proposé une tâche générique – maximiser la variabilité des paraphrases par rapport à la phrase d'origine – et une mesure simple μ_{var} fondée sur le taux d'erreurs en caractères (TEC). La maximisation de cette mesure permet d'évaluer la capacité des générateurs à proposer des paraphrases variées. Ce critère, combiné avec la conservation du sens et la naturalité, permet de comparer les performances des générateurs de paraphrases.

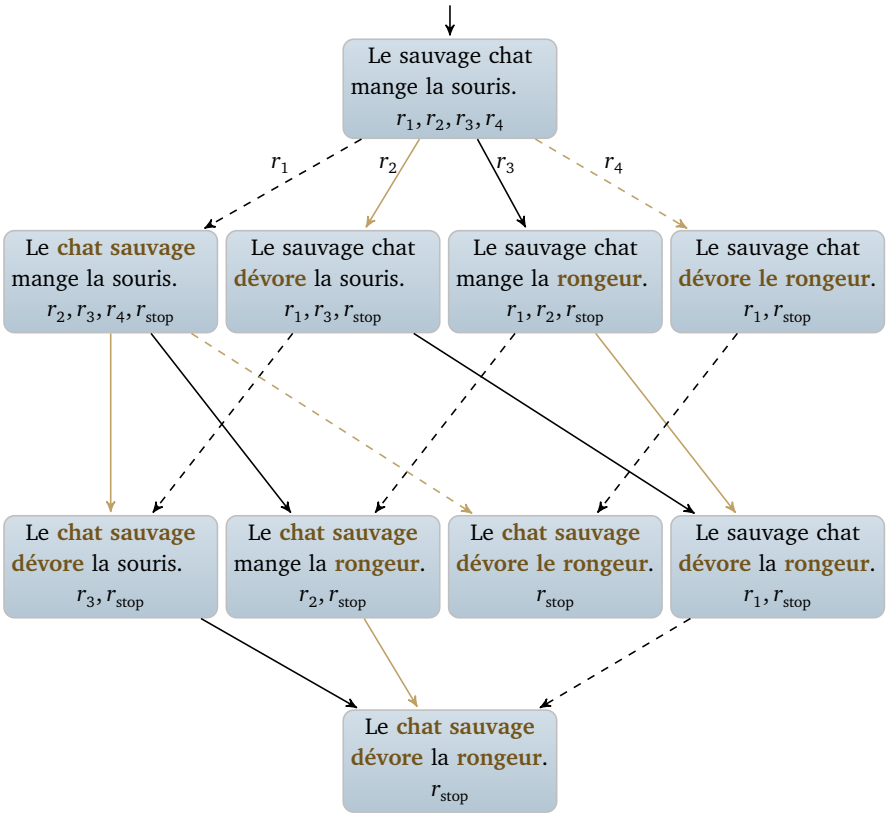
Enfin, nous avons proposé un paradigme différent, adapté à la production de paraphrases, qui consiste à voir la production de paraphrases comme l'application de règles de transformation sur la phrase d'origine. Cette approche offre en particulier l'avantage de découpler le problème de la production de paraphrases du modèle d'évaluation.

Le sauvage chat mange la souris.

(a) Phrase d'origine

r_1	{([2; 3], **chat sauvage**)}	- →
r_2	{([4; 4], **dévore**)}	→
r_3	{([6; 6], **rongeur**)}	→
r_4	{([4; 6], **dévore le rongeur**)}	- →

(b) Règles de transformations



(c) Graphe d'exploration

FIGURE 6.3: Illustration de la production de paraphrases par règles de transformation. Pour simplifier le schéma, les nœuds liés par les règles *stop* (noté r_{stop}) sont omis. Notons que certaines phrases atteignables ne sont pas des paraphrases correctes.

7 UN GÉNÉRATEUR HOLISTIQUE

Dans le chapitre précédent nous avons proposé une approche différente de l'approche classique par méthode statistique. Cette solution par application de règles de transformation est plus adaptée à la production de paraphrases. Les défauts de cette méthode sont d'ordre pratique. Premièrement, un graphe construit par exemple avec la table de paraphrases de la section 4.3, peut atteindre plusieurs milliers de branches par nœud. Deuxièmement, les outils fondés sur l'algorithme de Viterbi ne sont plus compatibles avec le paradigme proposé dans le chapitre 6. La production de paraphrases par règles de transformation s'avère donc être un problème calculatoire difficile. Toute mise en œuvre de l'approche que nous proposons implique la conception ou l'utilisation d'un algorithme d'exploration efficace.

Dans la section 7.1, nous allons proposer un nouvel algorithme, fondé sur l'échantillonnage par méthode de Monte-Carlo, pour résoudre les problèmes d'exploration de la production de paraphrases. Cet algorithme repose sur une fonction de compromis entre exploration et exploitation. C'est principalement elle qui a nécessité d'être adaptée au problème de production de paraphrases. Nous présenterons les choix que nous avons effectués à propos de cette fonction dans la section 7.2. Enfin, dans la section 7.3, nous détaillerons notre mise en œuvre de l'algorithme.

7.1 PARAPHRASES, JEU DE GO ET ÉCHANTILLONNAGE DE MONTE-CARLO

Né il y a plusieurs milliers d'années en Chine, le jeu de GO est joué au Japon depuis plus de 1 200 ans. C'est un jeu à deux joueurs. L'un après l'autre, ceux-ci doivent poser un « pion », appelé *pierre*, sur un terrain carré, appelé *goban*, composé de dix-neuf lignes et dix-neuf colonnes. Un joueur joue avec des pierres blanches, l'autre avec des pierres noires, comme le montre la figure 7.1. Nous ne

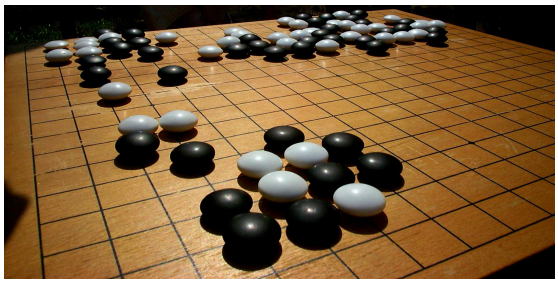


FIGURE 7.1: Photographie d'un jeu de GO [1].

détaillerons pas plus le jeu dans ce document. Les règles sont disponibles sur le site de la Fédération Française de Go [2010]. Le point important est que le GO est

1. Photographie originale de Donar Reiskoffer.

l'un des derniers jeux traditionnels où les ordinateurs n'arrivent pas à vaincre les meilleurs joueurs humains. Ceci n'est pas dû à la complexité des règles du jeu de GO, peu nombreuses et extrêmement simples. La difficulté à réaliser un programme de GO performant peut s'expliquer par deux aspects du jeu de GO :

- la combinatoire du jeu est très grande. En effet, il y a $19 \times 19 = 361$ possibilités pour le premier coup, 360 pour le second, etc. Les parties pouvant facilement dépasser les 260 coups, le nombre de parties possibles peut être très grossièrement approximé, par $\frac{360!}{100!} \approx 10^{600}$.
- la difficulté à évaluer des parties non finies. En effet, alors que donner le vainqueur en fin de partie est une tâche très simple, déterminer quel joueur est en avance en cours de partie peut s'avérer extrêmement difficile. Ce problème vient en partie de l'impact global que peuvent avoir certains coups. Dans certaines configurations, un coup peut jouer sur une situation locale mais aussi avoir des répercussions sur une situation de l'autre coté du goban.

L'écrasante supériorité de l'Homme sur les machines, pour le jeu de GO, est sur le point de disparaître. En 2006, Kocsis et Szepesvàri ont proposé un algorithme pour les problèmes de jeux à deux joueurs : RAMC (Recherche dans un Arbre par échantillonnage de Monte-Carlo) [Kocsis et Szepesvári, 2006]. Cet algorithme a plusieurs propriétés intéressantes :

- l'arbre de recherche grandit de façon non uniforme en privilégiant les séquences de coups les plus prometteuses sans élaguer de branches ;
- il n'utilise pas de connaissances expertes pour évaluer les états intermédiaires.

Ces propriétés font de RAMC un algorithme idéal pour les problèmes avec un grand facteur de branchement pour lesquels il n'existe pas de fonction d'évaluation fiable des états intermédiaires. En conséquence, des programmes fondés sur cet algorithme se sont révélés particulièrement efficaces au jeu de GO [Gelly et Wang, 2006].

À première vue, il y a peu de rapport entre le jeu de GO et la production de paraphrases. Sauf que dans notre cadre, décrit dans la section 6.3, où les paraphrases sont construites par applications successives de règles de transformation, ces mêmes règles peuvent être vues comme des coups à jouer. Le « jeu » consiste alors à chercher la séquence de coups, c'est-à-dire de règles à jouer, pour maximiser la fonction du modèle de la paraphrase. Dans ce paradigme, la production de paraphrases à deux caractéristiques :

- c'est un problème avec un fort facteur de branchement. Par exemple, avec des phrases d'un vingtaine de mots et la table de paraphrases présentée dans la section 4.3, le nombre de règles utilisables depuis la racine est compris entre cinq cents et mille ;
- nous ne disposons pas de fonction d'évaluation intermédiaire. En effet, comme nous l'avons signalé dans la section 6.3, afin de ne pas contraindre le modèle de la paraphrase, nous souhaitons utiliser une approche holistique, c'est-à-dire qui considère la paraphrase dans son ensemble.

Nous sommes bien dans une configuration où les algorithmes de type RAMC ont prouvé leur efficacité. En revanche, il existe deux différences majeures entre le jeu de GO et la production de paraphrases :

- l'absence d'un second joueur pour la production de paraphrases ;
- l'arbre est beaucoup moins profond pour la paraphrase que pour le jeu de GO. En effet, alors que pour le jeu de GO il peut atteindre plusieurs centaines de

coups, celui de la paraphrase est borné par le nombre de mots dans la phrase d'origine ;

Nous verrons dans la section 7.2 comment nous traitons l'absence de second joueur. L'impact de la différence de profondeur des arbres sera exhibé lors des évaluations de notre programme dans le chapitre suivant.

Dans cette section, nous proposons une description de l'algorithme RAMC que nous avons adapté pour la production de paraphrases. Nous nous plaçons dans le cadre décrit dans la section 6.3 où les paraphrases sont construites par applications successives de règles de transformation. Nous présenterons d'abord, dans la section 7.1.1, l'algorithme RAMC. Puis, dans la section 7.1.2, nous illustrerons son fonctionnement avec un exemple.

7.1.1 *Fonctionnement de l'algorithme*

Nous allons maintenant décrire le fonctionnement de l'algorithme RAMC. Un schéma simplifié de l'algorithme est présenté dans la figure 7.2.

L'étape d'échantillonnage constitue le principal élément de l'algorithme. Dans la figure 7.2, cette étape correspond au rectangle d'indice 1. Lors d'une itération de l'étape d'échantillonnage, le programme réalise une simulation qui consiste à construire une séquence de nœuds-règles, $n_0, r_0, n_1, r_1, \ldots, n_S$, partant du nœud initial n_0 et terminant dans un nœud *stop* n_S. Une simulation est construite itérativement à partir du nœud initial. Pour poursuivre la construction d'une simulation à partir de la séquence $n_0, r_0, \ldots, r_{i-1}, n_i$, il y a deux façons de choisir la prochaine règle r_i conduisant au nœud n_{i+1}. Dans la figue 7.2, ce choix correspond au losange d'indice 2.

La première méthode est choisie lorsque le nœud n_i n'a jamais été exploré dans une simulation antérieure. Dans la figure 7.2, cette étape correspond au rectangle d'indice 3. Dans ce cas, le score de ce nœud va être estimé par échantillonnage. L'algorithme sélectionne donc la prochaine règle aléatoirement, parmi les règles applicables depuis ce nœud. À partir de ce moment, toutes les règles sont choisies de cette façon jusqu'à la fin de la simulation, c'est-à-dire jusqu'à ce qu'un nœud *stop* soit atteint. Cette phase constitue l'échantillonnage de Monte-Carlo. Afin de réduire l'espace nécessaire en mémoire, seul le premier nœud du parcours aléatoire est considéré comme ayant été exploré pour les simulations ultérieures. Ainsi, après k simulations, l'algorithme conserve des informations sur seulement k nœuds. Ceci permet d'éviter de stocker des informations sur des nœuds dans des branches rarement explorées, donc moins prometteuses.

La seconde méthode de sélection est utilisée lorsque le nœud courant n_i a déjà été exploré lors d'une simulation. Dans la figure 7.2, cette étape correspond au rectangle d'indice 4. Dans ce cas, le programme dispose déjà d'informations sur les nœuds atteignables. La prochaine règle est donc choisie parmi les différentes règles utilisables, selon un compromis entre exploration et exploitation :

- l'exploitation consiste à encourager le tirage de la meilleure règle trouvée lors des simulations précédentes afin d'essayer de l'améliorer.
- l'exploration consiste à essayer les règles qui ont été moins choisies lors des simulations précédentes afin de voir si elles ne conduisent pas à une bonne solution.

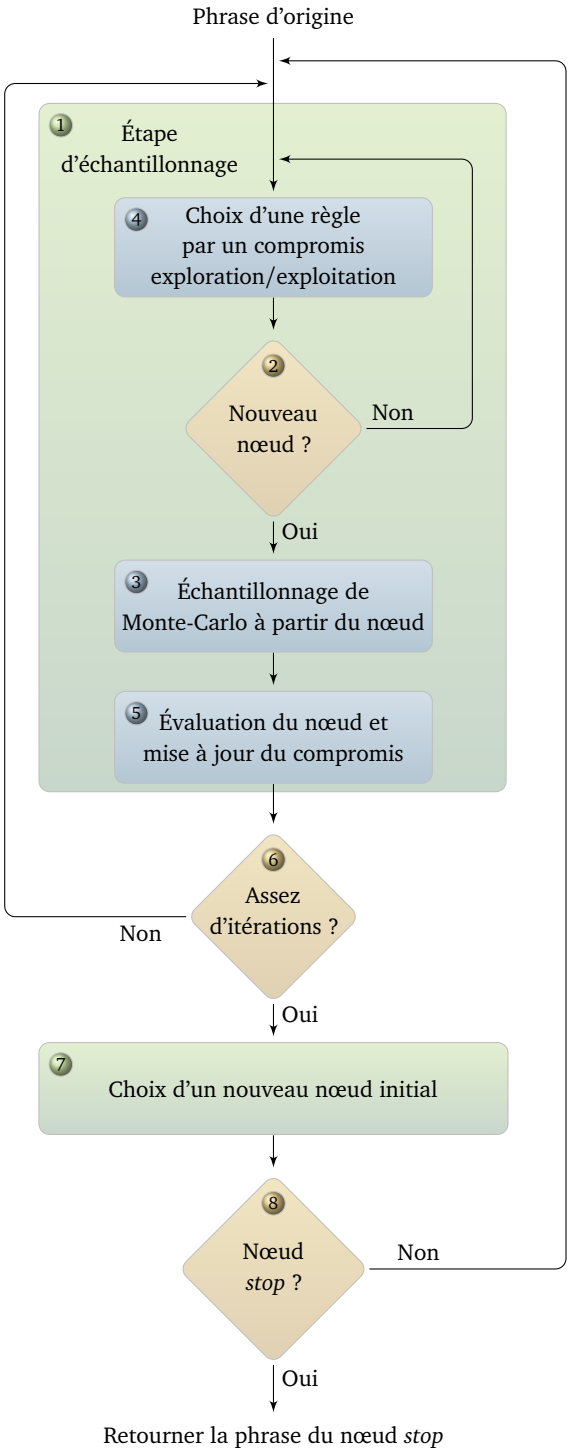


FIGURE 7.2: Schéma simplifié de l'algorithme RAMC.

Nous discutons de cette fonction de sélection dans la section 7.2.

À la fin de chaque simulation, le nœud *stop* terminant la simulation est évalué à l'aide de la fonction de score σ liée au problème. Dans la figure 7.2, cette étape correspond au rectangle d'indice 5. Pour la paraphrase, σ peut être la fonction de score du modèle, présentée dans la section 4.1. Il est à noter que cette évaluation est possible car la fonction de score dispose d'une solution complète, par définition des nœuds *stop*. Cette approche holistique permet donc à la fonction σ de calculer des traits globaux sur le nœud. Le score calculé lors de cette étape est utilisé pour mettre à jour tous les nœuds de la simulation qui ont été atteints lors du tirage d'une règle selon un compromis exploration/exploitation. Cette mise à jour consiste à prendre en compte la simulation dans les compromis exploration/exploitation qui seront faits lors des simulations suivantes. Comme cette procédure dépend de la fonction « compromis » choisie pour l'algorithme, nous la détaillerons aussi dans la section 7.2.

L'étape d'échantillonnage est itérée jusqu'à ce que le nombre de simulations soit jugé suffisant. Dans la figure 7.2, ce choix correspond au losange d'indice 6. Il existe plusieurs critères d'arrêt envisageables : il peut dépendre de la stabilité des scores depuis le nœud initial, du temps de calcul, du nombre de simulations, etc. Pour notre première mise en œuvre de l'algorithme, nous avons souhaité utiliser un critère simple : après un nombre fixé η de simulations, l'étape d'échantillonnage prend fin. La valeur de η détermine l'effort calculatoire que peut fournir l'algorithme pour résoudre le problème demandé. Pour la production de paraphrases, nous fixons empiriquement $\eta = 150\,000$ simulations.

Une fois l'échantillonnage terminé, l'algorithme sélectionne une règle disponible depuis l'état initial. Dans la figure 7.2, cette étape correspond au rectangle d'indice 7. Cette règle est celle avec le score le plus élevé, indépendamment de la composante exploration. Elle est appliquée sur le nœud initial et le nœud atteint devient le nouveau nœud initial. Si le nœud atteint est un nœud *stop*, le programme termine en retournant la solution associée. Sinon, l'étape d'échantillonnage reprend à partir de ce nœud considéré comme un nouveau nœud initial. Dans la figure 7.2, ce choix correspond au losange d'indice 8.

7.1.2 *Itération de l'étape d'échantillonnage sur un exemple*

Comme nous venons de l'expliquer, l'étape d'échantillonnage est la plus importante de l'algorithme RAMC. Afin d'aider la compréhension de cette étape, nous illustrerons notre descriptif en reprenant l'exemple développé dans la section 6.3.

Pour cette exemple, nous souhaitons produire une paraphrase de la phrase « Le sauvage chat mange la souris. ». Pour ce faire, nous disposons des quatre règles de transformation suivantes :

r_1 {([2; 3], **chat sauvage**)}
r_2 {([4; 4], **dévore**)}
r_3 {([6; 6], **rongeur**)}
r_4 {([4; 6], **dévore le rongeur**)}

Les illustrations des différentes étapes de l'algorithme sont présentées à la figure 7.3.

La figure 7.3a représente l'état initial d'une simulation. Une simulation commence avec le nœud racine comme nœud courant. Pour cet exemple, nous supposons

que le nœud racine a déjà été exploré lors de simulations précédentes. Des scores, s_1, s_2, s_3 et s_4, sont donc respectivement associés aux quatre règles r_1, r_2, r_3 et r_4 utilisables depuis la racine.

L'étape suivante, à savoir le comportement de l'algorithme depuis un nœud exploré précédemment, est représentée à la figure 7.3b. Lorsque le nœud courant a déjà été exploré, comme c'est le cas ici avec le nœud racine, l'algorithme sélectionne une règle en utilisant une fonction de compromis entre exploration et exploitation. Comme nous le verrons dans la section suivante, cette fonction utilise les scores s_1, s_2, s_3 et s_4 et peut aussi utiliser des informations comme le nombre d'explorations réalisées pour chaque règle. Dans cet exemple, nous supposons que la règle sélectionnée est r_2. Cette règle est appliquée sur le nœud courant et conduit à un nouveau nœud. Comme ce nœud a été atteint suite à un compromis exploration/exploitation, il est conservé en mémoire. Le nœud courant devient alors « Le sauvage chat **dévore** la souris. ». Puisque le quatrième mot a été modifié en appliquant la règle r_2, la règle r_4 n'est plus utilisable depuis ce nœud, conformément au modèle présenté dans la section 6.3. En effet, r_4 modifie aussi le quatrième mot. En revanche, puisque ce nœud est différent de la phrase d'origine, une règle d'arrêt de l'algorithme r_{stop} est désormais utilisable. Pour notre exemple, nous supposons que ce nœud n'a jamais été exploré lors d'une itération précédente. Il n'y a donc pas de score associé à chaque règle ce qui empêche d'utiliser le compromis exploration/exploitation.

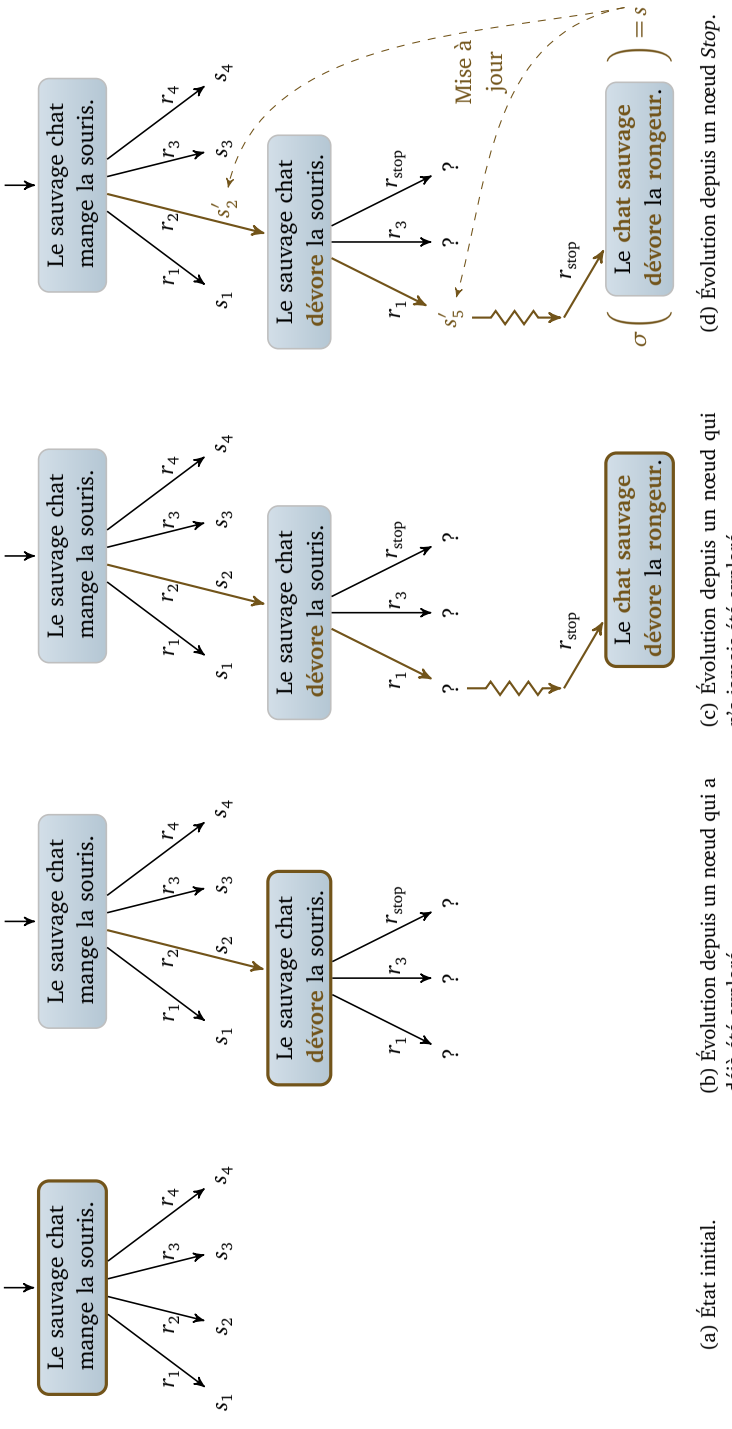
L'étape suivante, à savoir le comportement de l'algorithme depuis un nœud qui n'a jamais été exploré, est représentée à la figure 7.3c. Lorsque le nœud courant n'a jamais été exploré, comme c'est le cas maintenant, l'algorithme poursuit sur un mode d'exploration aléatoire. Une règle est tirée aléatoirement parmi celles utilisables et est appliquée sur le nœud courant. Cette opération est répétée jusqu'à ce qu'une règle r_{stop} soit utilisée. Les nœuds rencontrés lors de ce parcours aléatoire ne sont pas conservés en mémoire. Pour cet exemple, nous supposons que la règle r_1 est tirée, suivie de la règle r_3 et enfin la règle r_{stop}. Ceci nous conduit au nœud « Le **chat sauvage dévore** la **rongeur**. ».

L'étape suivante, à savoir le comportement de l'algorithme depuis un nœud r_{stop} est représentée à la figure 7.3d. Lorsque le nœud courant est un nœud r_{stop}, comme maintenant, celui-ci est évalué globalement à l'aide de la fonction σ spécifique au problème. Cette dernière retourne un score s utilisé pour mettre à jour les scores des nœuds rencontrés dans cette simulation. Les nœuds traversés lors du parcours aléatoire ne sont pas mis à jour. Dans notre exemple, le score de r_2 du nœud racine et le score de r_1 du nœud suivant sont mis à jour. Nous détaillerons cette procédure de mise à jour dans la section suivante.

La simulation est maintenant terminée. Un seul nœud a été ajouté à l'arbre d'exploration et un nœud final a été évalué globalement. L'information provenant de cette évaluation a été utilisée pour guider les prochaines simulations vers les nœuds les plus prometteurs. L'algorithme va désormais lancer une nouvelle simulation ou choisir un nouveau nœud racine en fonction de son critère d'arrêt.

7.2 COMPROMIS EXPLORATION/EXPLOITATION ET MISE À JOUR

Comme nous venons de le voir, lorsque la construction d'une simulation arrive dans un nœud n exploré précédemment, l'algorithme doit choisir la prochaine



(a) État initial.

(b) Évolution depuis un nœud qui a déjà été exploré.

(c) Évolution depuis un nœud qui n'a jamais été exploré.

(d) Évolution depuis un nœud *Stop*.

FIGURE 7.3: Illustration de l'étape d'échantillonnage sur un exemple.

règle r_n^* suivant un compromis exploration/exploitation. C'est cette fonction qui détermine l'efficacité de la convergence de l'algorithme. Nous présentons dans cette section la construction de la fonction de compromis que nous avons directement adaptée de celle utilisée pour le jeu de GO.

Le principe du compromis exploration/exploitation de RAMC est d'attribuer un score $Q_p(n, r_i)$ à chaque règle r_i disponible depuis l'état courant. Ce score est censé refléter les chances qu'une règle donnée conduise à la solution optimale. Lorsque l'algorithme doit choisir une règle suivant ce compromis, il sélectionne donc celle de plus grand score.

Initialement, le score des règles doit être défini *a priori*. Dans notre implantation de l'algorithme, nous choisissons un majorant strict de fonction d'évaluation σ du problème. Notons que nous imposons donc à la fonction d'évaluation des paraphrases d'être bornée. La formule de mise à jour est donc la suivante :

$$Q'_p(n, r_i) = \begin{cases} Q_p(n, r_i) & \text{si } Q_p(n, r_i) \text{ est défini} \\ \max \sigma + \epsilon & \text{sinon} \end{cases} \tag{7.1}$$

avec ϵ une valeur aléatoire très faible, entre 0 et 0,01 dans notre mise en œuvre, et σ la fonction d'évaluation du problème. Pour la paraphrase, nous utilisons la fonction définie dans la section 4.1. Cette initialisation permet de forcer l'algorithme à tirer au moins une fois chaque règle depuis un nœud pour initialiser les scores, avant de commencer le compromis exploration/exploitation. D'autres formes d'initialisation sont envisageables en utilisant, par exemple, des connaissances expertes sur les différentes règles disponibles, comme les probabilités de la table de paraphrases. Le choix d'une autre forme d'initialisation ainsi que son impact sur les performances sont laissés en perspective.

Maintenant que les scores sont initialisés, il reste à définir la façon de les mettre à jour au fil des simulations ainsi que la fonction permettant le calcul du compromis exploration/exploitation. Plusieurs fonctions de compromis exploration/exploitation sont possibles. Une des stratégies les plus simples consiste à choisir de faire une exploitation en prenant $r_n^* = \underset{i}{\operatorname{argmax}} Q'_p(n, r_i)$ avec une probabilité α et de faire une exploration avec une probabilité $1 - \alpha$ en choisissant une autre règle aléatoirement. Dans notre première mise en œuvre de RAMC pour la paraphrase, nous avons adapté la formule utilisée pour le jeu de GO. Celle-ci est présentée dans la section suivante. Nous y ajoutons l'heuristique présentée dans la section 7.2.2.

7.2.1 *Avoir confiance en une règle*

Pour cette première utilisation de l'algorithme pour les paraphrases, nous avons repris la formule proposée dans Gelly et Wang [2006] pour calculer les scores du compromis exploration/exploitation. Cette fonction est appelée UCT [2] pour *borne supérieure de confiance appliquée aux arbres*. Elle utilise la fonction UCB [3] (pour *borne supérieure de confiance*) proposée par Auer et coll. [2002]. Cette dernière est conçue pour le jeu des machines-à-sous – ou bandit manchot – où plusieurs machines sont réglées avec une probabilité de gain différente. L'objectif est d'estimer

2. *Upper Confidence bound for Tree search*
3. *Upper Confidence Bound*

ces probabilités tout en minimisant les pertes, c'est-à-dire le nombre d'essais effectués. En ajoutant un terme reflétant la confiance de chaque estimateur, la formule UCB offre une stratégie pour borner logarithmiquement la perte. UCT adapte cette fonction à une problématique de maximisation d'espérance dans un arbre. Nous sommes conscients que cette fonction devrait être adaptée à cause du changement de contexte entre le problème du jeu de GO et celui de la production de paraphrases. Mais cette approche va nous servir de point de départ dans la réalisation de notre outil d'optimisation.

La fonction UCT consiste donc à choisir la règle qui maximise la fonction $Q_{\text{P}}^{\oplus}(n, r_i)$:

$$r_n^* = \underset{i}{\operatorname{argmax}} Q_{\text{P}}^{\oplus}(n, r_i) \tag{7.2}$$

avec $Q_{\text{P}}^{\oplus}(n, r_i)$ la fonction UCB proposée par Auer et coll. [2002] :

$$Q_{\text{P}}^{\oplus}(n, r_i) = Q'_{\text{P}}(n, r_i) + \sqrt{\frac{2\log(s(n))}{s(n, r_i)}} \tag{7.3}$$

où $s(n)$ est le nombre de simulations qui sont passées par le nœud n et $s(n, r_i)$ le nombre de simulations qui ont choisi la règle r_i depuis le nœud n. Le second terme de cette fonction peut être vu comme un bonus servant à favoriser les règles qui ont été faiblement explorées.

À la fin d'une simulation $n_0, r_0, n_1, r_1, \ldots, n_S$, le nœud *stop* est évalué grâce à une fonction σ. Pour chaque nœud n_k de la simulation, si la règle appliquée r_k a été choisie suivant le compromis exploration/exploitation, les différentes informations liées au nœud n_k sont mises à jour ainsi :

$$s(n_k) \leftarrow s(n_k) + 1 \tag{7.4}$$

$$s(n_k, r_k) \leftarrow s(n_k, r_k) + 1 \tag{7.5}$$

$$Q_{\text{P}}(n_k, r_k) \leftarrow \begin{cases} max(Q_{\text{P}}(n_k, r_k), \sigma(n_S)) & \text{si } Q_{\text{P}}(n_k, r_k) \text{ est défini} \\ \sigma(n_S) & \text{sinon} \end{cases} \tag{7.6}$$

La principale modification que nous réalisons par rapport à l'algorithme original fondé sur UCT est dans cette étape de mise à jour. En effet, pour la production de paraphrases, l'objectif est de trouver la paraphrase qui maximise la fonction de score et non pas de maximiser la probabilité de gagner un jeu alors qu'un adversaire cherche à vous faire perdre, comme pour le jeu de GO. Dans UCT, $Q_{\text{P}}(n_k, r_k)$ est la moyenne des valeurs de $\sigma(n_S)$ lors des différentes simulations. Cette moyenne est alors un estimateur de l'espérance de la fonction d'évaluation. Nous avons donc choisi de remplacer cette moyenne par un opérateur maximum en raison de la différence majeure entre les deux problèmes : l'absence d'adversaire. Cette modification n'est pas forcément heureuse car elle fait perdre beaucoup des propriétés mathématiques du compromis exploration/exploitation d'UCB. Nous avons néanmoins essayé cette modification en ayant conscience qu'il sera probablement nécessaire de corriger la méthode de compromis ultérieurement. Ces modifications sont laissées en perspectives de nos travaux.

7.2.2 *Ne pas prendre en compte l'ordre d'application des règles*

Lorsque l'espace de recherche est très grand, UCT peut être très lent à converger. Pour chaque nœud, il faut faire une simulation aléatoire sur toutes les règles avant de commencer une exploration suivant le compromis exploration/exploitation. L'heuristique RAVE [4] (pour *estimation rapide des valeurs des actions*) a été proposée pour résoudre ce problème [Gelly et Silver, 2007]. RAVE est une version formalisée de l'heuristique « qu'importe l'ordre des coups » déjà utilisée dans le jeu de GO [Bruegmann, 1993].Pour la paraphrase, en l'absence de joueur adverse et compte tenu de notre modèle, l'ordre d'application des règles n'est pas important. L'ajout de cette heuristique dans le compromis exploration/exploitation nous paraît donc très à propos.

L'heuristique RAVE fonctionne sur le principe suivant. Jusqu'à maintenant, seul le score des règles utilisées pendant la simulation était mis à jour. Plus formellement, pour une simulation $n_0, r_0, n_1, r_1, \ldots, n_S$ donnée, lors de la mise à jour d'un nœud n_k, seule la règle r_k est modifiée. Même si les autres règles $r_k, r_{k+1}, \ldots, r_{S-1}$ sont utilisables depuis n_k, leur score n'est pas mis à jour pour le nœud n_k puisqu'elles seraient utilisées après. En fait, si l'ordre n'importe pas, ces règles auraient aussi bien pu être utilisées depuis n_k et conduire à la même solution avec la même probabilité. L'idée de RAVE est justement de prendre en compte cette information. Notons que puisque les règles sont choisies séquentiellement, l'ordre importe toujours : le choix de la règle r_k a rendu impossible le choix d'autres règles plus tard dans la simulation, ce qui a favorisé le choix des règles $r_k, r_{k+1}, \ldots, r_{S-1}$. C'est pourquoi RAVE ne remplace pas UCT mais lui ajoute une composante qui aura de moins en moins de poids au fil des simulations.

RAVE calcule donc un second score $Q^{\oplus}_{\text{R}}(n, r_i)$ qui est défini de façon similaire à $Q^{\oplus}_{\text{P}}(n_i, r_i)$:

$$Q^{\oplus}_{\text{R}}(n, r_i) = Q'_{\text{R}}(n, r_i) + \sqrt{\frac{2\log(s_R(n))}{s_R(n, r_i)}} \tag{7.7}$$

où $s_R(n, r_i)$ est le nombre de simulations qui ont choisi la règle r_i depuis n'importe quel nœud suivant n, $s_R(n) = \sum_r s_R(n, r_i)$ et

$$Q'_{\text{R}}(n, r_i) = \begin{cases} Q_{\text{R}}(n, r_i) & \text{si } Q_{\text{R}}(n, r_i) \text{ est défini} \\ \max \sigma + \epsilon & \text{sinon} \end{cases} \tag{7.8}$$

Suite à l'introduction de RAVE, nous modifions également l'initialisation d'UCB de la façon suivante :

$$Q'_{\text{P}}(n, r_i) = \begin{cases} Q_{\text{P}}(n, r_i) & \text{si } Q_{\text{P}}(n, r_i) \text{ est défini} \\ Q_{\text{R}}(n, r_i) & \text{sinon et si } Q_{\text{R}}(n, r_i) \text{ est défini} \\ \max \sigma + \epsilon & \text{sinon} \end{cases} \tag{7.9}$$

4. *Rapid Action Value Estimate*

À la fin d'une simulation $n_0, r_0, n_1, r_1, \ldots, n_S$ des mises à jour supplémentaires des scores sont réalisées. Pour chaque nœud n_k de la simulation, les différentes informations liées au nœud n_k sont mises à jour. Ainsi pour tout r_j tel que $j \geq k$:

$$s_R(n_k) \leftarrow s_R(n_k) + 1 \tag{7.10}$$

$$s_R(n_k, r_j) \leftarrow s_R(n_k, r_j) + 1 \tag{7.11}$$

$$Q_{\text{R}}(n_k, r_j) \leftarrow \begin{cases} max(Q_{\text{R}}(n_k, r_j), \sigma(n_S)) & \text{si } Q_{\text{R}}(n_k, r_j) \text{ est défini} \\ \sigma(n_S) & \text{sinon} \end{cases} \tag{7.12}$$

Ce nouvel estimateur permet d'apprendre plus rapidement mais, comme nous l'avons fait remarqué, il est biaisé. Par exemple, si le système doit transformer la phrase entière, les choix réalisés à la fin de la simulation ont été contraints par ceux réalisés en début de simulation. Il faut donc utiliser cet estimateur uniquement lorsque l'estimateur original n'est pas fiable. Le compromis exploration/exploitation final que nous utilisons est donc une combinaison linéaire de UCB et de RAVE, proposé par [Gelly et Silver, 2007], qui favorise ce dernier lorsque le nombre de simulations est trop faible :

$$r_n^* = \underset{r_i}{\text{argmax}} \left(\alpha(n) \times Q_{\text{R}}^{\oplus}(n, r_i) + (1 - \alpha(n)) \times Q_{\text{P}}^{\oplus}(n, r_i) \right) \tag{7.13}$$

où

$$\alpha(n) = \sqrt{\frac{\beta}{3s(n) + \beta}} \tag{7.14}$$

Le paramètre β, appelé *paramètre d'équivalence*, contrôle la combinaison entre les deux méthodes d'estimation. Pour un nœud donné, lorsque le nombre de simulations est inférieur à β, alors la composante RAVE l'emporte sur la composante UCB, et inversement lorsque le nombre de simulations est supérieur. Pour un nombre d'itérations η de 1 500 000, nous fixons empiriquement $\beta = 1\,000$. L'étude de l'influence de ce paramètre est laissé en perspective.

7.3 MISE EN ŒUVRE DE L'ALGORITHME

L'algorithme présenté dans la section 7.1 nous semble une piste prometteuse pour la production de paraphrases et cette impression sera confirmée par les résultats du chapitre 8. À notre connaissance, nos travaux représentent la première tentative d'adaptation d'un tel algorithme dans le domaine du TAL et en particulier pour la production de paraphrases. C'est la raison pour laquelle nous avons conçu notre implantation de RAMC comme une plateforme d'expérimentation destinée à évoluer.

Après un premier prototype dans le langage OCAML afin de valider l'algorithme, nous avons écrit une mise en œuvre en langage C, plus performante et surtout modulaire. Ce programme que nous avons baptisé GPMC, pour *Générateur de Paraphrases par échantillonnage de Monte-Carlo*, est pensé pour être évolutif et facilement adaptable à plusieurs tâches d'optimisation. Ainsi, bien que dédié initialement à la production de paraphrases, il est possible de modifier simplement le programme pour l'utiliser dans un autre contexte, comme le jeu de GO ou la traduction automatique par exemple. Dans un tel contexte, bien qu'elles ne soient pas complètement

ignorées, les performances en termes de temps de calcul et d'occupation de mémoire ne sont pas prioritaires.

Architecturalement, GPMC est composé de quatre modules comme l'illustre la figure 7.4 :

- le programme principal ;
- les structures d'interface ;
- le module RAMC ;
- le module de tâche.

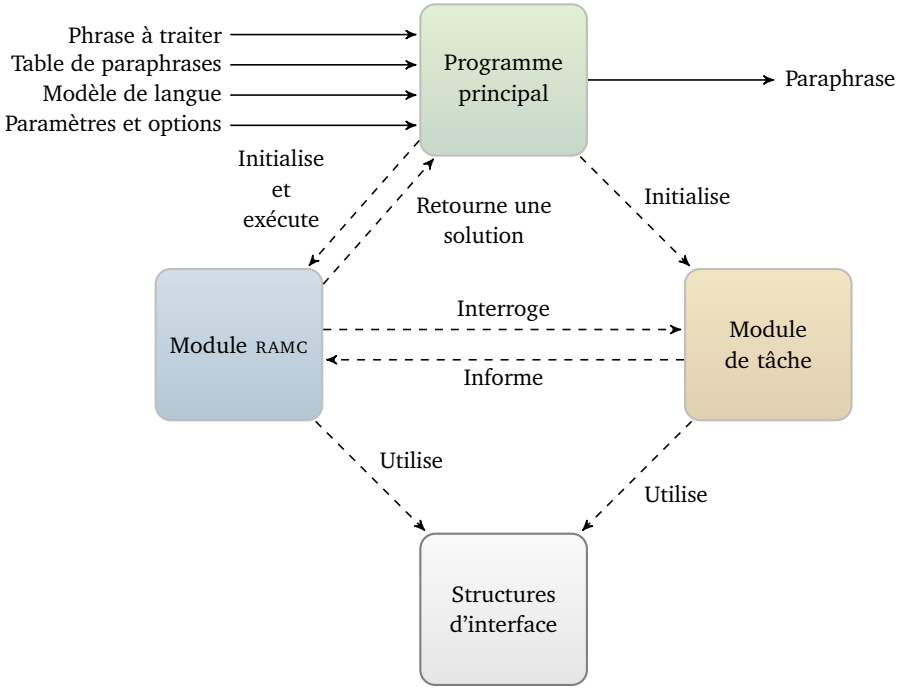


FIGURE 7.4: Architecture de GPMC. Le programme est décomposé en quatre modules pour pouvoir simplement l'adapter à différents problèmes d'optimisation.

Le programme principal coordonne la production d'une paraphrase. Il s'occupe principalement de la récupération des différents éléments nécessaires comme la phrase à traiter, l'emplacement de la table de paraphrases et l'emplacement du modèle de langue. Ce module récupère aussi les différents paramètres de l'algorithme RAMC, comme le nombre d'itérations η avant la fin de l'échantillonnage (voir section 7.1). Cette partie du programme lance les initialisations des modules de tâche et de RAMC. Enfin il lance la production de paraphrases en appelant le module RAMC et met en forme le résultat retourné pour fournir la meilleure paraphrase trouvée.

Un sous-module, appelé plus haut « structures d'interface », contient les structures pour permettre aux modules RAMC et de tâche de communiquer sans être dépendants de leur mise en œuvre interne respective. La structure et les méthodes fournies permettent principalement au module RAMC de désigner les règles de transformation sans y avoir accès. En effet, leur forme dépend du problème d'optimisation

à résoudre : par exemple, pour la paraphrase, elles correspondent à des positions et à un résultat comme le décrit la définition 6.1, alors que pour le jeu de GO une règle sera essentiellement une position du plateau. Pour conserver son indépendance vis-à-vis du problème d'optimisation, le module RAMC ne peut donc pas utiliser directement les règles de transformation.

Le module RAMC contient le cœur de l'algorithme présenté dans la section 7.1. Il s'occupe du graphe d'exploration et du compromis exploration/exploitation. Il est conçu pour être complètement indépendant du problème d'optimisation. Ce module inclut une fonctionnalité supplémentaire par rapport à l'algorithme décrit dans la section 7.1 : un « ramasse-miettes ». Cette fonction parcourt l'ensemble des nœuds du graphe qui sont conservés en mémoire et supprime ceux qui ne seront plus atteignables lors des échantillonnages futurs. La notion d'atteignabilité est définie par une fonction du module de tâche. Le ramasse-miettes est appelé après le choix d'un nouveau nœud racine. Cette fonction est « sûre » du point de vue de l'optimisation. Pour la problématique de la paraphrase, elle permet d'économiser beaucoup de mémoire : approximativement 90 % des nœuds enregistrés lors de la première série d'échantillonnages sont supprimés, ce pour un surcoût en temps de calcul négligeable.

Le module de tâche gère l'ensemble des éléments spécifiques au problème d'optimisation. Il offre un certain nombre de fonctions au module RAMC. Celles-ci permettent entre autres :

- d'évaluer un nœud *stop* par la fonction de score σ du problème ;
- de déterminer si un nœud est un nœud *stop* ;
- de fournir l'ensemble des règles utilisables depuis un nœud ;
- de fournir le nœud atteint en tirant une règle depuis un autre nœud ;
- de déterminer si un nœud est atteignable depuis la racine actuelle ;
- de réaliser une suite de tirages aléatoires à partir d'un nœud donné jusqu'à atteindre un nœud *stop* ;

Pour le problème de la paraphrase, ce module s'occupe du chargement de la table de paraphrases et du modèle de langue lors de sa phase d'initialisation. Pour cette première version, la fonction d'évaluation des paraphrases est celle proposée par le modèle de production statistique de paraphrases décrit dans le chapitre 4.

Les fonctions d'application des règles et celles servant à déterminer si une règle est applicable reprennent les principes décrits dans la section 6.3. Dans un premier temps, afin de calculer un score similaire à celui du système de référence du chapitre 4, nous imposons que toute la phrase soit « transformée » pour être évaluée, quitte à utiliser des transformations « identités ». Avec les données correspondantes, cette version de GPMC pourrait donc servir de système de traduction automatique s'il ne manquait pas un modèle de réordonnancement.

7.4 CONCLUSION

Dans ce chapitre nous avons présenté l'algorithme d'exploration RAMC construit pour le jeu de GO. Nous avons montré que la proximité entre les problématiques du jeu de GO et celles de la production de paraphrases fait que RAMC nous paraît pertinent pour corriger les limites des algorithmes fondés sur celui de Viterbi. Nous avons proposé une adaptation de l'algorithme originel pour la production de

paraphrases et nous l'avons mise en œuvre. Le programme que nous avons écrit comprend 8 000 lignes de code. Il va nous permettre de valider le paradigme proposé dans la section 6.3 ainsi que l'algorithme RAMC pour la production de paraphrases.

Comme nous l'avons signalé dans ce chapitre, certains des paramètres restent à étudier et leur impact ne sera pas traité dans ce document. C'est le cas par exemple du nombre d'itérations nécessaire, le choix de la fonction de compromis exploration/exploitation, etc.

Malgré cela, cette approche de la production nous semble très prometteuse. En effet, RAMC permet d'évaluer des solutions « complètes ». De plus, il est aussi possible d'adapter notre programme pour d'autres tâches du TAL, comme la traduction automatique, avec les mêmes bénéfices.

En permettant une évaluation globale des paraphrases, nous avons construit le premier générateur de paraphrases holistique par échantillonnage de Monte-Carlo : GPMC.

8

ÉVALUATION ET AMÉLIORATION DE NOTRE GÉNÉRATEUR HOLISTIQUE

Dans le chapitre précédent nous avons imaginé et mis en œuvre un premier générateur de paraphrases holistique. Dans ce chapitre, nous évaluerons ce générateur et proposerons plusieurs améliorations. Nous allons d'abord présenter une évaluation préliminaire dans la section 8.1. Nous améliorerons la méthode proposée dans les sections 8.2 et 8.3. Finalement, nous réaliserons une évaluation humaine des paraphrases produites dans la section 8.4.

Pour les expériences des sections 8.1, 8.2 et 8.3, nous mesurons les performances en terme de capacité d'optimisation du modèle de la paraphrase, conformément à ce que nous avons écrit dans le chapitre 4. Tous les résultats seront comparés avec ceux du système de référence du chapitre 4.

L'ensemble des évaluations fondées sur l'optimisation du score que nous présenterons est réalisé sur le jeu TEST 2. Afin d'éviter un biais, l'évaluation humaine finale est réalisée sur le jeu TEST 1. Ces deux corpus ont été présentés dans la section 3.4.

Notons que dans l'ensemble de cette section, les scores utilisés correspondent au logarithme népérien de la fonction présentée dans le chapitre 4. Bien que le problème reste le même, puisque maximiser une fonction ou son logarithme ne change pas le résultat, nous retenons ce mode de présentation pour deux raisons :

- la première est que MOSES, le générateur de référence, utilise ce modèle linéarisé par la fonction logarithme, comme nous l'avons signalé dans la section 4.5. En fait, l'ensemble des travaux sur la production statistique de paraphrases utilise aussi cette forme ;
- la seconde raison est que le score des paraphrases est extrêmement variable. En effet, celui-ci est calculé en multipliant le score de chaque entrée de la table de paraphrases utilisée. Ces scores, tous inférieurs à un, peuvent valoir jusqu'a 10^{-5} avec la table de la section 4.3. En fonction du découpage retenu, le score d'une paraphrase varie donc de plusieurs ordres de grandeur. De même, pour deux phrases d'origine différentes, les scores retournés peuvent être différents de plusieurs centaines d'ordres de grandeur. L'utilisation du logarithme permet alors de comparer les scores sur une échelle représentative des variations observées.

8.1 PERFORMANCES INITIALES ET STABILITÉ

La première expérience consiste à évaluer les capacités d'optimisation de GPMC. Puisque l'algorithme utilise une méthode d'échantillonnage aléatoire, nous souhaitons en particulier estimer la stabilité du programme.

Nous réalisons 10 exécutions de l'algorithme sur les 100 phrases du jeu TEST 2. Les résultats sont présentés à la figure 8.1. Le tableau 8.1 synthétise les différents résultats obtenus dans cette expérience.

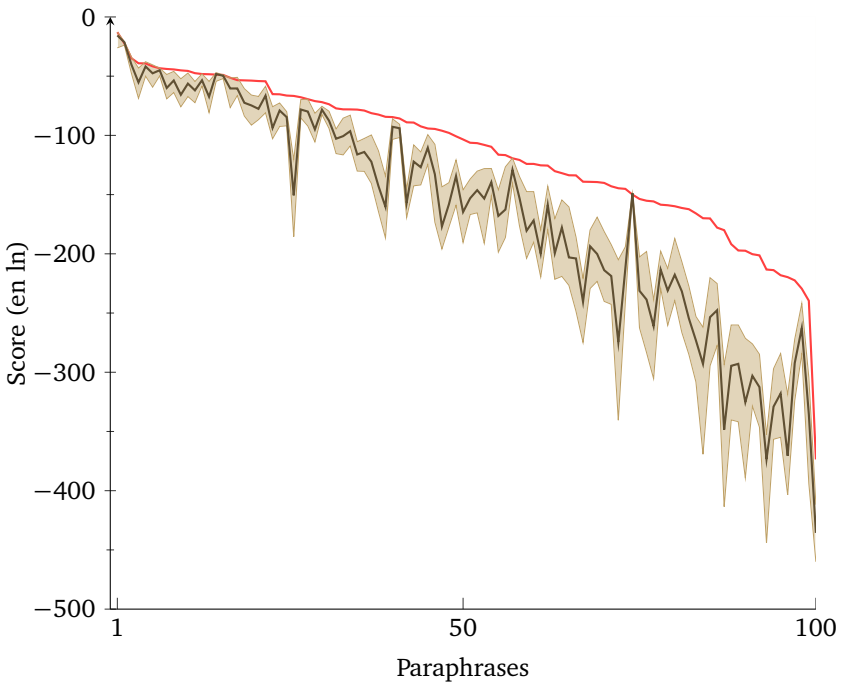


FIGURE 8.1: Capacités d'optimisation de la première version de GPMC sur 100 phrases du jeu TEST 2. En ordonnées, le logarithme du score des paraphrases produites. Les phrases sont triées par scores décroissants pour le système de référence – en rouge. La courbe marron correspond aux scores moyens de GPMC sur 10 exécutions. Les meilleurs et les moins bons scores produits lors de ces exécutions délimitent l'enveloppe autour de la courbe moyenne. On constate que GPMC est stable et proche de la référence lorsque la phrase d'origine a un score élevé. En revanche, l'écart augmente et la stabilité des résultats diminue pour les phrases avec un score plus faible.

La moyenne des scores pour une exécution est de $-163{,}9 \pm 1{,}9$ en logarithme, ce chiffre est une moyenne sur toutes les paraphrases et sur toutes les exécutions. Cette moyenne varie peu d'une exécution à l'autre, comme le montre l'écart-type de 1,9. En revanche, l'écart-type moyen pour chaque phrase est de 13,4. Il est relativement important au regard des scores moyens. L'algorithme manque clairement de stabilité pour une même phrase. Le score moyen est à comparer avec le score du système de référence qui est de $-113{,}8$. Ceci montre que pour 150 000 échantillonnages, GPMC, dans sa version de base, reste nettement moins performant que MOSES en terme d'optimisation.

La figure 8.1 montre que l'écart de performances entre GPMC et MOSES est lié à la valeur du score : plus le score de MOSES est faible et plus l'écart est important. Il en est de même avec la stabilité des résultats de GPMC. Cette évolution est en fait liée à la longueur des phrases. En effet, plus la phrase d'origine est longue et plus son score est faible. Ceci s'explique par le nombre de segments de la décomposition

Système	MOSES	GPMC
Moyenne des scores	−113,8	−163,9±1,9
Écart-type moyen par phrase	–	13,4
Temps d'exécution moyen	1,0±1,1	477,1±531,5

TABLEAU 8.1: Capacités d'optimisation de la première version de GPMC sur 100 phrases du jeu TEST 2.

optimale qui dépend souvent du nombre de mots de la phrase. Or le score est calculé à partir du produit des scores des segments qui sont chacun inférieur à 1. Nous pouvons le vérifier en constatant que la corrélation entre le score du système de référence et la longueur des phrases en mots est de −0,77. Nous constatons que la corrélation entre GPMC et la longueur des phrases passe à −0,91. GPMC est donc plus dépendant à la longueur des phrases que MOSES. Ceci est illustré par la corrélation de 0,97 entre la longueur des phrases et l'écart entre les scores de MOSES et de GPMC. Ces résultats se retrouvent dans les figures 8.2a et 8.2b.

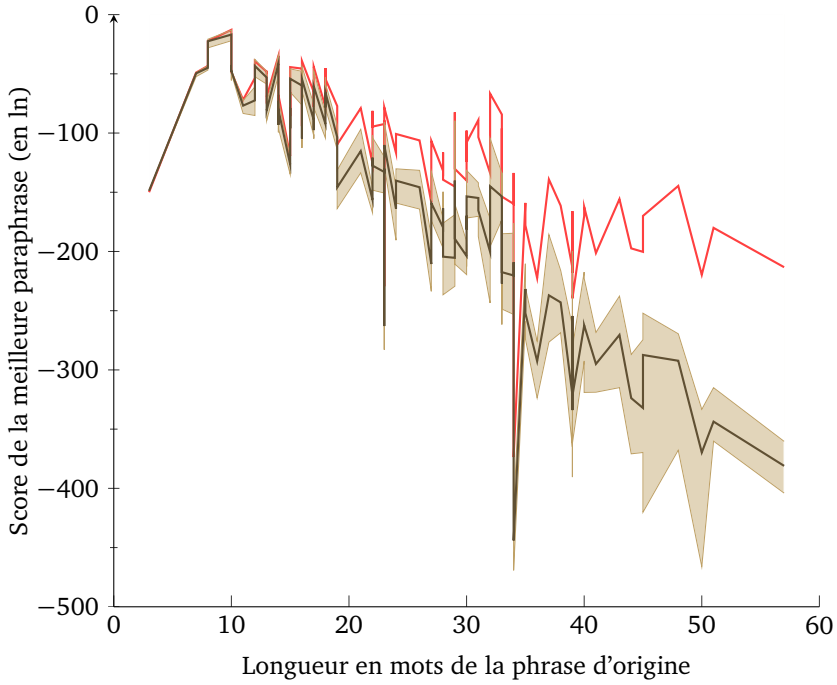
Même si à ce stade, le temps de calcul n'est pas une priorité, il reste un paramètre utile à observer. Rappelons que notre mise en œuvre de GPMC n'a pas été spécifiquement conçue pour réduire le temps de calcul comme indiqué dans la section 7.3. En particulier, certaines structures de données ont été conçues pour faciliter les modifications de l'algorithme, au détriment de la vitesse de calcul.

Le temps moyen de production d'une phrase pour MOSES est de 1,0±1,1 seconde sur une machine récente avec un processeur quadri-cœurs à 2,8 GHz et de 12 Go de mémoire vive. Notre algorithme est beaucoup plus lent avec un temps de calcul de 477,1±531,5 secondes. En particulier, le chargement en mémoire de la table de paraphrases et du modèle de langue prend 49,5±8,0 secondes dans GPMC. Ce temps ne se retrouve pas dans les statistiques de MOSES car celui-ci charge une seule fois le modèle de langue et utilise un chargement dynamique de la table de paraphrases.

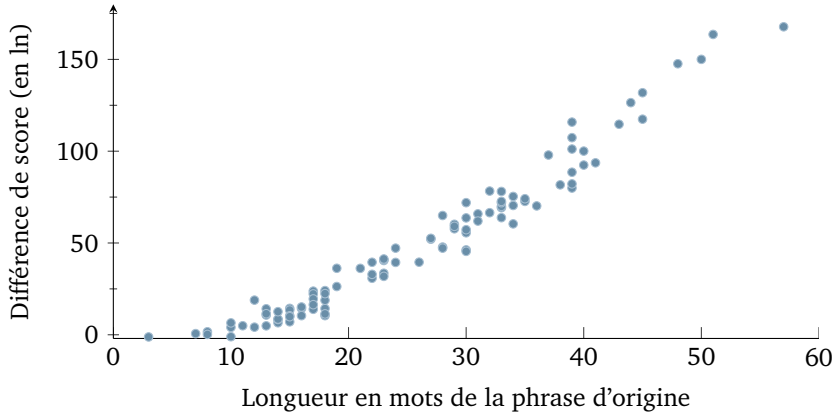
La corrélation entre le temps de calcul et la longueur des phrases d'origine nous informe sur la complexité observée de l'algorithme. Cette corrélation est de 0,88, ce qui est important. La figure 8.3 donne les temps d'exécution en fonction de la longueur en mots de la phrase d'origine. La courbe d'équation $y = 32{,}56 \times 1{,}09^x$ obtenue par régression (coefficient de détermination $R^2 = 0{,}99$) reflète la complexité temporelle de l'algorithme.

8.2 AJOUT DU SCORE DE DÉCOUPAGE OPTIMAL

Comme nous l'avons vu dans la section 5.3.1, il est possible de calculer le score d'une paraphrase potentielle *a posteriori*. Il suffit pour cela de disposer de la phrase



(a) Graphique 8.1 tracé en fonction de la longueur des phrases en mots. La largeur de l'enveloppe marron s'élargie avec la longueur de la phrase. Ceci montre que plus la phrase d'origine est longue et moins l'algorithme est stable.



(b) Différence entre MOSES et les scores moyens de GPMC en fonction de la longueur de la phrase. Plus la phrase d'origine est longue et moins les scores retournés par GPMC sont bons.

FIGURE 8.2: Performances de GPMC en fonction de la longueur des phrases en mots. Plus la phrase est longue et plus les résultats de GPMC sont instables et éloignés de la référence.

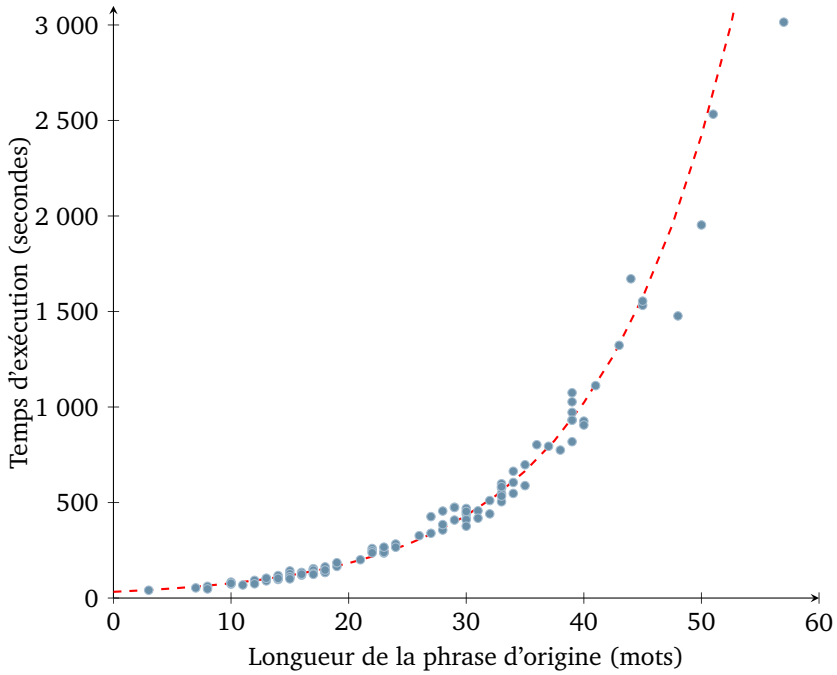


FIGURE 8.3: Relation entre temps de calcul et longueur des phrases pour la première version de GPMC. La courbe de régression en rouge reflète la complexité temporelle de l'algorithme.

Système	MOSES	GPMC	GPMC et découpage optimal
Moyenne des scores	−113,8	−163,9±1,9	−163,6±1,7
Écart-type moyen par phrase	–	13,4	14,0
Temps d'exécution moyen	1,0±1,1	477,1±531,5	431,9±478,4

TABLEAU 8.2: Comparaison des différents systèmes. L'ajout du score de découpage optimal dans GPMC n'améliore par les performances.

d'origine, de la paraphrase potentielle et de la table de paraphrases. Nous proposons donc d'utiliser l'algorithme 1 de la section 5.3.1 dans GPMC.

Théoriquement, l'utilisation de cet algorithme pour l'évaluation de chaque nœud devrait augmenter le temps d'exécution du programme. D'un autre côté, les scores associés seront toujours ceux correspondant au découpage optimal et ne dépendront plus du chemin qui a conduit à un nœud. Nous pouvons donc espérer une amélioration des scores obtenus. De plus, il est possible de fusionner les nœuds conduisant à la même forme de surface. Nous pouvons en fait espérer une réduction du temps de calcul grâce à ce dernier point. Enfin, il n'est plus nécessaire de « transformer » l'intégralité de la phrase d'origine pour évaluer une solution potentielle. Dans ce cas, nous nous retrouvons dans le cadre décrit à la section 6.3 où tous les nœuds, sauf le nœud initial, peuvent conduire à un nœud *stop*.

Pour cette expérience, nous utilisons uniquement le meilleur découpage plutôt que le score véritable afin de pouvoir comparer les scores avec ceux du système de référence. De façon analogue à la figure 8.1 précédente, la figure 8.4 présente les résultats sur le jeu TEST 2.

La moyenne des scores pour une exécution est de −163,6±1,7 en logarithme. Comme précédemment, ce chiffre est une moyenne sur toutes les paraphrases et sur toutes les exécutions. Il n'y a donc pas d'amélioration significative sur les scores moyens de la précédente version qui étaient, rappelons le, de −163,9±1,9.

Du point de vue de la stabilité, l'écart-type moyen pour chaque phrase passe de 13,4 à 14,0. L'utilisation du score de découpage optimal en l'état n'a donc pas conduit aux résultats escomptés.

Une amélioration est cependant observée pour le temps de calcul moyen puisqu'il passe de 477,1±531,5 à 431,9±478,4 secondes.

Le tableau 8.2 synthétise ces différents résultats. L'utilisation du score de découpage optimal seul ne semble pas avoir d'impact positif sur GPMC si ce n'est en termes de temps de calcul.

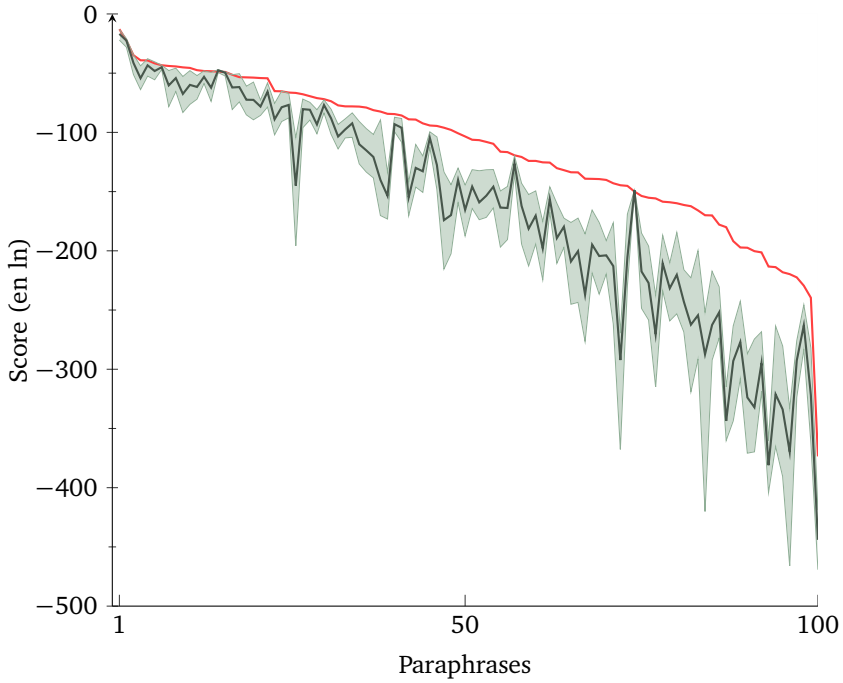


FIGURE 8.4: Évaluation de GPMC après l'ajout du score de découpage optimal. En ordonnées, le logarithme du score des paraphrases produites. Le système de référence est en rouge. La courbe verte correspond aux scores moyens de GPMC sur 10 exécutions. Les meilleurs et les moins bons scores produits délimitent l'enveloppe autour de la courbe verte. Il n'y a presque pas de différence avec la version précédente de GPMC (voir figure 8.1).

8.3 RÉDUCTION DE L'ESPACE D'EXPLORATION

Lors des différentes expériences, nous avons constaté que le système continue à explorer des parties du graphe où tous les nœuds *stop* ont déjà été évalués. En effet, si le score est suffisamment différent de ceux des autres branches, le compromis exploration-exploitation favorise pendant un certain temps la meilleure branche même si elle a été complètement explorée.

Nous proposons une nouvelle heuristique qui n'ajoute aucune approximation supplémentaire. Elle consiste à maintenir l'état des nœuds *stop* atteignables depuis un nœud. De la sorte, tout nœud *stop* est considéré comme complètement exploré dès qu'il a été exploré une fois. Si le nœud précédent dans l'échantillonnage ne conduit qu'à des nœuds complètement explorés, alors il est lui aussi considéré comme complètement exploré. Il transmet ce changement d'état au nœud précédent de l'échantillonnage qui pourra lui aussi éventuellement changer d'état.

Nous modifions alors le compromis exploration/exploitation dans notre mise en œuvre pour prendre en compte cette information. Au début d'un échantillonnage, nous interdisons le tirage de nœuds complètement explorés. Notons qu'ils

peuvent toujours être choisis comme nouveau nœud initial à la fin d'une série d'échantillonnages.

Cette nouvelle heuristique permet théoriquement de maximiser l'apport d'informations de chaque échantillonnage en réduisant l'espace d'exploration. En conjonction avec l'algorithme de score de découpage optimal, elle devrait donc permettre d'améliorer les performances, la stabilité et le temps de calcul de GPMC. Les résultats de l'utilisation de cette nouvelle heuristique sur le jeu TEST 2 sont présentés à la figure 8.5.

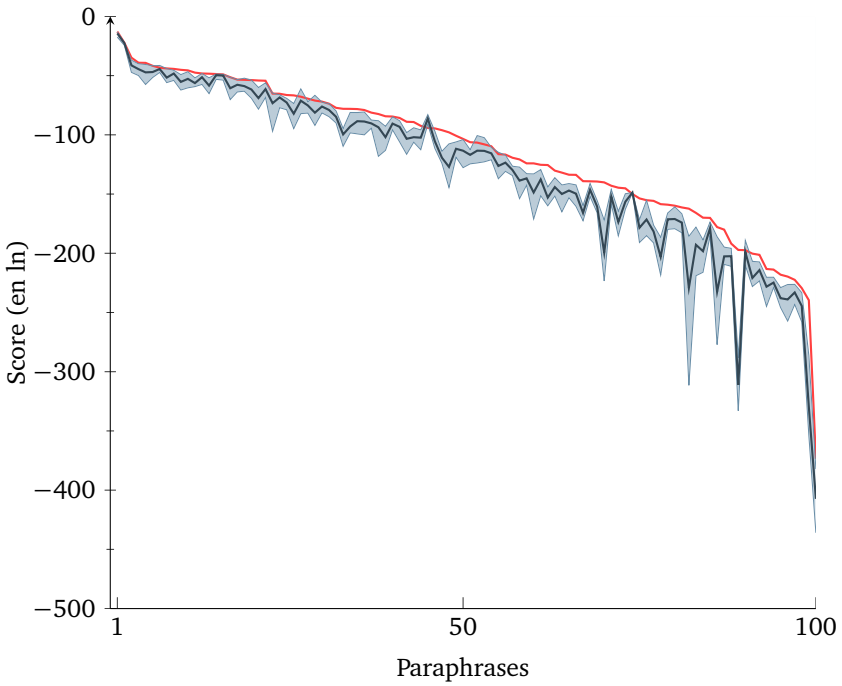


FIGURE 8.5: Évaluation de GPMC après l'ajout du score de découpage optimal et la réduction de l'espace d'exploration. En ordonnées, le logarithme du score des paraphrases produites. Le système de référence est en rouge. La courbe bleue correspond aux scores moyens de GPMC sur 10 exécutions. Les meilleurs et les moins bons scores produits lors de ces exécutions délimitent l'enveloppe autour de la courbe bleue. On constate une forte amélioration des résultats et de leur stabilité par rapport aux expériences des figures 8.1 et 8.4. Les résultats sont désormais très similaires à ceux du système de référence, voire les dépassent parfois.

Par rapport à la version précédente du système, la moyenne des scores pour une exécution passe de $-163{,}6 \pm 1{,}7$ en logarithme à $-129{,}0 \pm 0{,}7$. Cette importante amélioration permet d'avoir des résultats relativement proches de la référence qui est à $-113{,}8$. En terme de stabilité, l'écart-type moyen pour chaque phrase passe de 14,0 à 6,2, ce qui est aussi une amélioration sensible.

Notre heuristique introduit une dernière amélioration. En effet, les temps de calcul moyens sont presque divisés par deux puisqu'ils passent à $259{,}7 \pm 236{,}7$

secondes. La figure 8.6 illustre la corrélation entre temps de calcul et longueur de la phrase d'origine. Le gain est particulièrement important pour les phrases longues. Néanmoins, certaines phrases semblent ne pas bénéficier de l'amélioration. Une observation intéressante est qu'il semblerait que ces phrases correspondent aussi à celles où GPMC est nettement moins bon que MOSES. Nous n'avons pas pour le moment d'explication à ce phénomène mais l'observation précédente est sans doute une piste. Si l'on fait abstraction des neuf phrases qui ne semblent bénéficier de nos heuristiques, la courbe de régression passe de l'équation $y = 32{,}56 \times 1{,}09^x$ (coefficient de détermination $R^2 = 0{,}99$) pour le système initial à l'équation $y = 41{,}45 \times 1{,}06^x$ (coefficient de détermination $R^2 = 0{,}97$) pour le GPMC avec nos heuristiques.

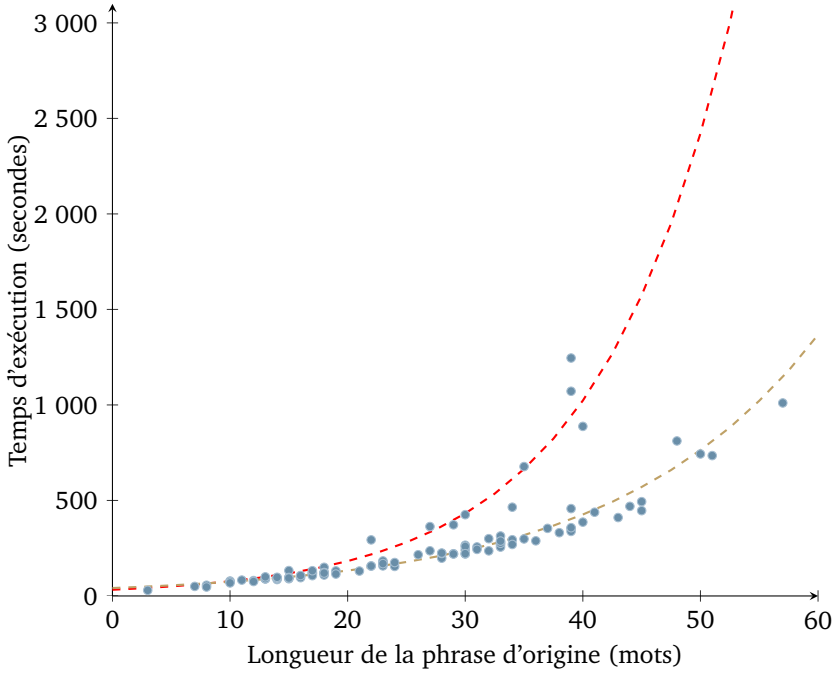


FIGURE 8.6: Temps de calcul après amélioration de GPMC, en fonction de la longueur des phrases d'origine. La courbe de régression en rouge correspond à celle de la première version de GPMC. La courbe de régression en marron correspond à celle de GPMC avec les différentes heuristiques introduites. Pour calculer cette dernière courbe de régression, nous n'avons pas pris en compte les neuf points qui semblaient suivre l'ancienne tendance. Le gain est particulièrement important pour les phrases longues.

Le tableau 8.3 récapitule les performances des différentes versions de GPMC ainsi que celles de MOSES. L'ajout de l'heuristique de réduction de l'espace de recherche conjuguée au score de découpage optimal nous permet de :

- réduire de 70 % l'écart entre GPMC et MOSES ;
- diviser par 2,2 l'instabilité de GPMC ;
- diviser par 1,8 le temps de calcul moyen de GPMC.

Cette version de notre programme atteint désormais des performances comparables avec l'état de l'art tout en offrant la possibilité d'évaluer globalement les paraphrases.

Système	MOSES	GPMC	GPMC et découpage optimal	GPMC après améliorations
Moyenne des scores	−113,8	−163,9±1,9	−163,6±1,7	−129,0±0,7
Écart-type moyen par phrase	–	13,4	14,0	6,2
Temps d'exécution moyen	1,0±1,1	477,1±531,5	431,9±478,4	259,7±236,7

TABLEAU 8.3: Comparaison des différents systèmes. L'ajout de toutes les heuristiques – réduction de l'espace de recherche et découpage optimal – à GPMC améliore fortement les performances. Elles sont maintenant comparables avec celles du système de référence.

8.4 PERFORMANCES FINALES

Il reste à mesurer l'impact des différences observées entre GPMC et MOSES sur une évaluation humaine. Nous allons donc comparer ces deux systèmes sur le jeu TEST 1. Nous allons pour ce faire reprendre le protocole présenté dans la section 3 et utilisé dans la section 5.1. Pour rappel, cent paraphrases sont produites pour chaque système. Celles-ci sont évaluées par deux juges sur des critères de conservation du sens et de grammaticalité. Les juges ont utilisé notre interface d'évaluation en ligne présentée à la section 3.2.3.

Les résultats de GPMC couplé avec ses heuristiques sont présentés dans le tableau 8.4. Les performances sont proches de ce qui est trouvé dans la section 5.1, aussi bien en termes de bonnes paraphrases qu'en termes d'accord entre les évaluateurs.

L'argumentation présentée dans la section 6.1 a montré la nécessité d'utiliser un critère de tâche pour pouvoir comparer deux systèmes de production de paraphrases. Nous utilisons ici la variabilité telle que définie dans la section 6.2. Notons qu'aucun des deux systèmes n'intègre cet objectif dans son modèle. La comparaison des systèmes est présentée dans le tableau 8.5.

Les résultats montrent que les deux systèmes sont très proches. MOSES est légèrement meilleur en particulier grâce à une plus grande variabilité dans les paraphrases produites. Malgré cela, compte tenu de la taille du jeu de test de cent phrases, les différences observées ne sont pas significatives avec un intervalle de confiance de 95 %. Ces résultats montrent que pour la même fonction d'optimisation, les capacités d'optimisation de GPMC sont comparables à l'état de l'art pour le problème de la paraphrase.

	Syntaxe correcte		Sens préservé		Paraphrase correcte	
	non	oui	non	oui	non	oui
non	59	4	50	7	67	8
oui	3	34	10	33	3	22
Kappa	0,75 (p-valeur $< 10^{-3}$) Accord substantiel					

TABLEAU 8.4: Évaluation humaine du générateur de paraphrases statistique GPMC avec l'intégration du score de découpage optimal et l'heuristique de réduction d'espace, sur le jeu TEST 1.

Système	MOSES	GPMC
Sens préservé	32 %	34 %
Syntaxe correcte	40 %	33 %
μ_{var} moyen	20,8 % ±12,1	17,1 % ±11,9
Moyenne harmonique des trois objectifs	9,6 %	8,4 %
Syntaxe correcte et sens préservé	22 %	22 %
μ_{var} moyen des paraphrases correctes	17,8 % ±7,4	11,4 % ±8,0
Moyenne harmonique des paraphrases correctes	9,8 %	8,0 %
Kappa	0,71 (p-valeur $< 10^{-3}$) Accord substantiel	

TABLEAU 8.5: Comparaison de GPMC et de MOSES lors d'une évaluation humaine sur le jeu TEST 1. Les résultats sont très proches.

8.5 CONCLUSION

Dans ce chapitre, nous avons exploré les capacités de notre générateur holistique de paraphrases. Nous avons montré que sa version de base pouvait être fortement améliorée en calculant pour chaque paraphrase son score de découpage optimal et en ajoutant une heuristique de réduction de l'espace d'exploration.

Nous avons montré que GPMC obtient des scores similaires à ceux de l'état de l'art. Grâce à l'absence de contraintes sur la fonction de score et à ses bonnes performances d'optimisation, notre algorithme ouvre la voie vers le développement de modèles de la paraphrase plus complets et mieux adaptés.

Les expériences ont montré que la profondeur de la solution optimale dans l'arbre à un impact direct sur les capacités d'optimisation de l'algorithme : plus il faut faire de choix de règles avant d'atteindre la solution optimale et plus l'algorithme a des difficultés à la trouver. Autant, ce n'est pas un problème pour la production de paraphrases, compte tenu des spécificités que nous avons exhibées pour ce domaine, autant l'utilisation de l'algorithme pour la traduction automatique pourrait nécessiter des améliorations préalables. En revanche, comme nous l'avons signalé dans le chapitre précédent, il reste de nombreuses pistes d'amélioration : l'influence des paramètres de programme, le choix de la fonction de compromis exploration/exploitation, etc. Cette première contribution des algorithmes RAMC dans le domaine de la production de paraphrases, via notre programme GPMC, laisse présager de nombreuses améliorations futures.

CONCLUSION

Les travaux présentés dans ce mémoire, autour de la production de paraphrases pour les systèmes vocaux humain-machine nous ont permis d'apporter un certain nombre de contributions à ce domaine du traitement automatique des langues. Nous avons étudié les différents aspects de la production automatique de paraphrases – la production, l'utilisation et l'évaluation des paraphrases – et nos contributions ont rempli deux objectifs :

– démontrer les spécificités de la production de paraphrases ;
– proposer les outils prenant en compte ces spécificités.

Nous avons montré tout au long de ce document que la production de paraphrases n'était pas qu'une question de conservation du sens. Lorsque l'on souhaite modifier la forme d'un message, c'est toujours pour améliorer une de ses caractéristiques, différente de son sens. Nous avons vu de nombreuses applications envisageables dans le cadre des systèmes vocaux humain-machine. Cette « tâche » est une spécificité de la paraphrase qui ne peut-être ignorée. Pour le prouver, nous avons démontré par l'absurde qu'un protocole d'évaluation qui se focalise sur la conservation du sens peut produire des résultats aberrants. De même, nous avons constaté que séparer la tâche du système de production réduit les chances de produire des paraphrases « utiles ». Il est donc nécessaire de prendre en compte cette tâche lors de la production et de l'évaluation des paraphrases.

La production de paraphrases est souvent considérée comme un cas particulier du problème de traduction automatique. Il est donc tentant de réutiliser directement les outils et protocoles développés pour ce dernier. Mais, comme nous venons de le dire, ces outils ne prennent pas en compte les particularités fondamentales de la paraphrase. De plus, ils héritent souvent de raccourcis propres à la traduction automatique qui n'ont plus lieu d'être pour les paraphrases. Nous avons donc mis au point deux outils spécifiquement pour les paraphrases : une plateforme d'évaluation et un générateur de paraphrases.

La création d'une plateforme d'évaluation vient du constat suivant : face à l'obligation, pour le moment, d'avoir recours à une évaluation humaine, chacun a proposé son ou ses protocoles d'évaluation. L'absence de consensus est peut être aussi due à la difficulté que représente la mise en place d'une campagne d'évaluation à partir de rien. Nous avons donc réalisé une plateforme d'évaluation qui simplifie cette mise en place. L'utilisation d'une interface en ligne permet, entre autres, de dissocier géographiquement et temporellement l'administration de la campagne – ajouter une série de paraphrases à évaluer, extraire les résultats, etc. – et le travail d'évaluation des juges. Le protocole associé que nous avons conçu synthétise les propositions de différents travaux sur la paraphrase. De plus, il est fait pour être le plus simple possible pour les juges et ainsi réduire l'effort nécessaire à l'évaluation. Enfin, comme nous l'avons démontré, il est nécessaire d'intégrer une tâche lors de l'évaluation des paraphrases. Pour prendre en compte cette spécificité et afin de pouvoir comparer aisément deux générateurs de paraphrases, nous avons proposé et justifié une tâche générique, la variabilité, ainsi qu'une façon de la mesurer. Au

total, notre plateforme nous a permis de réaliser l'évaluation de près d'un millier de paraphrases.

La création d'un générateur de paraphrases vient du constat suivant : en production statistique de paraphrases, l'utilisation d'un décodeur fondé sur l'algorithme de Viterbi n'est jamais remise en cause. Nous avons montré que ce type de système implique un certain nombre de contraintes : la fonction d'évaluation doit être incrémentale, calculable avec un historique limité, la paraphrase doit être entièrement « transformée », le décodage est réalisé suivant un ordre (de gauche à droite par exemple), etc. Certaines de ces contraintes, comme l'obligation de tout « transformer », ne tiennent pas compte des spécificités de la paraphrase. D'autres, comme celles d'incrémentalité et de limitation de l'historique, vont jusqu'à limiter inutilement le modèle de la paraphrase. Ainsi, par rapport au modèle du canal bruité, nous avons prouvé que la limitation de l'historique impose une approximation qui modifie fortement les sorties du système. Nous avons démontré que sans ces contraintes, il était aisé de calculer le score « véritable » des paraphrases, au sens du modèle du canal bruité. Il nous paraît essentiel de supprimer ces contraintes pour pouvoir réfléchir librement sur le modèle de la paraphrase. Nous avons donc proposé une méthode originale, fondée sur l'échantillonnage de Monte-Carlo, pour la production de paraphrases. Celui-ci permet une autre approche de la paraphrase où cette dernière est construite par application successive de règles de transformation. Notre algorithme permet surtout de s'affranchir de toute contrainte sur la fonction d'évaluation des paraphrases et donc sur le modèle. Lors de l'évaluation de notre algorithme, nous avons montré que celui-ci produit des résultats comparables à ceux de l'état de l'art pour un même modèle.

La plateforme d'évaluation et le générateur de paraphrases devraient être rendus prochainement disponibles pour la communauté.

La production automatique de paraphrases a encore besoin de faire des progrès pour pouvoir être utilisée dans une solution industrielle. Notre algorithme de production reste perfectible, en particulier dans le choix du compromis exploration-exploitation. Malgré cela, grâce à nos travaux, il est désormais possible de travailler librement sur le modèle des paraphrases. Comme nous l'avons déjà dit, la production de paraphrases est pour nous un mélange entre trois composantes et l'amélioration des générateurs de paraphrases nous semble possible en améliorant :

- la mesure de naturalité. C'est pour nous la première chose à faire. La suppression des contraintes devrait permettre l'utilisation de modèles de langues plus précis que les modèles n-grammes ;
- la mesure de conservation du sens. D'un côté, les tables de paraphrases par langues pivots et leurs filtrages méritent d'être étudiés plus précisément. D'un autre côté, notre algorithme permet de mélanger simplement et naturellement plusieurs types de règles y compris des règles « discontinues ». Ce dernier point nous semble particulièrement prometteur ;
- la mesure d'adéquation à la tâche. Nous avons peu traité des diverses applications des paraphrases pour les systèmes vocaux humain-machine. En dehors de la synthèse vocale, nous n'avons pas expérimenté les autres tâches énoncées dans la section 1.3. Nous avons délibérément concentré nos efforts sur la problématique de production des paraphrases. Mais maintenant que la

forme de la mesure de tâche n'est plus contrainte, l'intégration de ces tâches dans les modèles sera nécessaire pour prouver l'utilité des paraphrases ;
- le réglage de compromis entre ces trois composantes. Jusqu'ici, on s'efforçait d'avoir des mesures comprises entre zéro et un et de calculer leur produit pour donner un score aux paraphrases. Une refonte de ce modèle nous semble essentielle et nous pensons que les corpus de paraphrases évalués manuellement pourraient permettre d'apprendre à régler le compromis entre le sens, la naturalité et la tâche.

Dans cette thèse, nous avons défini et délimité un problème. Puis nous avons choisi une approche et reproduit l'état de l'art afin de pouvoir étudier les forces et les faiblesses de cette solution. Ceci nous a permis de faire nos propres propositions pour lever les limitations que nous avons repérées.

Pour nous, cette étape d'appropriation est primordiale dans la démarche scientifique. Elle nous a permis de remettre en cause des hypothèses, souvent ignorées, introduites par des outils provenant d'un autre domaine. Nous sommes persuadé que les innovations sont à la croisée des mondes, mais elles nécessitent d'être adaptées aux spécificités de chaque monde. Si pour le jeu de Go et la production de paraphrases, la solution semble être d'explorer au hasard, peut-être que pour la recherche scientifique, il faut tout de même introduire un biais.

Ghislain PUTOIS, Jonathan CHEVELU et Cédric BOIDIN : Paraphrase generation to improve Text-To-Speech Synthesis. Dans *Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech)*, Tokyo, septembre 2010. International Speech Communication Association. URL http://www.interspeech2010.org/program/session.php?id=2230.

Jonathan CHEVELU, Ghislain PUTOIS et Yves LEPAGE : The True Score of Statistical Paraphrase Generation. Dans *Proceedings of the 23rd Conference on Computational Linguistics (Coling): Posters*, pages 144-152, Pékin, août 2010. Coling 2010 Organizing Committee. URL http://www.aclweb.org/anthology/C10-2017.

Jonathan CHEVELU, Yves LEPAGE, Thierry MOUDENC et Ghislain PUTOIS : L'évaluation des paraphrases : pour une prise en compte de la tâche. Dans *Actes de Traitement Automatique des Langues Naturelles (TALN)*, Montréal, juillet 2010. ATALA. URL http://www.iro.umontreal.ca/~felipe/TALN2010/Xml/Papers/all/taln2010_submission_122.pdf.

Adrien LARDILLEUX, Jonathan CHEVELU, Yves LEPAGE, Ghislain PUTOIS et Julien GOSME : Lexicons or phrase tables? An investigation in sampling-based multilingual alignment. Dans *Proceedings of the 3rd Workshop on Example-Based Machine Translation (EBMT)*, Dublin, novembre 2009. CNGL. URL http://computing.dcu.ie/~mforcada/ebmt3/.

Cédric BOIDIN, Verena RIESER, Lanneke VAN DER PLAS, Lemon OLIVER et Jonathan CHEVELU : Predicting how it sounds: Re-ranking dialogue prompts based on TTS quality for adaptive Spoken Dialogue Systems. Dans *Proceedings of the Interspeech Special Session: Machine Learning for Adaptivity in Spoken Dialogue*, pages 2487-2490, Brighton, septembre 2009. ISCA. URL http://www.interspeech2009.org/conference/programme/session.php?id=6510.

Jonathan CHEVELU, Thomas LAVERGNE, Yves LEPAGE et Thierry MOUDENC : Transformation Rules and Monte-Carlo Sampling: a Different Approach for Statistical Paraphrase Generation. Dans Hiroyuki KAMEDA, Masato TOKUHISA, Sumio OHNO, Masami SUZUKI et Jonas SJÖBERGH, éditeurs : *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 230-235, Sapporo, septembre 2009a. URL http://sig.media.eng.hokudai.ac.jp/pacling2009/index.html.

Jonathan CHEVELU, Thomas LAVERGNE, Yves LEPAGE et Thierry MOUDENC : Introduction of a new paraphrase generation tool based on Monte-Carlo sampling. Dans *Proceedings of the ACL-IJCNLP Conference Short Papers*, pages 249-252, Singapore, août 2009b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-2063.

ACADÉMIE FRANÇAISE : Dictionnaire de l'Académie française, neuvième édition, de ouvroir à parfondre. *Journal officiel*, numéro 2, avril 2006. URL http://atilf.atilf.fr/academie9.htm. (Cité à la page 27.)

Jonathan ALLEN, M. Sharon. HUNNICUTT et Dennis H. KLATT : *From text to speech : the MITalk system*. Cambridge University Press, New York, avril 1987. ISBN 978-0-521-30641-8. (Cité à la page 17.)

Automatic Language Processing Advisory Committee ALPAC : Languages and machines: computers in translation and linguistics. Rapport technique, National Academy of Sciences, Washington, 1966. (Cité à la page 40.)

Ion ANDROUTSOPOULOS et Prodromos MALAKASIOTIS : A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research*, volume 38, pages 135-187, 2010. URL http://dx.doi.org/10.1613/jair.2985. (Cité à la page 28.)

Third International Workshop on Paraphrasing (IWP), octobre 2005. Asia Federation of Natural Language Processing. URL http://nlp.nagaokaut.ac.jp/IWP2005/. (Cité aux pages 150, 152, 153, 156, 158 et 159.)

Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, juillet 2002. Association for Computational Linguistics. URL http://aclweb.org/anthology-new/P/P02/. (Cité aux pages 154 et 158.)

Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), mai 2003. Association for Computational Linguistics. (Cité aux pages 150 et 155.)

Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL), juin 2005. Association for Computational Linguistics. (Cité aux pages 149, 152 et 159.)

Proceedings of ACL-08: HLT, juin 2008. Association for Computational Linguistics. URL http://aclweb.org/anthology-new/P/P08/. (Cité aux pages 160 et 161.)

Peter AUER, Nicolò CESA-BIANCHI et Claudio GENTILE : Adaptive and Self-Confident On-Line Learning Algorithms. *Journal of Computer and System Sciences*, volume 64, numéro 1, pages 48-75, février 2002. ISSN 0022-0000. URL http://dx.doi.org/10.1006/jcss.2001.1795. (Cité aux pages 124 et 125.)

Colin BANNARD et Chris CALLISON-BURCH : Paraphrasing with bilingual parallel corpora. Dans *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)* Ass [2005], pages 597-604. URL http://dx.doi.org/10.3115/1219840.1219914. (Cité aux pages 27, 33, 34, 39, 47, 48, 51, 59, 61, 68, 72, 76, 85 et 98.)

Regina BARZILAY et Lillian LEE : Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. Dans *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* Ass [2003], pages 16-23. URL http://aclweb.org/anthology-new/N/N03/N03-1003.pdf. (Cité aux pages 36, 37, 39, 40, 42, 47, 51, 85 et 86.)

Regina BARZILAY et Kathleen R. MCKEOWN : Extracting Paraphrases from a Parallel Corpus. Dans *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 50-57. Association for Computational Linguistics, juillet 2001. URL http://dx.doi.org/10.3115/1073012.1073020. (Cité aux pages 27, 34, 48, 49, 51, 67, 91, 98 et 108.)

Alexandra BIRCH et Philipp KOEHN : Statistical machine translation: IBM models and word alignment. Dans *Presentation at Fourth Machine Translation Marathon "Open Source Tools for Machine Translation"*, Dublin, janvier 2010. URL http://www.mtmarathon2010.info/web/Program_files/birch2010wordalignment.pdf. (Cité à la page 66.)

Cédric BOIDIN, Verena RIESER, Lanneke VAN DER PLAS, Lemon OLIVER et Jonathan CHEVELU : Predicting how it sounds: Re-ranking dialogue prompts based on TTS quality for adaptive Spoken Dialogue Systems. Dans *Proceedings of the Interspeech Special Session: Machine Learning for Adaptivity in Spoken Dialogue (Interspeech)* Int [2009], pages 2487-2490. URL http://www.interspeech2009.org/conference/programme/session.php?id=6510. (Cité aux pages 88 et 106.)

Houda BOUAMOR, Aurélien MAX et Anne VILNAT : Amener des utilisateurs à créer et évaluer des paraphrases par le jeu. Dans *Démonstration à Traitement Automatique des Langues Naturelles (TALN)*. Association pour le Traitement Automatique des Langues, juin 2007. URL http://www-lipn.univ-paris13.fr/taln09/pdf/TALN_155.pdf. (Cité aux pages 29, 43 et 58.)

Houda BOUAMOR, Aurélien MAX et Anne VILNAT : Acquisition de paraphrases sous-phrastiques depuis des paraphrases d'énoncés. Dans *Actes de RECITAL-TALN 2010*. Association pour le Traitement Automatique des Langues, juillet 2010. URL http://perso.limsi.fr/hbouamor/publis/extraction-paraphrases-TALN10.pdf. (Cité à la page 68.)

Chris BROCKETT et William Bill DOLAN : Support Vector Machines for Paraphrase Identification and Corpus Construction. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 1-8. URL http://nlp.nagaokaut.ac.jp/IWP2005/pdf/brockett.pdf. (Cité à la page 29.)

Bernd BRUEGMANN : Monte-Carlo Go. http://www.cgl.ucsf.edu/go/Programs/Gobble.html, 1993. Consulté en septembre 2010. (Cité à la page 126.)

Didier CADIC, Cédric BOIDIN et Christophe D'ALESSANDRO : Vocalic Sandwich, a Unit Designed for Unit Selection TTS . Dans *Proceedings of the 10th Annual*

Conference of the International Speech Communication Association (Interspeech) Int [2009], pages 2079-2082. URL http://www.isca-speech.org/archive/interspeech_2009/i09_2079.html. (Cité aux pages 19 et 25.)

Peter CAHILL, Jinhua DU, Andy WAY et Julie CARSON-BERNDSEN : Using Same-Language Machine Translation to Create Alternative Target Sequences for Text-To-Speech Synthesis. Dans *Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)* Int [2009], pages 1307-1310. URL http://www.interspeech2009.org/conference/programme/session.php?id=3610. (Cité aux pages 35, 39, 47, 72, 98, 106 et 111.)

Chris CALLISON-BURCH : *Paraphrasing and Translation*. Thèse de doctorat, Université d'Édimbourg, Édimbourg, 2007. URL http://www.cs.jhu.edu/~ccb/publications/callison-burch-thesis.pdf. (Cité aux pages 40 et 42.)

Chris CALLISON-BURCH : Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk. Dans Philipp KOEHN et Rada MIHALCEA, éditeurs : *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 286-295. Association for Computational Linguistics, août 2009. URL http://www.aclweb.org/anthology/D/D09/D09-1030.pdf. (Cité à la page 101.)

Chris CALLISON-BURCH, Trevor COHN et Mirella LAPATA : ParaMetric: An Automatic Evaluation Metric for Paraphrasing. Dans *Proceedings of the 22nd International Conference on Computational Linguistics (Coling)*, pages 97-104. Coling 2008 Organizing Committee, août 2008. URL http://www.aclweb.org/anthology/C08-1013. (Cité aux pages 40, 47 et 108.)

Chris CALLISON-BURCH, Philipp KOEHN et Miles OSBORNE : Improved Statistical Machine Translation Using Paraphrases. Dans Moore et coll. [2006], pages 17-24. URL http://www.aclweb.org/anthology/N/N06/N06-1003.pdf. (Cité aux pages 30, 39, 106 et 108.)

Eugene CHARNIAK, Kevin KNIGHT et Kenji YAMADA : Syntax-based Language Models for Statistical Machine Translation. Dans *Proceedings of the tenth machine translation summit (MTsummit IX)*. Association for Machine Translation in the Americas, septembre 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.640. (Cité à la page 63.)

Jonathan CHEVELU, Nelly BARBOT, Olivier BOEFFARD et Arnaud DELHAY : Comparing set-covering strategies for optimal corpus design. Dans Nicoletta CALZOLARI, Khalid CHOUKRI, Bente MAEGAARD, Joseph MARIANI, Jan ODJIK, Stelios PIPERIDIS et Daniel TAPIAS, éditeurs : *Proceedings of the 6th International Language Resources and Evaluation Conference (LREC)*. European Language Resources Association (ELRA), mai 2008. ISBN 2-9517408-4-0. URL http://www.lrec-conf.org/proceedings/lrec2008/summaries/750.html. (Cité aux pages 19 et 25.)

David CHIANG : A Hierarchical Phrase-Based Model for Statistical Machine Translation. Dans *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)* Ass [2005], pages 163-170. URL http://www.aclweb.org/anthology/P/P05/P05-1033.pdf. (Cité aux pages 60 et 63.)

Jacob COHEN : A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, volume 20, numéro 1, pages 37-46, avril 1960. URL http://dx.doi.org/10.1177/001316446002000104. (Cité à la page 55.)

Fabien CROMIÈRES : *Vers un plus grand lien entre alignement, segmentation et structure des phrases*. Thèse de doctorat, Université Joseph Fourier, janvier 2010. (Cité à la page 66.)

Ido DAGAN, Oren GLICKMAN et Bernardo MAGNINI : *The PASCAL Recognising Textual Entailment Challenge*, volume 3944 de *Machine Learning Challenges. Lecture Notes in Computer Science*, pages 177-190. Springer, 2006. URL http://www.cs.biu.ac.il/~nlp/RTE1/Proceedings/dagan_et_al.pdf. (Cité à la page 28.)

William Bill DOLAN et Chris BROCKETT : Automatically constructing a corpus of sentential paraphrases. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 9-16. URL http://nlp.nagaokaut.ac.jp/IWP2005/pdf/dolan.pdf. (Cité aux pages 38, 40, 42, 48, 67 et 91.)

Florence DUCLAYE, François YVON et Olivier COLLIN : Learning paraphrases to improve a question-answering system. Dans *Proceedings of the 10th Conference of the European Chapter of the Association of Computational Linguistics (EACL)*, pages 35-41. Association for Computational Linguistics, avril 2003. URL http://staff.science.uva.nl/~mdr/NLP4QA/10duclaye-et-al.pdf. (Cité aux pages 30 et 106.)

Richard DURBIN, Sean R. EDDY, Anders KROGH et Graeme MITCHISON : *Biological Sequence Analysis*. Cambridge University Press, Cambridge, avril 1998. ISBN 978-0-5216-2971-3. URL http://dx.doi.org/10.2277/0521629713. (Cité à la page 37.)

Umberto ECO : *Dire presque la même chose* . Grasset, Paris, septembre 2007. ISBN 978-2-246-65971-6. (Cité aux pages 28 et 83.)

EUROPARL : European Parliament Proceedings Parallel Corpus 1996-2009, janvier 2010. URL http://www.statmt.org/europarl/. (Cité à la page 57.)

FÉDÉRATION FRANÇAISE DE GO : Règle française du jeu de go. http://jeudego.org/_pdf/regleGo.pdf, 2010. Consulté en septembre 2010. (Cité à la page 117.)

Hélène FRANÇOIS : *Synthèse de la parole par concaténation d'unités acoustiques : construction et exploitation d'une base de parole continue*. Thèse de doctorat, Université de Rennes 1, Lannion, décembre 2002. (Cité à la page 19.)

Atsushi FUJITA et Kentaro INUI : A Class-oriented Approach to Building a Paraphrase Corpus. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 25-32. URL http://nlp.nagaokaut.ac.jp/IWP2005/pdf/fujita.pdf. (Cité aux pages 27 et 91.)

Atsushi FUJITA, Kentaro INUI et Yuji MATSUMOTO : Exploiting Lexical Conceptual Structure for Paraphrase Generation. Dans Robert DALE, Kam-Fai WONG, Jian SU et Oi Yee KWONG, éditeurs : *Natural Language Processing – IJCNLP 2005*, volume 3651 de *Lecture Notes in Computer Science*, pages 908-919. Springer Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11562214_79. 10.1007/11562214_79. (Cité aux pages 36, 38 et 39.)

Sylvain GELLY et David SILVER : Combining online and offline knowledge in UCT. Dans *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 273-280. Association for Computing Machine, juin 2007. ISBN 978-1-59593-793-3. URL http://dx.doi.org/10.1145/1273496.1273531. (Cité aux pages 126 et 127.)

Sylvain GELLY et Yizao WANG : Exploration exploitation in Go: UCT for Monte-Carlo Go. Dans *NIPS: Neural Information Processing Systems Conference, On-line trading of Exploration and Exploitation Workshop*. Neural Information Processing Systems, décembre 2006. URL http://www.lri.fr/~gelly/paper/nips_exploration_exploitation_mogo.pdf. (Cité aux pages 118 et 124.)

Irving John GOOD : The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, volume 40, numéro 3-4, pages 237-264, décembre 1953. ISSN 0006-3444. URL http://dx.doi.org/10.1093/biomet/40.3-4.237. (Cité à la page 62.)

Samer HASSAN, Andras CSOMAI, Carmen BANEA, Ravi SINHA et Rada MIHALCEA : UNT: SubFinder: Combining Knowledge Sources for Automatic Lexical Substitution. Dans *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval)*, pages 410-413. Association for Computational Linguistics, juin 2007. URL http://www.aclweb.org/anthology/S/S07/S07-1091. (Cité aux pages 36, 37 et 39.)

Andrew J. HUNT et Alan W. BLACK : Unit selection in a concatenative speech synthesis system using a large speech database. Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 373-376. IEEE Computer Society, mai 1996. ISBN 0-7803-3192-3. URL http://dx.doi.org/10.1109/ICASSP.1996.541110. (Cité à la page 17.)

James W. HUNT et Malcolm Douglas MCILROY : An Algorithm for Differential File Comparison. Rapport technique CSTR 41, Bell Laboratories, Murray Hill, juin 1976. URL http://www.cs.dartmouth.edu/~doug/diff.ps. (Cité aux pages 49 et 56.)

Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech), septembre 2009. International Speech Communication

Association. URL http://www.interspeech2009.org/. (Cité aux pages 150 et 151.)

Hongyan JING et Kathleen R. MCKEOWN : Cut and paste based text summarization. Dans *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 178-185, San Francisco, avril 2000. Association for Computational Linguistics, Morgan Kaufmann Publishers Inc. URL http://www.aclweb.org/anthology/A/A00/A00-2024.pdf. (Cité à la page 33.)

Nobuhiro KAJI, Daisuke KAWAHARA, Sadao KUROHASHI et Satoshi SATO : Verb Paraphrase based on Case Frame Alignment. Dans *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics* Ass [2002], pages 215-222. URL http://dx.doi.org/10.3115/1073083.1073120. (Cité aux pages 34, 36, 37 et 39.)

Nobuhiro KAJI, Masashi OKAMOTO et Sadao KUROHASHI : Paraphrasing Predicates from Written Language to Spoken Language Using the Web. Dans Susan DUMAIS, Daniel MARCU et Salim ROUKOS, éditeurs : *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 241-248. Association for Computational Linguistics, mai 2004. URL http://www.aclweb.org/anthology/N/N04/N04-1031.pdf. (Cité à la page 34.)

Slava M. KATZ : Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 35, numéro 3, pages 400-401, mars 1987. ISSN 0096-3518. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1165125. (Cité aux pages 62 et 63.)

David KAUCHAK et Regina BARZILAY : Paraphrasing for Automatic Evaluation. Dans Moore et coll. [2006], pages 455-462. URL http://www.aclweb.org/anthology/N/N06/N06-1058.pdf. (Cité à la page 30.)

Maurice G. KENDALL : A New Measure of Rank Correlation. *Biometrika*, volume 1–2, numéro 30, pages 81-89, juin 1938. URL http://dx.doi.org/10.1093/biomet/30.1-2.81. (Cité à la page 93.)

Kevin KNIGHT : Squibs and Discussions: Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, volume 25, numéro 4, pages 607-615, décembre 1999. URL http://www.aclweb.org/anthology/J/J03/J03-1002.pdf. (Cité à la page 73.)

Kevin KNIGHT et Daniel MARCU : Statistics-Based Summarization - Step One: Sentence Compression. Dans *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 703-710. Association for the Advancement of Artificial Intelligence, août 2000. ISBN 978-0-262-51112-4. URL http://www.aaai.org/Conferences/AAAI/aaai00.php. (Cité aux pages 29 et 106.)

Levente Kocsis et Csaba Szepesvári : Bandit Based Monte-Carlo Planning. Dans Johannes Fürnkranz and Tobias Scheffer and Myra Spiliopoulou, éditeur : *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4212 de *Lecture Notes in Computer Science*, pages 282-293. Springer, septembre 2006. ISBN 978-3-540-45375-8. URL http://dx.doi.org/10.1007/11871842_29. (Cité à la page 118.)

Philipp Koehn : Europarl: a parallel corpus for statistical machine translation. Dans *Proceedings of the tenth machine translation summit (MTsummit X)*, pages 79-86. Asia-Pacific Association for Machine Translation, septembre 2005. URL http://www.iccs.informatics.ed.ac.uk/~pkoehn/publications/europarl-mtsummit05.pdf. (Cité aux pages 57 et 67.)

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin et Evan Herbst : Moses: Open Source Toolkit for Statistical Machine Translation. Dans Annie Zaenen et Antal van den Bosch, éditeurs : *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 177-180. Association for Computational Linguistics, juin 2007. URL http://aclweb.org/anthology-new/P/P07/P07-2045.pdf. (Cité aux pages 61, 64 et 74.)

Philipp Koehn, Franz Josef Och et Daniel Marcu : Statistical phrase-based translation. Dans *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* Ass [2003], pages 48-54. URL http://dx.doi.org/10.3115/1073445.1073462. (Cité à la page 59.)

Aleksandra Krul, Géraldine Damnati, François Yvon et Thierry Moudenc : Corpus Design Based on the Kullback-Leibler Divergence for Text-to-Speech Synthesis Application . Dans *Interspeech 2006 - ICSLP, Ninth International Conference on Spoken Language Processing*, pages 1647-1649. International Speech Communication Association, septembre 2006. URL http://www.isca-speech.org/archive/interspeech_2006/i06_1647.html. (Cité à la page 19.)

Mathieu Lafourcade : Making people play for Lexical Acquisition with the Jeux De Mots prototype. Dans *Proceedings of the Seventh International Symposium on Natural Language Processing (SNLP)*. specialty research unit in Natural language processing and Intelligent information System Technology, décembre 2007. ISBN 978-874-623-062-9. URL http://hal-lirmm.ccsd.cnrs.fr/lirmm-00200883/PDF/MLF-snlp2007-v5.pdf. (Cité à la page 58.)

J. Richard Landis et Gary G. Koch : The measurement of observer agreement for categorical data. *Biometrics*, volume 33, numéro 1, pages 159-174, mars 1977. URL http://www.jstor.org/stable/2529310. (Cité à la page 55.)

Philippe Langlais, Alexandre Patry et Fabrizio Gotti : Recherche locale pour la traduction statistique par segments. Dans *Actes de Traitement Automatique des*

Langues Naturelles (TALN), pages 119-128. Association pour le Traitement Automatique des Langues, juin 2008. URL http://www.iro.umontreal.ca/~felipe/bib2webV0.81/cv/papers/paper-taln-2008.pdf. (Cité à la page 97.)

Adrien LARDILLEUX, Jonathan CHEVELU, Yves LEPAGE, Ghislain PUTOIS et Julien GOSME : Lexicons or phrase tables? An investigation in sampling-based multilingual alignment. Dans *Proceedings of the 3rd Workshop on Example-Based Machine Translation (EBMT)*, pages 45-52. Centre for Next Generation Localisation, CNGL, novembre 2009. URL http://hal.archives-ouvertes.fr/hal-00439806/PDF/ebmt3-lardilleux.pdf. (Cité à la page 66.)

Adrien LARDILLEUX et Yves LEPAGE : Atruly multilingual, high coverage, accurate, yet simple, sub-sentential alignment method. Dans *Proceedings of AMTA 2008 The 8th conference of the Association for Machine Translation in the Americas (AMTA)* , pages 125-132. Association for Machine Translation in the Americas, octobre 2008. URL http://hal.archives-ouvertes.fr/hal-00368737/PDF/amta08-lardilleux.pdf. (Cité à la page 66.)

Yves LEPAGE et Etienne DENOUAL : Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 57-64. URL http://www.aclweb.org/anthology/I/I05/I05-5008.pdf. (Cité aux pages 30, 38, 39 et 67.)

Vladimir Iosifovich LEVENSHTEIN : Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, volume 10, numéro 8, pages 707-710, février 1966. (Cité à la page 109.)

Dekang LIN et Patrick PANTEL : Discovery of inference rules for question-answering. *Natural Language Engineering*, volume 7, numéro 4, pages 343-360, décembre 2001. ISSN 1351-3249. URL http://dx.doi.org/10.1017/S1351324901002765. (Cité aux pages 37, 39, 48 et 74.)

Aurélien MAX : Local rephrasing suggestions for supporting the work of writers. Dans Bengt NORDSTRÖM et Aarne RANTA, éditeurs : *Advances in Natural Language Processing: 6th International Conference (GoTAL)*, pages 324-335. Springer, août 2008. ISBN 978-3-540-85286-5. URL http://www.limsi.fr/Individu/amax/papers/Max_08_GoTAL.pdf. (Cité aux pages 30, 33, 37, 39, 47, 59, 72, 76, 84, 98, 106 et 108.)

Aurélien MAX : Sub-sentencial Paraphrasing by Contextual Pivot Translation. Dans Su et coll. [2009], pages 18-26. URL http://www.aclweb.org/anthology/W/W09/W09-2503. (Cité à la page 39.)

Kathleen R. MCKEOWN : Paraphrasing questions using given and new information. *Computational Linguistics*, volume 9, numéro 1, pages 1-10, 1983. ISSN 0891-2017. URL http://portal.acm.org/citation.cfm?id=973242. (Cité aux pages 36, 37 et 39.)

Igor MEL'ČUK : The Meaning-Text Approach to the Study of Natural Language and Linguistic Functional Models. *LACUS Forum 24*, volume 24, pages 5-19, 1998. URL http://www.lacus.org/volumes/index.php?volume=24. (Cité à la page 33.)

George A. MILLER : WordNet: A Lexical Database for English. *Communications of the ACM*, volume 38, numéro 11, pages 39-41, novembre 1995. ISSN 0001-0782. URL http://cacm.acm.org/magazines/1995/11/8720-wordnet/abstract. (Cité à la page 36.)

Robert C. MOORE, Jeff BILMES, Jennifer CHU-CARROLL et Mark SANDERSON, éditeurs. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, juin 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N06/. (Cité aux pages 151, 154 et 161.)

Eric MOULINES et Francis CHARPENTIER : Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, volume 9, numéro 5-6, pages 453-467, décembre 1990. ISSN 0167-6393. URL http://dx.doi.org/10.1016/0167-6393(90)90021-Z. (Cité à la page 17.)

Crystal NAKATSU et Michael WHITE : Learning to Say It Well: Reranking Realizations by Predicted Synthesis Quality. Dans Nicoletta CALZOLARI, Claire CARDIE et Pierre ISABELLE, éditeurs : *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1113-1120. Association for Computational Linguistics, juillet 2006. URL http://dx.doi.org/10.3115/1220175.1220315. (Cité aux pages 35 et 39.)

Alexis NASR : *Un modèle de reformulation automatique fondé sur la Théorie Sens Texte: Application aux langues contrôlées*. Thèse de doctorat, Université Paris 7, 1996. URL http://pageperso.lif.univ-mrs.fr/~alexis.nasr/Rech/Publi/these.pdf.gz. (Cité aux pages 29, 38 et 39.)

Franz Josef OCH : Minimum Error Rate Training in Statistical Machine Translation. Dans *Proceedings of 41th Annual Meeting of the Association for Computational Linguistics*, pages 160-167. Association for Computational Linguistics, juillet 2003. URL http://dx.doi.org/10.3115/1075096.1075117. (Cité à la page 74.)

Franz Josef OCH et Hermann NEY : A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, volume 29, numéro 1, pages 19-51, mars 2003. URL http://www.aclweb.org/anthology/J/J03/J03-1002.pdf. (Cité à la page 66.)

Takashi ONISHI, Masao UTIYAMA et Eiichiro SUMITA : Paraphrase Lattice for Statistical Machine Translation. Dans Jan HAJIČ, Sandra CARBERRY, Stephen CLARK et Joakim NIVRE, éditeurs : *Proceedings of the ACL 2010 Conference Short Papers*, pages 1-5, Uppsala, juillet 2010. Association for Computational Linguistics. URL http:

//www.aclweb.org/anthology/P/P10/P10-2001.pdf. (Cité à la page 39.)

Miles OSBORNE : Decoding for Syntax. http://www.inf.ed.ac.uk/teaching/courses/mt/lectures/scfgDecoding.pdf, site internet. Consulté en janvier 2010. (Cité à la page 111.)

Kishore PAPINENI, Salim ROUKOS, Todd WARD et Wei-Jing ZHU : Bleu: a Method for Automatic Evaluation of Machine Translation. Dans *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics* Ass [2002], pages 311-318. URL http://dx.doi.org/10.3115/1073083.1073135. (Cité aux pages 30, 42, 48 et 74.)

Martin J. PICKERING et Victor S. FERREIRA : Structural Priming : A Critical Review. *Psychological bulletin*, volume 134, numéro 3, pages 427-459, mai 2008. ISSN 0033-2909. URL http://dx.doi.org/10.1037/0033-2909.134.3.427. (Cité à la page 26.)

Vincent POLLET et Andrew BREEN : Synthesis by Generation and Concatenation of Multiform Segments. Dans *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech)*, pages 1825-1828. International Speech Communication Association, septembre 2008. URL http://www.isca-speech.org/archive/interspeech_2008/i08_1825.html. (Cité à la page 17.)

Richard POWER et Donia SCOTT : Automatic generation of large-scale paraphrases. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 57-64. URL http://nlp.nagaokaut.ac.jp/IWP2005/pdf/power.pdf. (Cité à la page 39.)

William H. PRESS, Saul A. TEUKOLSKY, William T. VETTERLING et Brian P. FLANNERY : *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2e édition, février 1993. ISBN 978-0-5214-3720-2. URL http://dx.doi.org/10.2277/0521437202. (Cité à la page 74.)

Ghislain PUTOIS, Romain LAROCHE et Philippe BRETIER : Enhanced Monitoring Tools and Online Dialogue Optimisation Merged into a New Spoken Dialogue System Design Experience. Dans *Proceedings of the SIGDIAL 2010 Conference*, pages 185-192. Association for Computational Linguistics, septembre 2010. URL http://www.sigdial.org/workshops/workshop11/proc/pdf/SIGDIAL32.pdf. (Cité à la page 26.)

Chris QUIRK, Chris BROCKETT et William DOLAN : Monolingual Machine Translation for Paraphrase Generation. Dans Dekang LIN et Dekai WU, éditeurs : *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 142-149. Association for Computational Linguistics, juillet 2004. URL http://research.microsoft.com/apps/pubs/?id=69170. (Cité aux pages 27, 39, 43, 47, 48, 51, 54, 59, 61, 72, 76, 85, 86 et 98.)

Philip E. Rubin, Thomas Baer et Paul Mermelstein : An articulatory synthesizer for perceptual research. *Journal of the Acoustical Society of America*, volume 70, numéro 2, pages 321-328, août 1981. URL http://dx.doi.org/10.1121/1.386780. (Cité à la page 17.)

Satoshi Sekine : Automatic paraphrase discovery based on context and keywords between NE pairs. Dans *Third International Workshop on Paraphrasing (IWP)* Asi [2005], pages 80-87. URL http://nlp.nagaokaut.ac.jp/IWP2005/pdf/sekine.pdf. (Cité aux pages 27, 30, 43, 91, 106 et 108.)

Claude Elwood Shannon et Warren Weaver : *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949. ISBN 0-252-72548-4. (Cité à la page 59.)

Yusuke Shinyama et Satoshi Sekine : Paraphrase acquisition for information extraction. Dans *Proceedings of the second international workshop on Paraphrasing (IWP)*, pages 65-71. Association for Computational Linguistics, juillet 2003. URL http://dx.doi.org/10.3115/1118984.1118993. (Cité à la page 29.)

Andreas Stolcke : SRILM - an Extensible Language Modeling Toolkit . Dans *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP2002 - INTERSPEECH 2002)*, pages 901-904. International Speech Communication Association, septembre 2002. URL http://www-speech.sri.com/cgi-bin/run-distill?papers/icslp2002-srilm.ps.gz. (Cité à la page 63.)

Keh-Yih Su, Jian Su, Janyce Wiebe et Haizhou Li, éditeurs. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, août 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-1. (Cité aux pages 156 et 160.)

Christoph Tillmann et Hermann Ney : Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation. *Computational Linguistics*, volume 29, numéro 1, pages 97-133, mars 2003. URL http://www.aclweb.org/anthology/J/J03/J03-1005.pdf. (Cité à la page 72.)

TLFi : Trésor de la Langue Française informatisé. http://atilf.atilf.fr/tlf.htm, site internet. Consulté en septembre 2010. (Cité à la page 28.)

UIT : *Méthodes d'évaluation subjective de la qualité de transmission (Recommandation UIT-T P.800)*. Union Internationale des Télécommunications (UIT), août 1996. URL http://www.itu.int/rec/T-REC-P.800-199608-I/fr. (Cité à la page 89.)

Ozlem Uzuner, Boris Katz et Thade Nahnsen : Using Syntactic Information to Identify Plagiarism. Dans *Proceedings of the Second Workshop on Building Educational Applications Using NLP* Ass [2005], pages 37-44. URL http://www.aclweb.org/anthology/W/W05/W05-0207. (Cité à la page 29.)

Vladimir VAPNIK : *Statistical Learning Theory*. John Wiley & Sons Inc, New York, 1998. ISBN 978-0471030034. (Cité à la page 34.)

Andrew VITERBI : Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, volume 13, numéro 2, pages 260-269, avril 1967. ISSN 0018-9448. URL http://dx.doi.org/10.1109/TIT.1967.1054010. (Cité aux pages 18 et 72.)

Daniel R. WHITE et Mike S. JOY : Sentence-based natural language plagiarism detection. *Journal on Educational Resources in Computing (JERIC)*, volume 4, numéro 4, page 2, décembre 2004. ISSN 1531-4278. URL http://dx.doi.org/10.1145/1086339.1086341. (Cité à la page 29.)

Takayoshi YOSHIMURA, Keiichi TOKUDA, Takashi MASUKO, Takao KOBAYASHI et Tadashi KITAMURA : Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis. Dans *Proceedings of the Sixth European Conference on Speech Communication and Technology (Eurospeech)*, pages 2347-2350. International Speech Communication Association, septembre 1999. URL http://www.isca-speech.org/archive/eurospeech_1999/e99_2347.html. (Cité à la page 17.)

Li YUJIAN et Liu BO : A Normalized Levenshtein Distance Metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 1091-1095, 2007. ISSN 0162-8828. URL http://dx.doi.org/10.1109/TPAMI.2007.1078. (Cité à la page 110.)

Ying ZHANG, Stephan VOGEL et Alex WAIBEL : Interpreting BLEU/NIST scores: How much improvement do we need to have a better system. Dans *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051-2054. European Language Resources Association, mai 2004. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.1732. (Cité à la page 74.)

Shiqi ZHAO, Xiang LAN, Ting LIU et Sheng LI : Application-driven Statistical Paraphrase Generation. Dans Su et coll. [2009], pages 834-842. URL http://www.aclweb.org/anthology/P/P09/P09-1094. (Cité aux pages 11, 27, 34, 38, 39, 41, 42, 59, 61, 72, 86, 87, 98, 106 et 108.)

Shiqi ZHAO, Cheng NIU, Ming ZHOU, Ting LIU et Sheng LI : Combining Multiple Resources to Improve SMT-based Paraphrasing Model. Dans *Proceedings of ACL-08: HLT* Ass [2008], pages 1021-1029. URL http://www.aclweb.org/anthology/P/P08/P08-1116. (Cité à la page 76.)

Shiqi ZHAO, Haifeng WANG, Xiang LAN et Ting LIU : Leveraging Multiple MT Engines for Paraphrase Generation. Dans Chu-Ren HUANG et Dan JURAFSKY, éditeurs : *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*, pages 1326-1334, Pékin, août 2010. Coling 2010 Organizing Committee. URL http://www.aclweb.org/anthology/C/C10/C10-1149.pdf. (Cité à la page 41.)

Shiqi ZHAO, Haifeng WANG, Ting LIU et Sheng LI : Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora. Dans *Proceedings of ACL-08: HLT* Ass [2008], pages 780-788. URL http://www.aclweb.org/anthology/P/P08/P08-1089. (Cité aux pages 37 et 39.)

Liang ZHOU, Chin-Yew LIN, Dragos Stefan MUNTEANU et Eduard HOVY : ParaEval: Using Paraphrases to Evaluate Summaries Automatically. Dans Moore et coll. [2006], pages 447-454. URL http://www.aclweb.org/anthology/N/N06/N06-1057.pdf. (Cité à la page 30.)

George Kingsley ZIPF : *The Psycho-Biology of Language*. Houghton Mifflin, 1935. (Cité à la page 62.)

RÉSUMÉ

Cette thèse s'intéresse au lien entre ce qui est prononcé et le système vocal humaine-machine qui le prononce. Plutôt que de proposer des systèmes capables de tout vocaliser, nous envisageons le message comme une variable qui peut être modifiée. L'élément primordial d'un message est son sens. Il est donc possible de changer les mots utilisés si cela conserve le sens du message et améliore les systèmes vocaux. Cette modification s'appelle « production de paraphrases ».

Dans cette thèse, nous proposons une étude de la production statistique de paraphrases pour les systèmes vocaux humain-machine. Pour ce faire, nous présentons la conception d'un système de référence et d'une plateforme d'évaluation en ligne. Nous mettons en lumière les différentes limites de l'approche classique et nous proposons un autre modèle fondé sur l'application de règles de transformation. Nous montrons qu'il est nécessaire de prendre en compte l'utilisation souhaitée des paraphrases lors de leur production et de leurs évaluations, pas uniquement du critère de conservation du sens. Enfin, nous proposons et étudions un nouvel algorithme pour produire des paraphrases, fondé sur l'échantillonnage de Monte-Carlo et l'apprentissage par renforcement. Cet algorithme permet de s'affranchir des contraintes habituelles de l'algorithme de Viterbi et donc de proposer librement de nouveaux modèles pour la paraphrase.

Mots-clés : langage naturel, traitement du (informatique) ; synthèse automatique de la parole ; systèmes conversationnels (informatique) ; optimisation combinatoire.

ABSTRACT

Paraphrase generation for human-machine voice interaction systems

This thesis focuses on the relationships between what is uttered and human-machine spoken dialogue systems that utter it. Instead of relying on all-purpose speech-synthesis engines, we consider that a message to synthesize is a variable that can be modified. As the primary characteristic of a message is its meaning, changing words so as to improve speech quality is allowable, provided that meaning is preserved. Performing such modifications is "paraphrase generation". This PhD thesis presents a study of statistical paraphrase generation for human-machine spoken dialogue systems. We first introduce to the design of a state-of-the-art paraphrase generator and an online evaluation platform. We then shed some light on some limitations of standard approaches to paraphrase generation and put forward an alternative model based on transformation rules. We show that usages of paraphrases must be taken into account during generation and evaluation, along with the meaning preservation criterion. At last, we introduce a new algorithm for paraphrase generation based on Monte-Carlo sampling and reinforcement learning. Studies of its behavior are reported. This algorithm overcomes some usual limitations of the Viterbi algorithm and paves the way for new paraphrase generation models.

Keywords: natural language processing (computer science); speech synthesis; interactive computer systems; combinatorial optimization.

Thèse en « Informatique et applications » réalisée au sein de l'équipe ISLanD, Laboratoire Greyc – Campus Côte de Nacre, Boulevard du Maréchal Juin – BP 5186 – 14032 Caen cedex