Université de Montréal

Comparaison de deux techniques de décodage pour la traduction probabiliste

Par

Ali Awdé

Département d'Informatique et de Recherche Opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc)
en informatique

Mars, 2003

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Comparaison de deux techniques de décodage pour la traduction probabiliste

Présenté par :

Ali Awdé

a été évalué par un jury composé des personnes suivantes :

François Major

……..……

(Président - rapporteur)

Philippe Langlais

……..……

(Directeur de recherche)

Jian Yun Nie

……..……

(Membre du jury)

Mémoire accepté *le 10 juillet 2003*

Résumé

La traduction automatique a connu une révolution majeure ces dernières années dans le domaine du traitement automatique des langues naturelles et de l'autre coté, les besoins en matière de traducteurs automatiques fiables augmentent sans cesse. De ce fait, nous nous sommes intéressés à ce domaine afin de concevoir un traducteur automatique basé sur un modèle statistique.

Premièrement, nous avons entraîné les trois premiers modèles d'alignement proposés par IBM tout en vérifiant l'efficacité de l'outil public GIZA.

Dans une seconde partie, nous avons concentré nos efforts sur la réalisation et la comparaison de décodeurs traductionnels. Un décodeur est un programme capable de traduire une phrase source en une phrase cible étant donné un modèle de traduction. Plus précisément, nous avons développé et comparé deux décodeurs pour les modèles de traduction IBM2. Le premier décodeur implémente une technique de programmation dynamique pour parcourir une partie importante de l'espace de recherche tandis que le second est un algorithme vorace (*greedy)* qui ne parcourt de manière adhoc qu'un très petit sous-ensemble de l'espace de recherche.

Mots clés: traduction automatique statistique, algorithme de recherche, modèle d'alignement, algorithme vorace, algorithme DP, GIZA.

Abstract

In previous years, machine translation witnessed a major revolution in the area of natural language processing and the needs for reliable automatic translators increase ceaselessly. Therefore, we interested to bend in this domain to design an automatic translator based on statistical models.

First, we trained the first three alignment models proposed by IBM while verifying at the same time the efficiency of the public toolkit GIZA.

Then, we developed the translation decoder, to which our work is more concentrated. The decoder's job is translating a source sentence to a target sentence given a model of translation. On the other hand, we developed and compared two decoders for the models of translation IBM2. The first decoder implements a technique of dynamic programming to cover an important part of the research space whereas the second is a greedy algorithm, which covers in an adhoc way only a very small subset of the research space.

Key word: Statistical Machine translation, search algorithm, IBM alignment model, greedy algorithm, DP algorithm, GIZA.

R E M E R C I E M E N T S

C’est une habitude saine que de remercier au début d’un tel travail tous ceux qui, plus ou moins directement, ont contribué à le rendre possible. C’est avec mon enthousiasme le plus vif et le plus sincère que je voudrais rendre mérite à tous ceux qui à leur manière m’ont aidé à mener à bien ce mémoire.

Je tiens d’abord à témoigner de ma plus profonde gratitude à mon directeur de recherche, monsieur Philippe Langlais. Il a su, par son extrême dévouement, sa gentillesse débordante, sa grande disponibilité et son soutien financier rendre mon travail fort agréable, voire même amusant. De conseils judicieux en mots d’encouragements, il a toujours été d’une aide précieuse et je lui en suis très reconnaissant.

Merci au département d’informatique et de recherche opérationnelle et à la Faculté des études supérieures de m’avoir si généreusement accordé une bourse de rédaction.

Je remercie tous les membres de RALI qui m’ont aidé surtout George, Simona, Michael, Elliott et Luc à évaluer mes résultats. Ainsi que tous les autres membres qui m’ont accueilli de manière très chaleureuse.

Je dois et j’adresse un remerciement tout particulier à mes amis qui me soutiennent toujours et particulièrement : Jacqueline, Hind, Mélanie et Youssef pour leurs aides dans ce mémoire (je ne peux pas nommer tous mes amis!).

Finalement, une attention toute particulière est dirigée vers ma famille, et en particulier mon père et ma mère qui n’ont pas négligé les sacrifices tout au long des mes études. Merci infiniment d’avoir toujours été si attentionnés et dévoués.

Table des matières

Résumé 3
Abstract 4
R E M E R C I E M E N T S 5
Table des matières 6
Liste de figures 8
Liste de tableaux 9
Table de notation 10
Chapitre 1 11
Introduction 11
1.1. Histoire 11
1.2. L'importance applicative. 13
1.3. Aperçu sur le contenu de mémoire : 14
Chapitre 2 16
Traduction statistique 16
2.1. L'idée originale Shanon (canal bruité) 16
2.2. Le modèle de langue 19
2.3. Les modèles de traduction 20
2.3.1. Les alignements 20
2.3.2. Idée initiale 22
2.3.3. Les modèles de traduction proposés par IBM 23
2.3.3.1. Modèle de traduction probabiliste IBM1 23
2.3.3.2. Modèle de traduction probabiliste IBM2 24
2.3.3.3. Modèle de traduction probabiliste IBM3. 25
2.4. Conclusion 26
Chapitre 3 27
L'estimation des paramètres 27
3.1. Le projet EGYPT 27
3.1.1. Bitexte 28
3.1.2. Whittle 28
3.1.3. GIZA 30
3.1.4. Cairo 31
3.2. Conclusion 32
Chapitre 4 33
Les expériences avec GIZA 33
4.1. Préparation du corpus 33
4.2. Les paramètres. 34
4.3. L'espace mémoire 35
4.4. Temps d'exécution 37
4.5. Mesure de la qualité d'un modèle de traduction 39
4.6. Quelques exemples et chiffres pour commenter les résultats : 40
4.6.1. Les alignements des mots 41
4.6.2. Les probabilités de transfert : 42
4.6.3. Les fertilités 43
4.7. Une comparaison entre RMTTK et GIZA. 45

4.8. Conclusion 47
Chapitre 5 48
Décodage. 48
5.1. L'algorithme DP 50
5.1.1. Principe 50
5.1.2. Description 50
5.1.3. Filtrage 52
5.1.4. Implémentation 56
5.1.5. Les problèmes rencontrés durant l'implémentation 57
5.1.6. Exemple 59
5.1.7. Exemples de résultats obtenus 65
5.2. L'algorithme de recherche "Greedy". 66
5.2.1. Principe 66
5.2.2. Description 67
5.2.3. Implémentation des opérations 68
5.2.4. Exemple 70
5.2.5. Le nombre d'itérations et les temps de traduction 72
5.2.6. Exemples de résultats obtenus 74
5.3. Greedy initialisé par la traduction produite par DP 75
5.3.1. Exemples de résultats obtenus 76
Chapitre 6 78
Évaluation des résultats 78
6.1. WER et SER 79
6.2. BLEU 81
6.3. Évaluations et comparaison des décodeurs 82
6.4. Évaluation humaine 83
6.5. Exemple des traductions évaluées 85
Chapitre 7 87
Conclusion 87
Bibliographie 89
Annexe1 91

Liste de figures

Figure 1: Illustration du canal bruité. 17
Figure 2: Chercher la phrase anglaise ê qui maximise la probabilité P(e|f). 17
Figure 3: Un alignement dont chaque mot français est aligné à un seul mot anglais. 20
Figure 4: Un autre alignement possible de ces phrases mais moins probable. 20
Figure 5: Un alignement dont chaque mot anglais est associé à un seul mot français. 21
Figure 6: Un alignement dont un ensemble de mots français est connecté à un ensemble de mots anglais. 21
Figure 7: Un exemple simple d'alignement, chaque mot est aligné avec sa traduction 22
Figure 8: Exemple d'un corpus bitexte d'entraînement. 29
Figure 9: Illustration du format de corpus généré par Whittle. 29
Figure 10: Un exemple de cairo. 32
Figure 11: L'espace mémoire occupé par l'entraînement de GIZA. 36
Figure 12: Le temps d'exécution de GIZA sur une machine peu puissante. 37
Figure 13: Le temps d'exécution par rapport à la taille du corpus. 38
Figure 14: La perplexité par rapport aux itérations des modèles 1, 2 et 3 entraînés sur un corpus de 352 983 paires de phrases françaises / anglaises 40
Figure 15: Le meilleur alignement obtenu pour cette paire de phrases par IBM1. 41
Figure 16: Le meilleur alignement obtenu pour cette phrase par IBM2. 41
Figure 17: Le meilleur alignement obtenu pour cette phrase par IBM3. 42
Figure 18: L'architecture de la traduction probabiliste [Nießen et al. 1998]. 48
Figure 19: L'accroissement du temps de la traduction avec la longueur de la phrase. 53
Figure 20: L'accroissement de temps avec la taille de l'ensemble de vocabulaire actif 55
Figure 21: L'algorithme de recherche DP. 56
Figure 22: La distribution de longueur des traductions anglaises des phrases françaises de 10 mots. 58
Figure 23: Le format de la matrice (mot, couverture, fertilité, score, position précédent, mot précédent), l'accès à une hypothèse se fait par (mot, couverture, ligne, colonne). 61
Figure 24: La traduction initiale par alignement un à un des mots à leur traduction la plus probable selon le modèle de transfert. 70
Figure 25: L'initialisation. 70
Figure 26: Itération 1, une permutation à la position 5 et 6. 70
Figure 27: Itération 2, une substitution à la positin 4. 71
Figure 28: À l'initialisation, le score d'alignement est -133.518. 71
Figure 29: Itération 1: Le mot (*other*) à la position 8 est aligné avec 2 mots (*d' autres*); ... 71
Figure 30: Itération 2: Substitution du mot (*the*) à la position 4 par (*some)*. 71
Figure 31: Itération 3: Permutation des mots à la position 5 et 6. 72
Figure 32: Itération 4: Substitution du mot (*things*) à la position 6 par (*solutions*). 72
Figure 33: La moyenne d'itérations, le nombre de substitutions et permutations augmentent linéairement avec la longueur de la phrase à traduire. 72
Figure 34: La distribution du nombre d'itérations et de substitutions. 73
Figure 35: Les nombres de phrases itérées par greedy+. 76
Figure 36: Les proportions d'acceptabilité de traduction des évaluateurs pour chaque décodeur. 84

Liste de tableaux

Tableau 1: Les probabilités des alignements sur un corpus de phrases de 8 mots (anglais /français). 24
Tableau 2: Les nombres de paires de phrases et les paramètres de transfert de chaque modèle de traduction. 34
Tableau 3: Les dix premières probabilités de transfert du mot *"canadian"* (modèle 2 et 3) prises des résultats d'entraînement sur un corpus de 1.6 millions de phrases. 42
Tableau 4: Les dix premières probabilités de transfert du mot *"rights"* (modèle 2 et 3) entraîné sur un corpus de 1.6 millions paires de phrases. Soit n(Φ|e) la probabilité que le mot e a une fertilité *Φ*. 43
Tableau 5: Les probabilités de transfert et de fertilité du mot "update". 44
Tableau 6: Un exemple des mots anglais ayant une large fertilité "*nodding*". 44
Tableau 7: Les temps d'exécution d'un entraînement avec GIZA et RMTTK. 45
Tableau 8: Deux mots exemples de RMTTK et GIZA, on a seulement pris les dix premières probabilités pour chaque mot. 46
Tableau 11: Les résultats de décodeur sans filtrage strict (N=50; BETA=10^6) 63
Tableau 12: Les résultats de décodeur avec un filtrage sur les nombres de mots anglais associés à chaque mot français (N=15) et un seuil (BETA=1.9) sur les hypothèses. .. 63
Tableau 13: Les résultats de décodeur avec un filtrage sur les nombres de mots anglais associés à chaque mot français (N=7) et un seuil (BETA=1.25) sur les hypothèses. .. 64
Tableau 14: des exemples de corpus test de Hansard et une comparaison avec la traduction humaine. Un filtrage a été appliqué N=10, BETA=1.5. 65
Tableau 15: Exemples de traduction, extraits d'un corpus test (N=10). Humain est la traduction produite par un traducteur. 75
Tableau 16: Les résultats de l'évaluation des décodeurs. 82
Tableau 17: Les choix des évaluateurs et ses différents avis avec les pourcentages. 85

Table de notation

e_i	Le mot anglais à la position i.
e^I	Une phrase anglaise de I mots.
I	La longueur de la phrase anglaise.
i	Position en e^I, $i=0,1,...,I$
f_j	Le mot français de la position j.
f^J	Une phrase française de J mots.
J	La longueur de la phrase française.
j	Position en f^J, $j=0,1,...,J$
$t(e_i\|f_j)$	Probabilité de transfert
a	Alignement
$a(i\|j,I,J)$	Probabilité d'alignmement
$ø_i$	Fertilité du mot e_i . (Modèle 3)
$n(ø\|e)$	Probabilité de la fertilité. (Modèle 3)

Chapitre 1

Introduction

La traduction automatique (T.A.) d'une langue humaine à une autre en utilisant les ordinateurs, désignée dans la littérature anglophone sous le terme de « *Machine Translation* » (M.T.), est un but de l'informatique depuis longtemps et à l'ère d'Internet et du commerce électronique, le besoin de communiquer rapidement dans toutes les langues devient une priorité. La mondialisation du commerce a eu des effets considérables sur l'essor de l'industrie de la langue, et plus particulièrement en traduction où la demande ne cesse de croître.

Les besoins majeurs de la traduction automatique se concentrent principalement sur la traduction de textes scientifiques, techniques, commerciaux, officiels et médicaux. La traduction d'œuvres littéraires reste assez marginale**.**

1.1. Histoire[1]

La plupart des grands projets de traduction automatique sont nés entre 1958 et 1966 des besoins de traduction à partir du russe engendrés par la guerre froide. Il existe deux générations de systèmes de traduction:

- La première génération de programmes est ce qu'on appelle des systèmes directs. Ils se basent sur des équivalences de termes, traduisent mot à mot à partir de la consultation d'un dictionnaire et ne font aucune analyse. Ce sont des systèmes bilingues (ils traitent une seule paire de langues) et unidirectionnels. Ces systèmes sont limités, mais peuvent s'avérer utiles dans certains cas d'application restreinte (avec un vocabulaire limité).

- La deuxième génération de programmes de traduction regroupe les systèmes de traduction automatique et les systèmes de transfert. Ces programmes de deuxième génération ont un principe plus complexe que celui des systèmes directs. Les systèmes de transfert sont basés sur trois modules: ***l'analyse*** du texte en langue source, le ***transfert***, et la ***génération*** dans la langue cible. Actuellement, les systèmes à architecture basée sur le transfert sont les plus

[1] Extraits pris du site du *Centre Pluridisciplinaire de Sémio linguistique Textuelle,* Université Toulouse-Le Mirail http://www.univ-tlse2.fr/gril/

couramment utilisés. Ils permettent plus facilement d'intégrer une nouvelle langue que les systèmes directs, pour lesquels ajouter une langue revient à créer un nouveau système.

Systran (utilisé entre autres par la Foreign Technology Division de l'armée de l'air américaine) est le système de transfert le plus connu du grand public (c'est le système auquel donne accès le moteur de recherche Altavista[2]). Il est basé principalement sur la consultation de grands dictionnaires bilingues à grande couverture et mis au point à grands renforts des ressources humaines.

La recherche en traduction automatique a été freinée vers 1966, suite à la sortie du rapport ALPAC de la National Science Foundation qui concluait à l'impossibilité d'une traduction automatique de qualité.

Cependant le Canada, en raison de sa politique bilingue a connu une activité faste en traduction automatique. METEO est un système de traduction automatique parmi les premiers systèmes de deuxième génération, il est entré en exploitation le 24 mai 1977 à Montréal. C'est un système très spécifique qui traduit toutes les prévisions météorologiques destinées aux grand public, émises par le service d'environnement atmosphérique du Canada. Une mise à jour de ce système est encore utilisée quotidiennement à cette tâche.

Le milieu des années 1980 a été le témoin de l'essor des mémoires de traduction. Une mémoire de traduction est une base de données contenant un grand nombre de traductions existantes. L'idée des mémoires est de fournir automatiquement à un traducteur des traductions déjà faites à des phrases présentes dans la base. Les mémoires sont de plus souvent capables de proposer des suggestions pour des phrases proches de celles disponibles dans la base. Il existe plusieurs systèmes commerciaux exploitant cette idée (TransSearch [3], EUROLANG[4]…) qui est très populaire chez les traducteurs professionnels.

Jusqu'à la fin des années quatre-vingt le cadre dominant a été l'approche basée sur les règles linguistiques, mais depuis 1990 ce cadre a été rompu par l'entrée en scène de méthodes et de stratégies nouvelles. Un projet important de IBM a donné naissance à un prototype de traduction CANDIDE[5] [Berger et. al., 1994] introduisant l'approche

[2] http://babelfish.altavista.com/
[3] http://www.tsrali.com/. TransSearch, système conçu au sein de Laboratoire RALI à l'université de Montréal.
[4] the European Languages Centre, http://www.eurolang.com
[5] http://www-2.cs.cmu.edu/afs/cs/user/aberger/www/html/candide.html

probabiliste au sein de la communauté linguistique. L'équipe d'IBM a publié les résultats de ses expériences avec un système de traduction purement statistique. La caractéristique principale de cette approche est d'analyser une grande quantité de textes parallèles qui sont les traductions l'un de l'autre (bi-textes) et d'inférer à partir de ces textes les paramètres d'un modèle probabiliste de manière automatique.

1.2. L'importance applicative.

L'importance applicative de la traduction automatique dans notre société est attestée par le développement rapide des technologies. Avec le développement d'Internet, du courrier électronique, des Intranets d'entreprise, des systèmes de gestion de documents, les utilisateurs ont besoin de plus en plus de comprendre immédiatement l'information dans différentes langues.

Par exemple sur Internet près de 60%[6] des sites Web sont en anglais, le reste étant partagé entre des sites espagnols, allemands, français, chinois, etc. L'accès à toute l'information ne peut se faire que si l'on connaît une multitude de langues. Dans ce cas, un logiciel de traduction prend toute sa valeur : c'est un outil qui nous permet non seulement de comprendre un texte écrit dans une langue que nous ne maîtrisons pas, mais surtout de le comprendre instantanément sans avoir à faire appel à une autre personne. Un logiciel de traduction nous donne donc une autonomie potentielle qu'aucune autre solution ne peut offrir.

La plupart des entreprises doivent diffuser l'information en plusieurs langues. Les collaborateurs doivent travailler sur l'information en plusieurs langues et donc traduire les documents avant de pouvoir les traiter ou les utiliser.

Les services de traduction sont de plus en plus débordés. En offrant aux traducteurs humains des outils pour accélérer leur processus de traduction, ils gagnent du temps sur la première phase de traduction et peuvent se concentrer sur l'exactitude et la précision des termes choisis.

Par exemple, pour traduire un contrat, le traducteur doit être juriste et connaître le droit des deux langues pour pouvoir localiser (et pas seulement traduire) des contrats, donc pour pouvoir les adapter en fonction du droit de chaque pays.

[6] http://www.itu.int/mlds/briefingpaper/wipo/french/annexeI-fr.html

1.3. Aperçu sur le contenu de mémoire :

Notre contribution dans ce projet est constituée de trois points principaux :

Premièrement, nous avons entraîné les trois premiers modèles d'alignement proposés par IBM tout en vérifiant l'efficacité de l'outil public GIZA.

Deuxièmement, nous avons concentré nos efforts sur la réalisation et la comparaison de décodeurs traductionnels. Plus précisément, nous avons développé et comparé deux décodeurs pour les modèles de traduction IBM2. Le premier décodeur implémente une technique de programmation dynamique tandis que le second est un algorithme vorace (greedy).

Enfin, nous avons comparé les performances des algorithmes de recherche que nous avons décrits en nous appuyant sur des méthodes d'évaluation de la traduction automatique**.**

Dans cette partie, nous allons présenter un bref aperçu sur le contenu des différents chapitres de notre mémoire.

Le mémoire contient sept chapitres :

Chapitre 1: Dans ce chapitre, nous parlons de l'état de l'art de la traduction automatique. Plusieurs générations de logiciels de la TA sont énumérés ainsi que l'approche probabiliste de la traduction qui fait l'objet de notre travail. D'autre part, nous présentons les intérêts scientifiques et pratiques de notre recherche.

Chapitre 2: L'approche proposée par [Brown et al, 93] fait le sujet de ce chapitre. En effet, les différents modules d'un engin de traduction probabiliste seront étudiés comme le modèle de langue, le modèle de traduction et le décodage.

Chapitre 3: Nous présentons le package « EGYPT » formé de plusieurs outils : Whittle est destiné à la préparation du corpus pour GIZA, l'outil GIZA dont nous nous servons pour entraîner les modèles de traduction (IBM 1,2 et 3) et enfin, CAIRO utilisé pour visualiser les résultats obtenus par le modèle 3.

Chapitre 4: Toutes les expérimentations ainsi que les résultats de l'entraînement des modèles de traduction obtenus à l'aide de GIZA sont montrés dans ce chapitre. Nous dressons aussi une comparaison entre les différents modèles 1, 2 et 3 ainsi que GIZA et un outil semblable RMTTK conçu au RALI.

Chapitre 5: Les efforts principaux de notre travail se sont concentrés sur le développement de deux décodeurs. En effet, nous expliquons dans cette partie et nous comparons deux techniques de décodage; la programmation dynamique (DP) et l'algorithme vorace (greedy).

Chapitre 6: Dans le chapitre 6, nous décrivons quelques méthodes d'évaluation de la traduction (WER, BLEU et l'évaluation humaine), puis nous comparons les performances de ces deux algorithmes de recherche.

Chapitre 7: Nous concluons et résumons notre travail par les points les plus intéressants dans cette recherche ainsi que les perspectives qui peuvent étendre notre projet dans le futur.

Chapitre 2

Traduction statistique

En 1949, Warren Weaver a suggéré une approche de la traduction automatique basée sur les données statistiques. Cependant, les capacités limitées (en calcul et en mémoire) des ordinateurs de l'époque expliquent en grande partie que cette approche n'ait pas été poursuivie. Sur l'effet de la révolution technique, la traduction statistique (Statistical Machine Translation SMT) est présentement une approche largement convoitée, bien qu'encore marginale dans le secteur industriel. Au début des années 90s, la traduction probabiliste a été ressuscitée par les chercheurs [Brown et al. 1993].

Dans ce chapitre, on explique le principe de la traduction statistique qui repose sur la métaphore du canal bruité. Nous décrivons les problèmes liés à la réalisation d'un système de traduction probabiliste, à savoir l'entraînement d'un modèle de transfert, d'un modèle de langue et le problème du décodage.

Nous décrivons sommairement les modèles de langue qui sont bien maîtrisés [Goodman et al., 2001] et à la fin de ce chapitre, nous étudions d'avantage les modèles de traduction probabiliste et en particulier les trois premiers modèles proposés par [Brown et al. 1993].

2.1. L'idée originale Shanon (canal bruité)

L'approche la plus courante de la traduction probabiliste s'inscrit dans le cadre des approches dites "canal bruité" (après les travaux de Shanon) qu'on explique ici de manière intuitive.

Deux personnes, un émetteur E et un récepteur R, souhaitent communiquer via un canal bruité. Ce canal est "tellement bruité" qu'une phrase S déposée par E à l'entrée du canal est reçue par R comme une autre phrase T, traduction de S.

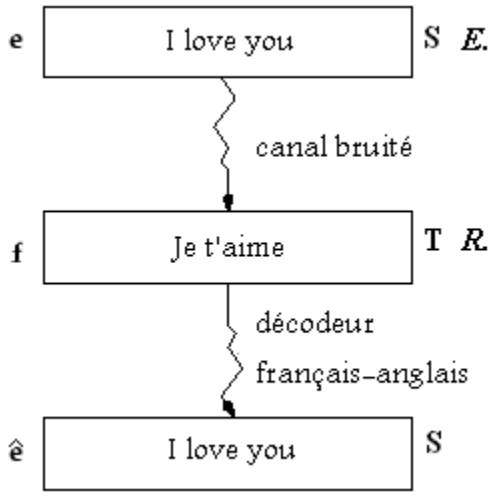

Figure 1: **Illustration du canal bruité. L'anglais est ici le langage source du canal et le français le langage cible.**

Le but pour R est de retrouver la phrase source à partir de la phrase reçue et ses connaissances du canal bruité. Chaque phrase de la langue source est une origine possible pour la phrase reçue T. On assigne une probabilité P(S|T) à chaque paire de phrases (S,T).

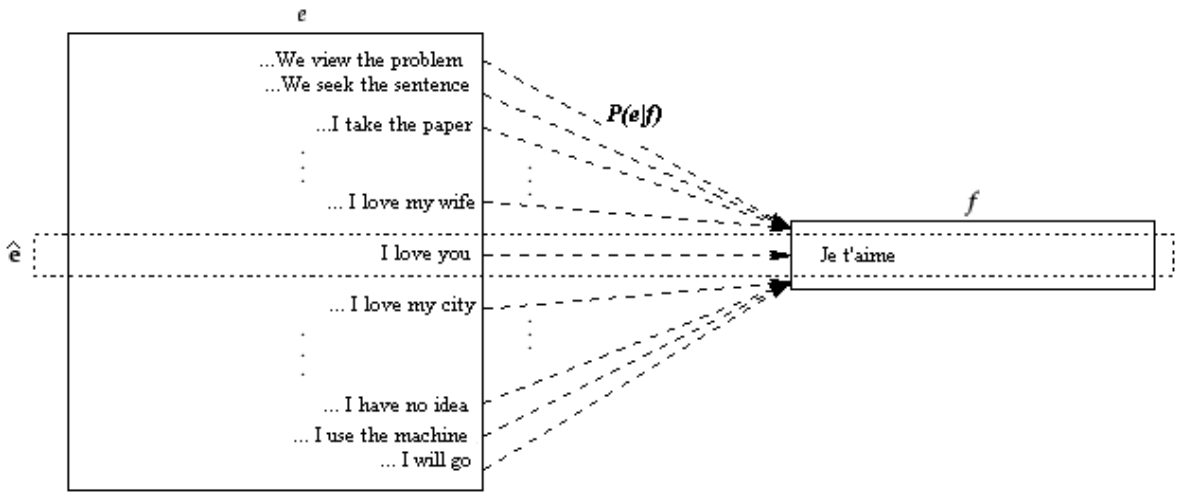

Figure 2: Chercher la phrase anglaise ê qui maximise la probabilité P(e|f).

On souhaite que le modèle du canal P(S|T) assigne une probabilité faible à des paires de phrases peu probables (exemple: We view the problem |Je t'aime) et une probabilité élevée à des paires plus probables (Je t'aime | I love you).

Dans ce travail, nous traduisons de la langue française vers la langue anglaise. Il est important de bien réaliser que dans le cadre de l'approche du canal bruité, la langue source est celle de l'émetteur et la langue cible est celle de récepteur. Lorsqu'on considère le décodeur, c'est le contraire. Cette distinction est souvent à l'origine de confusion dans la littérature, aussi prenons-nous la convention suivante dans ce mémoire.

Convention: *Nous prenons pour langue source (S) l'anglais et pour cible le français (T). On désigne par source l'entrée du canal bruité et cible la sortie du canal bruité ceci évite l'ambiguïté des termes source/cible dans l'approche du canal. (Voir figure 1).*

Le problème général de la traduction statistique est de trouver la phrase $\hat{e}$, étant donnée une phrase f^J, qui maximise $P(e^I|f^J)$. De manière plus formelle :

$$\hat{e} = \underset{e}{argmax}\, P(e^I \mid f^J)\,. \qquad \textbf{(2.1)}$$

D'après le théorème de Bayes:

$$P(e^I \mid f^J) = \frac{P(f^J \mid e^I) \times P(e^I)}{P(f^J)} \qquad \textbf{(2.2)}$$

Comme le dénominateur de l'équation (2.2) est indépendant de e^I, l'opération de maximisation devient alors:

$$\hat{e} = \underset{e}{Argmax}\, P(e^I \mid f^J) = \underset{e}{Argmax}\, P(e^I) \times P(f^J \mid e^I). \qquad \textbf{(2.3)}$$

On appelle $P(e^I)$, *un modèle de langue source*, tandis que le deuxième facteur $P(f^J|e^I)$ est appelé *un modèle de traduction*.

Alors, le problème de la traduction se divise en trois sous problèmes que nous décrivons par la suite :

1- Calculer les paramètres du modèle de langue;
2- Calculer les paramètres du modèle de traduction;
3- Réaliser un décodeur, c'est-à-dire un mécanisme capable d'effectuer l'opération de maximisation de l'équation (2.3) en un temps acceptable.

2.2. Le modèle de langue

Un modèle de langue est un modèle qui spécifie une distribution *P(e)* sur les chaînes e^i de la langue modélisée:

$$\sum_i P(e^i) = 1$$

Sans perte d'information, si l'on considère que e^I est une suite de *I* mots (une phrase de *I* mots), $e^I = w_1 \ldots w_I$, alors:

$$P(e^I) = \prod_{i=1}^{I} P(w_i \mid \underbrace{w_1 \ldots w_{i-1}}_{h}) \qquad \textbf{(2.4)}$$

Où h est appelée l'historique.

Un modèle de langue probabiliste peut être présenté comme une fonction donnant la probabilité d'observer un mot étant donné ceux déjà observés. Cette approche a déjà montré son utilité dans plusieurs applications dont la reconnaissance de la parole et de caractères ou encore la correction de fautes d'orthographe.

L'estimation des distributions P(w|h) où w est un mot et h l'historique (l'ensemble des mots déjà vus) est un problème complexe (trop de paramètres à estimer) que l'on simplifie habituellement de la manière suivante :

$$P(e^I) \approx P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\, P(w_4|w_2\, w_3) \ldots P(w_I|w_{I-2}w_{I-1})$$

La probabilité d'un mot est conditionnée "seulement" par les deux derniers mots dans l'historique de w. Cette simplification est appelée un modèle trigramme.

Nous utilisons dans notre travail un modèle de langue développé par le RALI sur un corpus de plus d'un million de phrases en anglais extraites du HANSARD. Pour plus d'information on peut consulter le travail de [Goodman et al, 2001].

Le modèle de langue est nécessaire pour que le modèle de traduction puisse concentrer ses masses de probabilités sur des paires de phrases à peu près raisonnables. Soit f^J une phrase connue donc bien formée, e^I l'est à peu près, grâce au modèle de langue.

2.3. Les modèles de traduction

Nous nous intéressons ici au problème de calcul de $P(f^J|e^I)$, la probabilité d'une phrase f^J étant donnée une phrase anglaise e^I. On appelle la méthode qui permet de calculer cette distribution « un modèle de traduction ».

Les cinq différents modèles de traduction d'IBM sont décrits dans le papier de [Brown et al 1993] mais c'est un papier compliqué à lire. Le lecteur pourrait cependant commencer par lire [Knight, 1999] qui propose un tutoriel très accessible à ces modèles.

2.3.1. Les alignements

Dans [Brown et al. 1993], un modèle de traduction est vu comme un modèle d'alignement de mots. On introduit l'idée d'alignement entre une paire de phrases (e^I,f^J) de façon que chaque mot de la phrase française soit associé au mot anglais qui le génère.

Les figures[7] 3, 4, 5 et 6 nous montrent plusieurs alignements qui sont tous acceptables avec des probabilités différentes (l'alignement de la figure 4 est moins probables que celui de la figure 3).

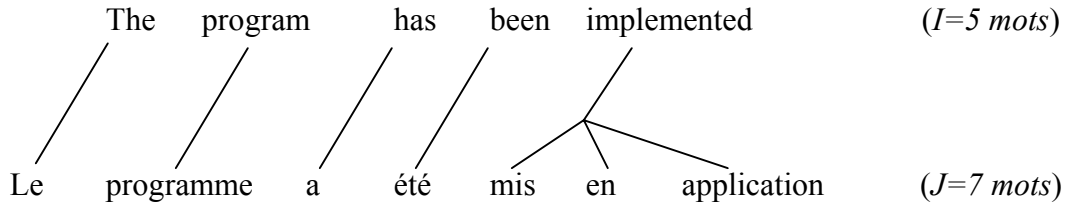

Figure 3: Un alignement dont chaque mot français est aligné à un seul mot anglais.

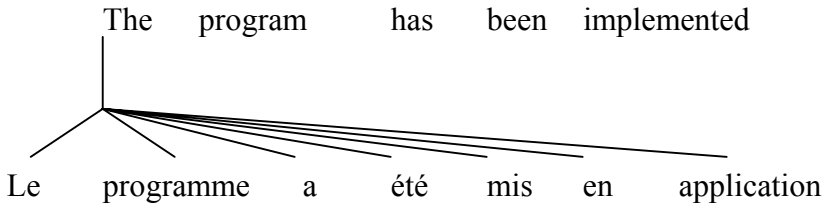

Figure 4: Un autre alignement possible de ces phrases mais moins probable.

[7]Exemples pris de [Brown et al., 1993].

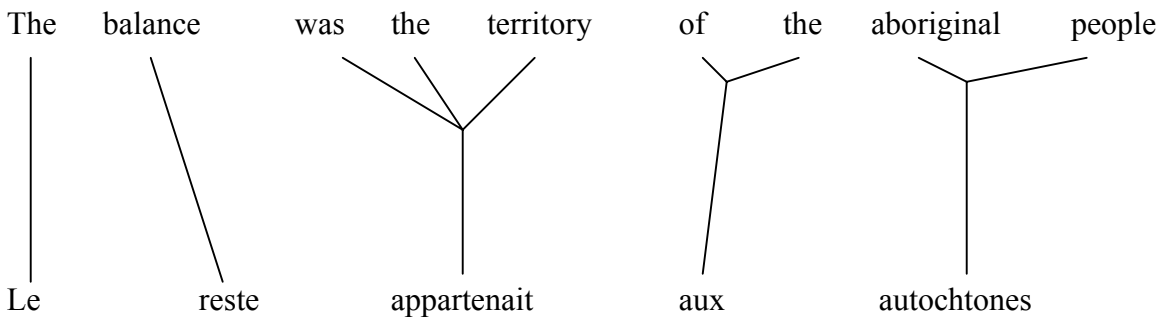

Figure 5: Un alignement dont chaque mot anglais est associé à un seul mot français.

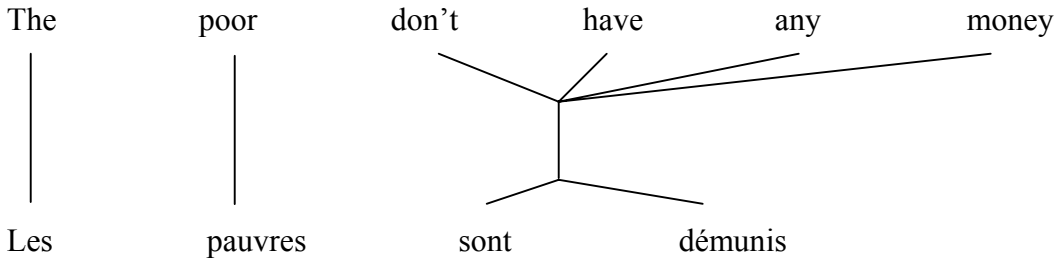

Figure 6: Un alignement dont un ensemble de mots français est connecté à un ensemble de mots anglais.

Nous montrons en figure 3 et 5 et 6 trois types d'alignements. Dans celui de la figure 3, chaque mot français est connecté à un et un seul mot anglais; Cependant un mot anglais peut-être associé à un ou plusieurs mots français. La figure 5 montre un alignement où un mot anglais n'est connecté qu'à un seul mot français. Enfin, la figure 6 présente un alignement général où plusieurs mots français peuvent se connecter à plusieurs mots anglais. Dans cet exemple, les quatre derniers mots anglais ensemble *(don't have any money)* sont alignés aux deux mots français (*sont démunis)*.

Les modèles que nous utilisons ne connaissent que les alignements pour lesquels un mot français est aligné à au plus un mot anglais. Un alignement peut donc être représenté comme un vecteur affectant, à chacune des positions françaises, une position anglaise. Par exemple, (*Le programme a été mis en application | the (1) program (2) has (3) been (4) implemented (5,6,7))* représente l'alignement de la figure 3.

L'ensemble des mots français associés à un mot anglais est dénommé par "*cept*", et si un mot anglais n'est connecté à aucun mot français on dit que le "*cept*" est vide. Ce cept vide qui n'a pas de position, est connecté par convention à la position 0 et on le note e_0.

Exemple: *(J'applaudis à la décision| e_0(3) I(1) applaud(2) the(4) decision(5*)), on voit que *"à"* est connecté au mot NULL qui est par convention à la position zéro de la phrase anglaise.

On note l'ensemble des alignements considérés par les modèles de traduction d'IBM entre les deux phrases f^J et e^I par $A(e,f)$.

2.3.2. Idée initiale

Pour mieux comprendre les modèles de traduction, prenons une paire de phrases anglaise et française telles qu'elles aient le même nombre de mots et que le mot français en position i soit aligné au mot anglais à cette même position (voir la figure 7) : on assume que les phrases sont reliées mot à mot (la relation (e^I,f^J) est une bijection).

Étant donnée la phrase anglaise e^I de I mots (e_1,e_2,…e_I) et sa traduction française f^J (f_1,f_2,...f_J), la probabilité $P(f^J|e^I)$ s'exprime par:

$$P(f^I \mid e^I) = \prod_i^I t(f_i \mid e_i) \qquad \textbf{(2.5)}$$

Exemple: (Jean aime Marie| John loves Mary), il est raisonnable d'assumer que John produit Jean, loves produit aime, et Mary produit Marie.

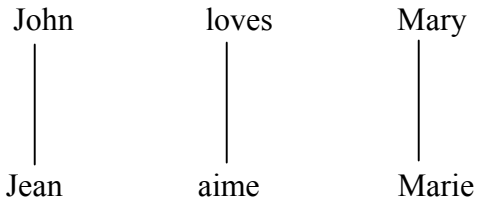

Figure 7: Un exemple simple d'alignement, chaque mot est aligné avec sa traduction dans la même position.

En pratique, une telle relation entre les phrases françaises et anglaises n'est bien sûr pas observée. Les phrases françaises traductions de phrases anglaises sont habituellement plus longues (par exemple la moyenne de longueur de la traduction française d'une phrase anglaise de 20 mots est de 21.3) .

2.3.3. Les modèles de traduction proposés par IBM

[Brown et al, 1993] propose cinq modèles de traduction 1, 2, 3, 4 et 5. Chaque modèle a sa propre prescription pour calculer la probabilité conditionnelle *P(f|e)*, qu'on appelle la probabilité de traduction. Toute paire de mots reliés (e_i,f_j) du corpus d'entraînement est un paramètre du modèle. On fait un choix initial de l'ensemble des paramètres du modèle, puis on applique un algorithme d'estimation afin d'améliorer la vraisemblance du corpus d'entraînement.

Notation: *On cherche à modéliser* $P(F=f^J|E=e^I)$ *où E est l'ensemble de phrases anglaises, F est l'ensemble de phrases françaises.* $e^I=e_1,...,e_I$ *et* $f^J=f_1,...,f_J$ *sont deux phrases particulières de E et F.* $A(e^I,f^J)$ *est l'ensemble des alignements liant une phrase anglaise donnée à une phrase française. On note par* $P(F=f^J,A=a|E=e^I)$ *la probabilité de jointe de* f^J *et d'un alignement particulier a.*

Alors :
$$P(f^J \mid e^I) = \sum_a P(f^J, a \mid e^I). \qquad \textbf{(2.6)}$$

Durant ce travail, nous avons expérimenté les trois premiers modèles de IBM que nous décrivons maintenant brièvement.

2.3.3.1. Modèle de traduction probabiliste IBM1

Pour modéliser $P(F=f^J,A=a|E=e^I)$, il y a trois paramètres à estimer:

1. Choisir la longueur *J* (nombre de mots) de la phrase française f^J que l'on cherche à générer (selon le modèle $P(J|e^I)$). La phrase française est habituellement plus longue que la phrase anglaise.
2. Pour chaque mot français, soit f_j le mot considéré, choisir une position (entre *0* et *J*) dans e^I associée à f_j selon la distribution a. e_{aj} est le mot qui est responsable de la génération du j^{ieme} mot de f^J. Dans le modèle 1, tous les alignements sont équiprobables et indépendants de la position du mot dans la phrase française. Chaque mot français f_j possède donc *I+1* positions possibles (+1 car le mot NULL e_0 est considéré).
3. Choisir en utilisant les distributions de transfert un mot français f_j sachant cette position déterminée.

Donc, $$P(f^J, a \mid e^I) = P(J \mid e^I) \prod_{j=1}^{J} \frac{t(f_j \mid e_{aj})}{(I+1)}. \quad \textbf{(2.7)}$$

Dans le cas du modèle 1, [Brown et al., 1993] ont démontré que $P(f^J|e^I)$ peut-être calculé de manière exacte par la formule suivante[8]:

$$P(f^J \mid e^I) = \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} t(f_j \mid e_i). \quad \textbf{(2.8)}$$

Où ε désigne la probabilité $P(J|e^I)$.

2.3.3.2. Modèle de traduction probabiliste IBM2

Dans IBM1, la probabilité d'alignement d'un mot anglais en position *i* avec un mot français en position *j* est indépendante des positions *i* et *j*. Toutes les positions sont possibles et équiprobables. Cependant l'expérience sur un corpus bilingue (*français/ anglais telles que les phrases sources et cibles soient de 8 mots*) montre que dans 70% de cas (Voir tableau 1), les mots correspondants ont les mêmes positions, dans 10% des cas les mots diffèrent d'une position et dans 5% ils diffèrent de deux positions.

Source / Cible	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.85	0.1	0.5					
2	0.15	0.7	0.1	0.05				
3	0.05	0.1	07	0.1	0.05			
4		0.5	0.1	0.7	0.1	0.05		
5			0.05	0.1	0.7	0.1	0.05	
6				0.05	0.1	0.7	0.1	0.05
7					0.05	01	07	0.15
8						0.05	0.1	0.85

Tableau 1: Les probabilités des alignements sur un corpus de phrases de 8 mots (anglais /français).

[8] Voir [Brown et al, 1993] pour les détails.

IBM2 remédie à cette simplification à outrance en introduisant des probabilités d'alignement ***a(i|j,J,I)*** **:** la probabilité qu'un mot anglais en position *i* soit associé à un mot français en position *j*, sachant les longueurs respectives (*I* et *J* comptées en mot) des deux phrases considérées. On élabore alors la probabilité d'alignement $a(a_j|j,J,I)$ comme un nouveau paramètre pour le modèle 2 sous la contrainte stochastique.

$$\sum_{i=0}^{I} a(i \mid j, J, I) = 1 \qquad \textbf{(2.9)}$$

$P(e^I|f^J)$ devient alors dans le cas du modèle 2 :

$$P(f^J \mid e^I) = \varepsilon \prod_{j=1}^{J} \sum_{i=0}^{I} t(f_j \mid e_i) a(i \mid j, J, I). \qquad \textbf{(2.10)}$$

On remarque que le modèle 1 est un cas particulier du modèle 2 en fixant $a(i|j,J,I)$ à $(I+1)^{-1}$. Ceci permet à [Brown et al, 1993] de proposer d'entraîner un modèle 1 (dont l'entraînement converge vers un optimum global) pour initialiser l'entraînement d'un modèle 2, pour lequel la convergence n'est que locale.

2.3.3.3. Modèle de traduction probabiliste IBM3.

Le modèle 3 introduit la notion de fertilité que nous utilisons indirectement dans nos décodeurs. Nous introduisons donc cette notion et laissons le soin au lecteur de lire [Brown et al, 1993] pour plus de détails. Le modèle 2 intègre indirectement cette notion de fertilité: Le nombre de mots français connectés à un mot anglais particulier donne un indicateur de sa fertilité. Cependant, le modèle 3 intègre la fertilité comme un paramètre. Introduisons une nouvelle variable aléatoire appelée fertilité (dénoté ϕ_e) qui représente la distribution du nombre de mots cible français générés par un mot source anglais particulier.

Par exemple :
(the) est souvent traduit par un mot (le, la, l'), mais peut également ne pas être traduit. (fertilité 1 ou bien 0)
(only) est très souvent traduit par un seul mot (seulement), mais de temps en temps par deux mots (ne . . . que). (fertilité 1 ou peut-être 2)
(update) est presque systématiquement traduit par 3 mots (mise à jour). (fertilité 3)

Dans le modèle 3, un mot anglais peut se connecter à aucun ou plusieurs mots français.

Les principaux paramètres du modèle 3 sont définis par:

- Les probabilités de fertilité : $n(\phi \mid e_i)$ est la probabilité que le mot e_i a une fertilité ϕ.

- Les probabilités de transfert : $t(f_j|e_i)$ est la probabilité que le mot e_i génère le mot français f_j.

- Les probabilités de distorsion : $d(j \mid i, I, J)$ est la probabilité que le mot français de la position j soit généré par le mot anglais de la position i (J et I les longueurs de phrases).

À noter que les probabilités de distorsion sont inversées par rapport à celles d'alignement du modèle 2.

[Brown et al., 1993] proposent de convertir les paramètres obtenus à la suite de l'entraînement du modèle 2 pour initialiser les paramètres du modèle 3.

2.4. Conclusion

Dans cette partie, nous avons présenté les différents problèmes de la traduction statistique ainsi que les trois premiers modèles proposés dans [Brown et al, 1993]. Pour la conception d'un décodeur, nous avons utilisé un trigramme développé au RALI et qui a déjà montré ses performances. D'autre part dans les prochains chapitres, nous mettrons à l'épreuve un outil du package Egypt baptisé Giza qui permet l'entraînement des modèles IBM. Ainsi donc, dans les chapitres trois et quatre, nous mettrons en exergue le fonctionnement et les différents modules du package Egypt. Le problème de décodage fera l'objet du chapitre 5.

Chapitre 3

L'estimation des paramètres

Nous avons abordé dans le chapitre précédent les modèles IBM d'un point de vue formel. Nous décrivons ici l'outil EGYPT qui permet d'entraîner les modèles IBM. Ce package mis au point par un groupe de travail de l'université Johns Hopkins [Al-Onaizan et al., 1999] est disponible aux chercheurs de tout organisme universitaire.

Nous expliquons les différents outils du package EGYPT utilisés pour estimer les paramètres des modèles IBM. Nous présentons les types de données (entrées et sorties) de chaque outil ainsi qu'une brève explication sur les techniques suivies pour la préparation des corpus bilingues nécessaires à l'entraînement.

3.1. Le projet EGYPT

Le package de traduction probabiliste "*EGYPT*", a été construit lors d'un "*MT workshop*" de l'université Johns Hopkins [Al-Onaizan et al., 1999][9]. L'équipe "EGYPT" avait les objectifs suivants à réaliser:

1. Construire un toolkit de traduction automatique statistique et le rendre disponible aux chercheurs de la communauté langagière.
2. Construire à l'aide de ce toolkit un système de traduction automatique tchèque-anglais.

EGYPT est conçu de plusieurs modules, nous en décrivons les principaux:

1. Whittle qui permet de préparer les données d'entraînement et de test.
2. GIZA qui réalise l'entraînement des modèles IBM 1, 2 et 3.
3. CAIRO qui permet de visualiser des alignements de mots.

[9] On peut télécharger cet outil de http://www.clsp.jhu.edu/ws99/projetcs/mt/toolkit/

3.1.1. Bitexte

Le point de départ de l'entraînement est ce que l'on désigne par bitexte. Un bitexte est un corpus bilingue parallèle (un texte dans une langue de départ et sa traduction) où les liens de traduction entre les phrases ou groupes de phrases sont explicites.

On peut obtenir un bitexte à partir d'un corpus bilingue en alignant le corpus au niveau des phrases[10]. Pour y arriver, deux types d'information sont exploités dans les algorithmes d'alignement :

-Les informations métriques : [Church et William Gale 1991a] utilisent la longueur des phrases (comptée en nombre de caractères ou mots) comme critère de mise en correspondance. Les auteurs ont en effet montré qu'il existe un rapport de proportionnalité entre la longueur d'une phrase en langue de départ et la longueur de sa traduction.

- Les informations à caractère linguistique : [Simard et al., 1992] proposent d'aligner des corpus bilingues en exploitant le fait que deux phrases en relation de traduction partagent souvent des mots communs ou proches « les cognates » : comme des données chiffrées, des noms propres, ou encore des mots partageant la même racine. (Exemple : accès/access, activité/activity, parlement/parliament…)

Nous avons utilisé dans notre mémoire un aligneur développé au RALI [Simard et al., 1992].

3.1.2. Whittle

Whittle permet la préparation d'un corpus bilingue au format requis par GIZA à partir du bitexte, Whittle calcule les fréquences de chaque mot puis associe un indice à chacun. Whittle produit alors un texte formé d'une suite d'indexes.

Cet outil permet entre autre de spécifier laquelle des deux langues sera la langue source (ici anglais), et gère également des options qui peuvent influer sur la qualité des modèles produits, comme la longueur maximale des phrases que l'on veut conserver à

[10] Voir [Langlais et al, 1998] pour une comparaison de différents algorithmes d'alignement de phrases.

l'entraînement, ou encore la fréquence minimale d'un mot en dessous de laquelle un mot sera associé à une forme inconnue (UNK). Whittle permet enfin d'extraire du bitexte initial une petite collection de phrases pour le test.

Le format d'entrée de Whittle est un bitexte où les phrases sont reliées une à une. (*figure 8*)

1. tabling of documents
2. house of commons
3. thursday , april 17 , 1986

1. pétitions
2. chambre des communes
3. le jeudi 17 avril 1986

Figure 8: Exemple d'un corpus bitexte d'entraînement.

Le format des sorties de Whittle et les entrées de GIZA sont des textes formés de nombres qui représentent les indexes des mots et les fréquences de chacune des phrases. Un exemple de 2 paires de phrases anglaise – française:

1 *(fréquence de paire)*
1 1 226 5008 621 6492 226 6377 6813 226 9505 5100 6824 226 5100 5222 *(anglais)*
2769 155 7989 585 1 578 6503 585 8242 578 8142 8541 578 12328 *(français)*
1*(fréquence de paire)*
1 1 226 6260 11856 11806 1293 *(anglais)*
11 1 1 11 155 14888 2649 11447 9457 8488 4168 *(français)*

Figure 9: Illustration du format de corpus généré par Whittle. Les textes entre parenthèses ne font partie du format. Cet extrait contient deux paires de phrases.

3.1.3. GIZA

Cette section décrit la partie d'Egypt qui extrait l'information d'un corpus bilingue. Ce module appelé GIZA est basé sur les algorithmes décrits dans [Brown et al., 1993a].

Types de données d'entrées

Les paramètres de transfert d'un modèle 1 sont représentés par une grande matrice creuse à deux dimensions et initialisés uniformément en donnant une probabilité de transfert à chaque paire de mots croisés au moins une fois dans une paire de phrases. (voir le tableau 2, pour plus de détails sur le nombre de paramètres de transfert). Les probabilités d'alignement du modèle 2 peuvent être initialisées uniformément ($1/I+1$).

Pour les probabilités de transfert, deux possibilités (au moins):
- Les initialiser uniformément (tout comme pour le modèle 1).
- Utiliser les valeurs obtenues après entraînement d'un modèle 1.

La deuxième solution est préférable car l'entraînement du modèle 2 ne converge que vers un optimum local. De meilleures estimées de départ, offrent donc plus de chance de converger vers une solution raisonnable. Les deux options sont offertes par GIZA et on a choisi l'entraînement du modèle 2 initialisé par les probabilités du modèle 1. Nous avons expérimenté les deux stratégies sur un corpus de 45 000 paires de phrases, la perplexité du modèle 2 initialisé par les paramètres du modèle 1 est de 30.59 cependant sans l'initialiser par les paramètres du modèle 1, elle est de 37.38. (nous expliquons la mesure de la qualité du modèle de traduction par la perplexité dans le chapitre 4, section 4.5).

De même, on peut entraîner le modèle IBM3 directement à un corpus, mais il est préférable d'initialiser les paramètres de IBM3 à partir des paramètres de IBM2. Cette étape n'est cependant pas aussi triviale que lors du passage de IBM1 à IBM2. Si les probabilités de transfert se récupèrent sans modification, les probabilités d'alignement de IBM2 doivent être inversées car elles sont l'inverse de celles du modèle 3 ($a(i|j,J,I)$ dans modèle 2 et $d(j|i,J,I)$ dans modèle 3). Le plus coûteux consiste à collecter les comptes pour initialiser les fertilités. [Brown 1993] proposent un passage du modèle 2 au modèle 3.

L'idée est de collecter les comptes normalisés par la probabilité de chaque alignement. Voir [Brown et al., 1993] pour un algorithme efficace.

Les sorties

Les résultats sont stockés sous forme de matrice creuse binaire :

A- La table de transfert (T-table)

Où les paramètres sont exprimés selon le format :

Source_id cible_id P(cible_id/source_id).

B-La table des fertilités N table.

Il est possible d'utiliser des fertilités uniformes lors de l'entraînement du modèle 3 à partir du modèle 2, mais [Brown et al, 1993a] suggèrent une initialisation de fertilités qui prend le modèle 2 en compte.

Où les paramètres sont exprimés selon le format :

Mot_id $n(0|e_i)$ $n(1|e_i)$ ……. $n(10|e_i)$.

C- Les tables d'alignement.

Les deux derniers modèles de traduction 2 et 3 possèdent des tables d'alignement.

-Dans le cas d'un modèle 2 : A-tables est exprimé selon ce format : *i j I J P(i | j, J,I)*.

-Dans le cas d'un modèle 3 : D-tables selon le format suivant : *i j I J P(j | i, J,I)*.

Où,

i = La position dans la phrase source.

j = La position dans la phrase cible.

I = La longueur de la phrase source.

J = La longueur de la phrase cible.

Et P(i | j, J, I) est la probabilité que le mot anglais de la position *i* sera à la position *j* dans une paire de phrases de longueur *I* et *J*.

3.1.4. Cairo

Cairo est un outil de visualisation développé pour les modèles de traduction statistiques d'IBM.. Il permet de visualiser l'alignement proposé par IBM3 d'une paire de

phrases ainsi que les probabilités impliquées. La figure 10 illustre une session avec cet outil.

Dans une interface utilisateur graphique (GUI), Cairo montre la paire de phrases en entrée (supposées l'une une traduction de l'autre) avec des lignes dessinées entre les mots alignés. Cette représentation peut être montrée verticalement ou horizontalement. La figure 10 montre l'interface de cairo sur une paire de phrases. Toujours sur l'interface, on peut également voir les probabilités de fertilités et les autres paramètres etc…

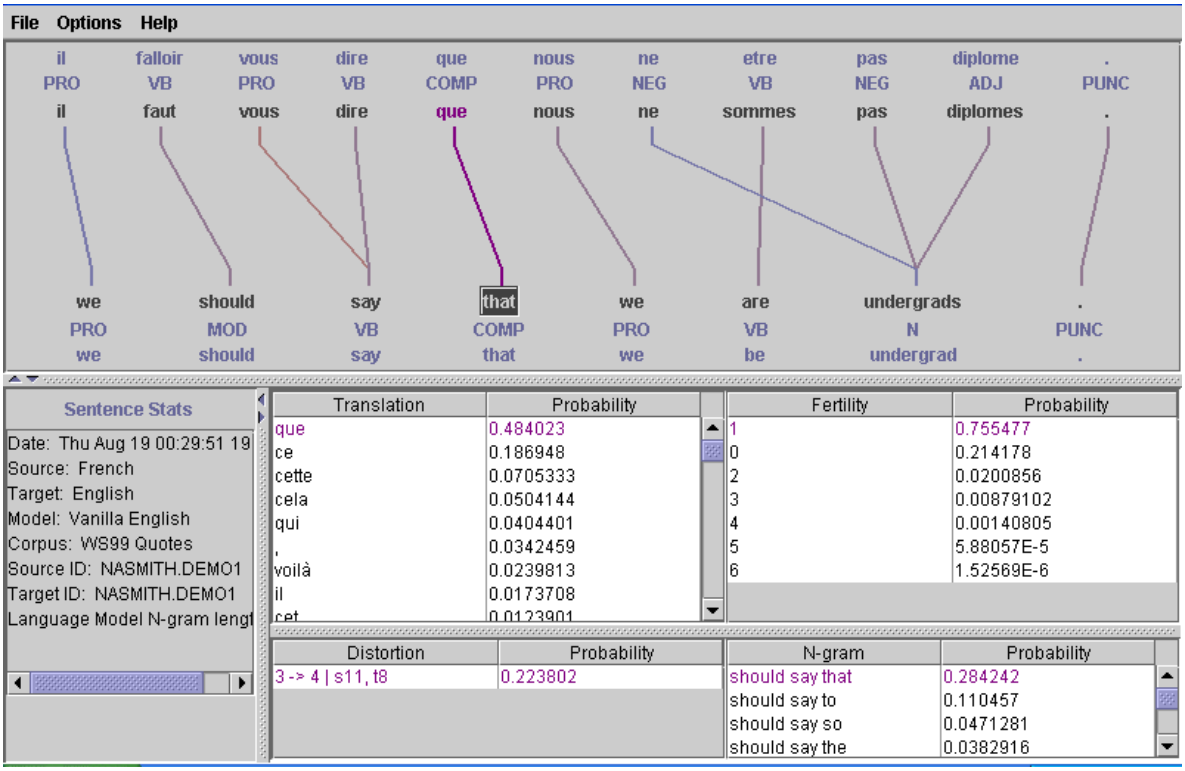

Figure 10: Un exemple de cairo.

3.2. Conclusion

Nous avons présenté les outils du package Egypt qui permet l'entraînement des modèles IBM. Après la présentation de ce package Egypt, nous voulons mettre en preuve GIZA qui implémente les modèles de traduction IBM. Dans ce contexte, nous avons utilisé un outil d'alignement de phrases conçu au RALI pour préparer le bitexte nécessaire pour l'entraînement. Nous présentons nos expériences sur GIZA dans le chapitre suivant tout en comparant cet outil avec un outil d'entraînement (RMMTK) développé au RALI.

Chapitre 4

Les expériences avec GIZA

Le corpus des débats parlementaires canadiens (connu sous le nom de Hansard) a été utilisé dans notre étude pour entraîner les paramètres des modèles de traduction. Ce corpus est constitué de 1 639 250 paires de phrases, 29 547 933 mots anglais et 31 826 112 mots français. Les tailles des vocabulaires anglais et français sont respectivement de 103 830 et de 83 106 mots différents.

Dans ce chapitre, nous décrivons les expériences réalisées avec GIZA (décrit dans le chapitre précédent) et comparons la qualité des trois modèles que nous avons entraînés à l'aide de ce package. Nous nous intéressons également aux temps d'entraînement et à l'espace mémoire requis par ce package. Ce type de considération à priori secondaire, prend toute son importance lorsqu'on sait que les tables utilisées peuvent facilement dépasser l'espace mémoire disponible sur une machine performante et rendre l'apprentissage inopérationnel.

Nous comparons enfin l'outil GIZA avec l'outil RMTTK (RALI Machine Translation Tool kit) développé au RALI en terme de performance, de temps de calculs et d'occupation mémoire.

4.1. Préparation du corpus

Afin de rendre l'entraînement efficace, nous avons effectué quelques prétraitements sur notre bitexte d'entraînement.

On élimine tous les mots anglais ou français apparus une seule fois en les remplaçant par le mot UNK, ceci permet :

1- De limiter le bruit dans les modèles car on entraîne les modèles seulement sur les mots apparus plus d'une fois (une estimation faite à partir d'une seule observation est généralement trop bruitée).

2- De gérer de manière passive la gestion des mots inconnus du modèle grâce à l'obtention du problème $P(f_j|UNK)$ et $P(UNK|e_i)$. Des mots inconnus interviennent entre autre lors de faute d'orthographe ou de frappe.
3- De réduire considérablement la taille des modèles résultants. Un corpus de 10 000 paires de phrases contient par exemple 1.7% de mots anglais apparaissant une seule fois et 1.9% de mots français. La réduction du nombre de paramètres observées lorsqu'on fait ce mapping est de 16.9% pour le modèle 1, de 15.6% pour le modèle 2, de 13.7% pour le modèle 3.

De plus, pour réduire les temps de calculs, on ne conserve du corpus que les phrases ne dépassant pas une longueur donnée (40 mots dans nos expériences). Enfin, un corpus de 1 451 924 paires différentes de phrases anglaises et françaises a été retenu dans nos expériences pour un vocabulaire bilingue de 65 903 mots français et 51 558 mots anglais.

4.2. Les paramètres.

On démarre l'entraînement du modèle avec une valeur initiale des paramètres qui n'est pas critique (où la somme des probabilités de transfert est égale à 1) car le modèle IBM1 converge vers un optimum global.

Nombre de paires de phrases	**Nombre de mots français différents**	**Nombre de mots anglais différents**	**Nombre de paramètres *t(f|e)* Modèle 1**	**Nombre de paramètres *t(f|e)* Modèle 2**	**Nombre de paramètres *t(f|e)* Modèle 3**
46 237	14 516	11 311	1 233 241	419 064	189 738
88 966	20 340	15 631	1 824 878	685 372	316 921
175 754	27 611	21 161	2 603 367	1 074 824	516 824
352 983	37 466	28 704	3 919 128	1 733 551	853 678
1 451 924	65 903	51 559	8 846 847	4 178 600	2 258 485

Tableau 2: Les nombres de paires de phrases et les paramètres de transfert de chaque modèle de traduction.

Après chaque itération, on élimine toutes les probabilités inférieures à un seuil donné (10^{-8}). On obtient après sept itérations de 8 846 847 paramètres de transfert pour le modèle 1, 4 178 600 pour le modèle 2 (53 % des paramètres du modèle 1), et 2 258 485 (54% des paramètres du modèle 2) pour le modèle 3. En moyenne, chaque mot anglais est associé à 171 mots avec le modèle 1, 82 avec le modèle 2 et 44 mots avec le modèle 3. La réduction de nombre de paramètres d'un modèle à l'autre est expliquée par le fait que le modèle 2 concentre ses probabilités aux mots les plus pertinents beaucoup plus que le modèle 1 et moins fort que le modèle 3.

Les paramètres d'alignement et de distorsion sont stockés dans une matrice à cinq dimensions. Le nombre de paramètres d'alignements pour le modèle 2 sur un corpus de 352 983 paires de phrases d'au plus 40 mots, est de 613 601 paramètres et les paramètres de distorsion du modèle 3 sont de 461 075 paramètres. Cependant sur un corpus de 1 451 924 paires de phrases, le nombre de paramètres d'alignements est de 618 636 pour le modèle 2 et de 511 517 pour le modèle 3.

Les fertilités sont stockées dans une matrice à huit dimensions parce qu'on a proposé un maximum de 7 pour la fertilité. Seul le modèle 3 a la fertilité comme paramètre. Le nombre de fertilité est égale à la taille de vocabulaires anglais parce que le modèle 3 accorde sept fertilités à chaque mot anglais (voir tableau 5).

4.3. L'espace mémoire

Les tables de probabilités de transfert et d'alignement occupent un espace important de la mémoire de la machine surtout lors de l'exécution du modèle 3 (3 tables associées au modèle 3 ; transfert, fertilités, distorsion ou alignement). Seuls les structures adéquates (table de hachage) permettent de restreindre cet espace (matrice creuse) et rendent l'entraînement de ces modèles possible.

Prenons les tables d'alignement comme un exemple pour voir la complexité de la représentation des tables de probabilités. On a besoin d'une structure supplémentaire pour stocker $a(i|j,J,I)$ (modèle 2 et 3). C'est une table coûteuse à coder, une table à 4 dimensions

occuperait $(L \times M)^2$ cases (float de 4 octets) en mémoire, où L et M sont respectivement les longueurs source et cible (comptées en mots) maximales des phrases du bitexte. Par exemple, dans notre cas, nous avons retenu les phrases d'au plus 40 mots (L=M=40 mots), une taille de 10 240 000 octets (soit 10 MO.)

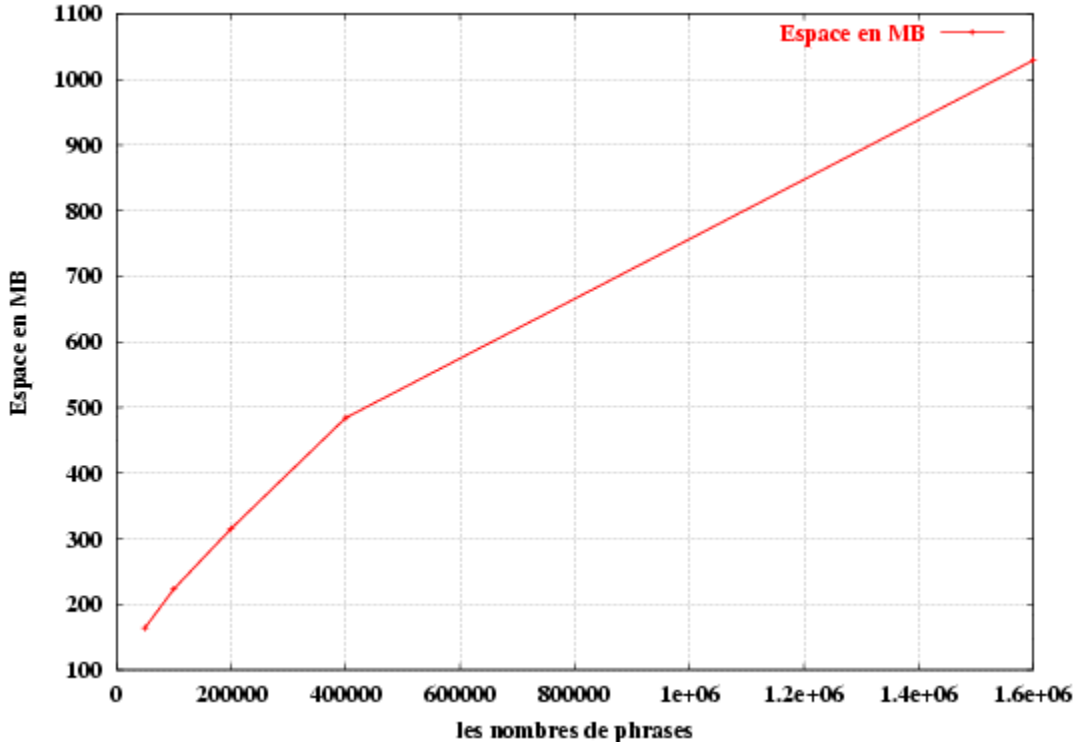

Figure 11: L'espace mémoire occupé par l'entraînement de GIZA.

La figure 11 nous montre la variation de l'espace mémoire occupé par GIZA. On a retenu l'espace mémoire maximal occupé durant chaque exécution.

Par exemple, on a remarqué que pour un corpus d'environ 50 000 paires de phrases, un espace de 164 MO est pris au maximum par GIZA. Pour un corpus de 400 000 paires, un espace de 484 MO a été réservé et la totalité de l'espace mémoire disponible (1024 MO) a été occupé lors de l'entraînement sur le bitexte au complet, ceci signifie qu'il nous est impossible avec GIZA d'entraîner de plus gros modèles sur les machines disponibles au département.

4.4. Temps d'exécution

L'espace mémoire a une grande influence sur le temps d'exécution, une machine ayant une capacité mémoire assez grande accélère l'exécution (on évite le swapping qui ralentit considérablement, en fait, de manière non raisonnable l'entraînement).

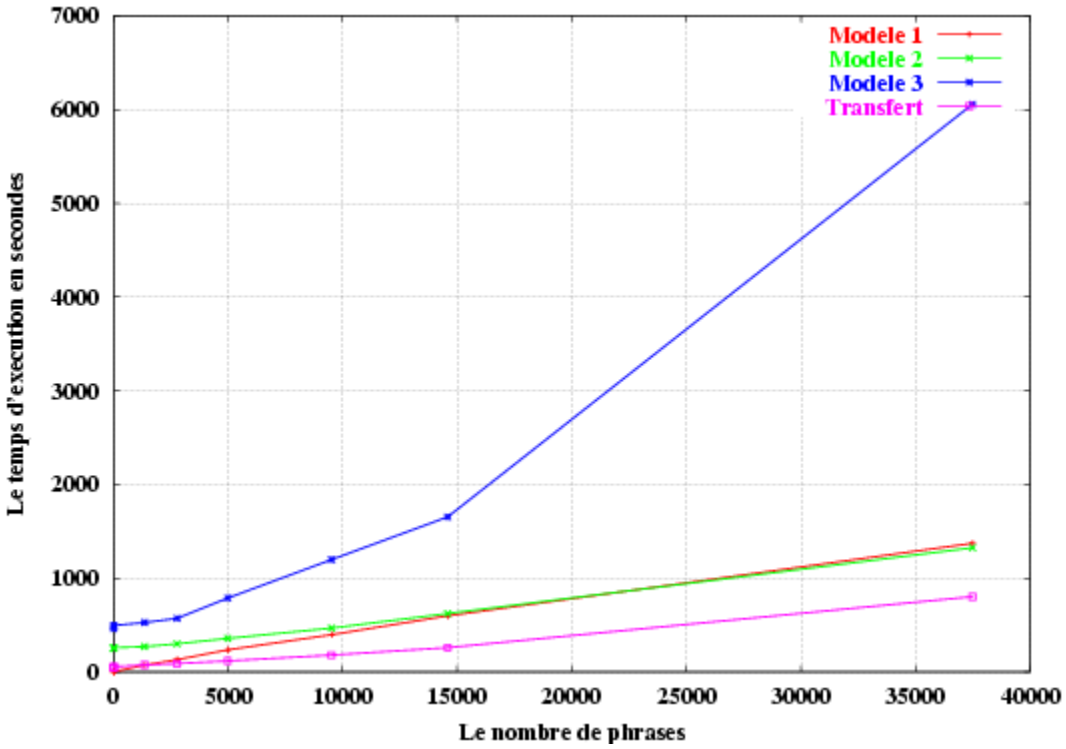

Figure 12: Le temps d'exécution de GIZA sur une machine peu puissante.

Au début, on a lancé le programme sur une machine[11] Pentium II de 128 MO de RAM (figure 12), pour 24 paires de phrases, le temps d'exécution était de l'ordre 809 secondes (13 minutes à peu près). Pour 37 483 paires, on a eu les résultats après 9 594 secondes (2 heures 40 minutes). Par contre, en essayant de lancer GIZA sur un corpus de 100 000 paires avec la même machine (chenonceaux), les temps d'exécution étaient trop importants (près de cinq heures pour une itération qui représente environ 1.5% du temps de traitement).

[11] Chenonceaux, une machine Linux. Pentinum 2, 400 MHZ, 128MO de SDRAM.

A cet effet, nos entraînements ont donc été réalisés par la suite sur des machines plus performantes[12] (clac de mémoire 1 024 MO avec une unité de traitement assez puissante). Le temps maximal (6 057 secondes) est enregistré lors de l'entraînement du modèle 3 sur un corpus de 37 483 phrases (machine lente). Cependant la durée du traitement sur un corpus de 400 000 paires était 3 078 secondes. Par conséquent, sur un corpus 10 fois plus volumineux que le premier, la durée du traitement n'est que la moitié du temps pris par le premier corpus (figure 13).

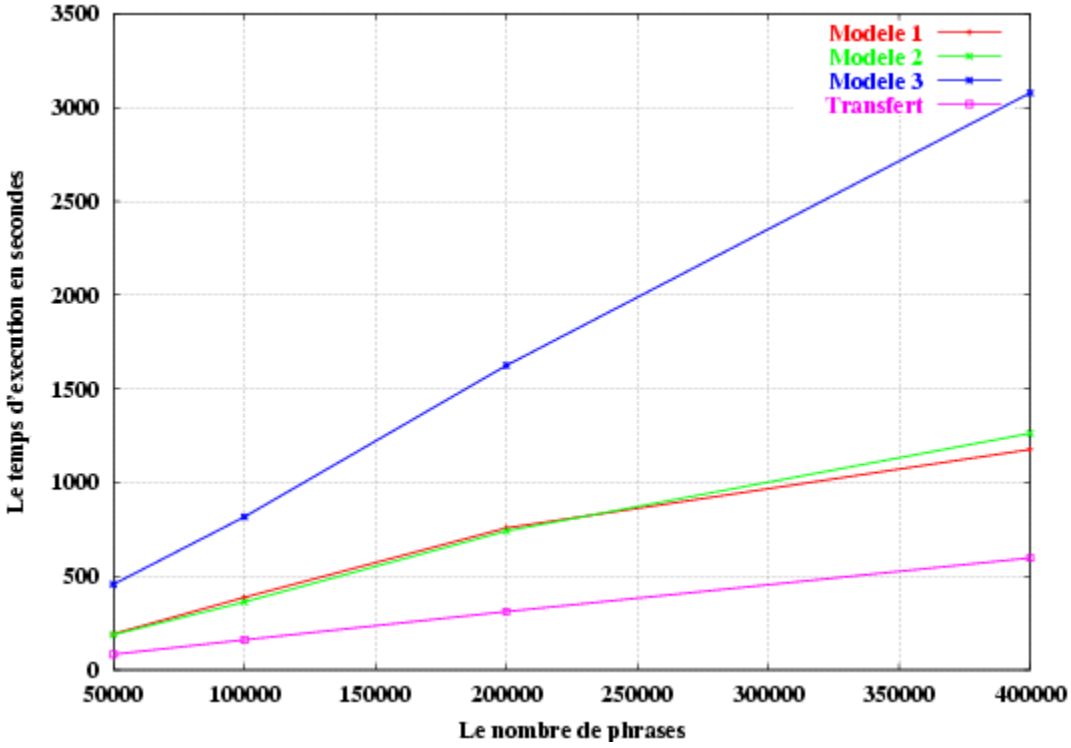

Figure 13: Le temps d'exécution par rapport à la taille du corpus.

Pour l'entraînement des modèles sur la totalité de notre corpus d'entraînement de 1 451 924 de paires de phrases, GIZA a pris environ 4 jours. Les quatre premières heures étaient réservées pour l'entraînement du modèle 1, le modèle 2 a requis 14 heures d'entraînement et le reste du temps étant pris par l'entraînement du modèle 3.

[12] Clac est une machine Lunix de deux processeurs athlon et de 1024MO de mémoire vive.

4.5. Mesure de la qualité d'un modèle de traduction.

Il existe différentes manières d'évaluer la qualité d'un modèle de traduction. Une possibilité consiste à évaluer globalement la performance d'un engin de traduction au complet. Nous reviendrons sur ce mode d'évaluation dans le chapitre 6 et nous intéressons dans cette section à l'évaluation directe bien que critiquable du seul modèle de traduction. Nous utilisons pour cela la perplexité qui représente le nombre moyen d'"hésitations" qu'aurait le modèle s'il devait traduire un texte de manière identique à une traduction de référence. Formellement, la perplexité est donnée par :

$$2^{\frac{-\sum_{i=1}^{S} log\ P(\ f^i|e^i\ )}{N}} \qquad \textbf{(4.1)}$$

Où la probabilité $P(f^i|e^i)$ est donnée par le modèle de traduction évalué. N est le nombre de mots du corpus, S le nombre de paires de phrases. Les algorithmes d'entraînements implémentés dans GIZA optimisent les paramètres des modèles de manière à minimiser la perplexité.

Il est important de noter que la perplexité d'un modèle ne reflète que très indirectement son aptitude à bien traduire ; elle nous donne une indication de son aptitude à reconnaître que deux phrases sont (ou ne sont pas) en relation de traduction selon un modèle mesuré sur le corpus d'entraînement.

Chaque modèle a été entraîné à l'aide de sept itérations auxquelles s'ajoute une itération pour transférer les paramètres d'alignement du modèle 2 au modèle 3 (les deux modèles possèdent en effet les mêmes tables de paramètres de transfert et des distributions d'alignement inversées). Lors de l'initialisation des paramètres du modèle 1, la perplexité est de 66 863.4. Après la première itération, elle passe à 250.8 et tombe à 91.1 à la fin de la 7ème itération. Les itérations sur le modèle 2 baissent la perplexité à 40.9 après 7 itérations (figure 14).

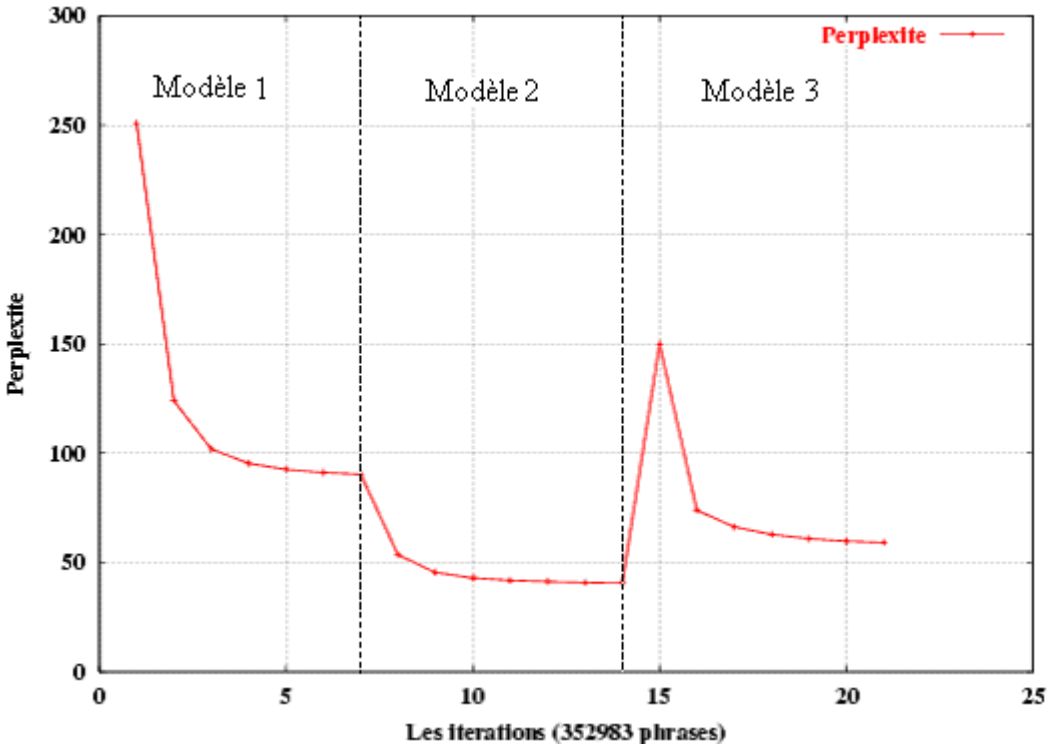

Figure 14: La perplexité par rapport aux itérations des modèles 1, 2 et 3 entraînés sur un corpus de 352 983 paires de phrases françaises / anglaises (la déficience du modèle 3 est démontré par une croissance brusque de la perplexité).

D'autre part, la perplexité décroît après chaque itération sauf quand on transite du modèle 2 au modèle 3 (voir la figure 14). Dans ce cas, ce saut ne justifie pas que le modèle 2 est meilleur. [Brown et al, 1993] disent que c'est normal et expliquent les causes de ce saut (voir [Brown et al, 1993] pour plus de détails). De plus, les expérimentations ont montré le modèle 3 est meilleur que le modèle 2.

4.6. Quelques exemples et chiffres pour commenter les résultats :

Dans cette section on présente plusieurs exemples pris au hasard afin d'illustrer les résultats qu'on a obtenus avec les différents modèles. On montre les alignements, les probabilités de transfert et les fertilités.

4.6.1. Les alignements des mots

Dans les exemples qui suivent, pour chaque modèle, on présente le meilleur alignement de mots obtenu pour cette paire de phrases :

Canadian charter of rights and freedoms (phrase anglaise de 6 mots).
La charte canadienne des droits et libertés (phrase française de 7 mots).

Modèle 1:

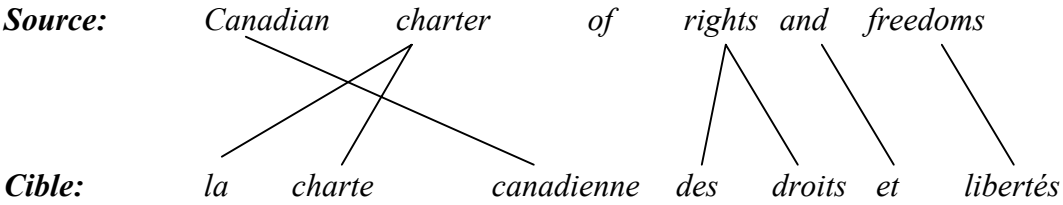

Figure 15: Le meilleur alignement obtenu pour cette paire de phrases par IBM1.

Dans cet exemple, on remarque que le mot "*charter* " est associé à deux mots "*la charte* ". On appelle ce genre d'association "**fertilité indirecte**" du modèle 1 (fertilité de charter est égale à 2) cependant "*of*" n'a pas été traduit par aucun mot (fertilité 0).

Modèle 2 :

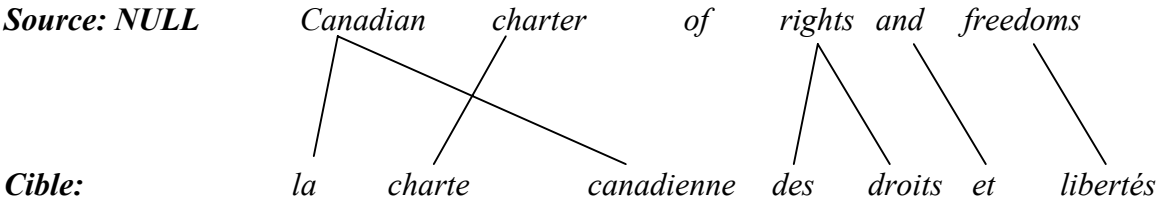

Figure 16: Le meilleur alignement obtenu pour cette phrase par IBM2.

Le modèle 1 suppose que toutes les positions sont équiprobables, cependant le modèle 2 accorde des probabilités d'alignement aux positions des mots français et anglais. C'est une explication possible du fait que dans le cas de l'alignement obtenu par le modèle 2, le mot en première position se connecte à la première position (*canadian -> la*).

Modèle 3:

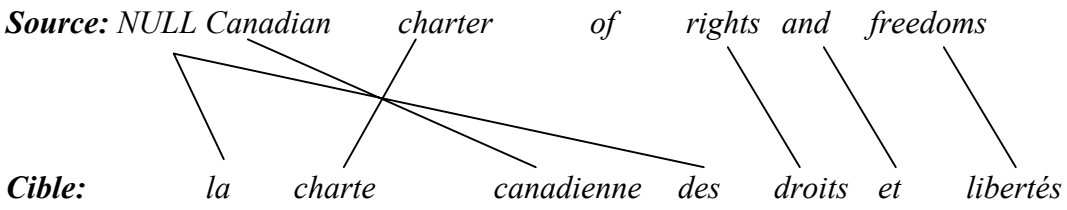

Figure 17: Le meilleur alignement obtenu pour cette phrase par IBM3.

Dans cet exemple, on voit l'effet produit par les paramètres de fertilité. Par exemple, le mot "*charte*" est associé à un seul mot (fertilité 1) toutefois il n'a trouvé aucune traduction pour le mot "*la*" (fertilité 0). Les mots "*la*" et "*des*" sont associés à *NULL* (e_0) et ce qu'on appelle des mots **spurious**.

4.6.2. Les probabilités de transfert :

Nous comparons les modèles de traduction à l'aide des paramètres de transfert sur des exemples que nous avons mentionnés dans la section précédente. Nous aimerions montrer que le modèle 3 est le meilleur.

Modèle 3		***Modèle 2***	
f	***t(f|e)***	***f***	***t(f|e)***
canadienne	0.259923	canadienne	0.2025
canadiens	0.251528	canadiens	0.194901
canadien	0.216328	canadien	0.176973
canadiennes	0.118263	canada	0.0852405
canada	0.107003	canadiennes	0.0772148
notre	0.0113057	les	0.0640237
nos	0.0101566	du	0.0479771
canadian	0.0072526	la	0.0288901
pays	0.00317151	des	0.0286695
nationale	0.00168392	de	0.0197803

Tableau 3: Les dix premières probabilités de transfert du mot *"canadian"* (modèle 2 et 3) prises des résultats d'entraînement sur un corpus de 1.6 millions de phrases.

On observe dans l'exemple du tableau 3 que les deux modèles 2 et 3 donnent des associations pertinentes. De manière intéressante, on observe également que le modèle 3

assigne des probabilités plus élevées aux mots les plus pertinents. Le filtre (seuil de 10^{-8}) qui élimine les associations les moins probables est partiellement responsable de cela. En effet, le modèle 2 assigne à "*canadian*" 122 mots, alors que le modèle 3 en considère seulement 75. Ainsi on peut bien comprendre pourquoi les modèles alignent le mot "*canadienne*" à "*canadian*" de l'exemple précédent.

Un autre exemple, prenons le mot "*rights*".

Modèle 3				***Modèle 2***	
f	***t(f|e)***	***Φ***	***n(Φ|e)***	***f***	***t(f|e)***
droits	0.796545	0	0,0454826	droits	0.608039
des	0.085716	1	0,8399	des	0.190462
droit	0.056123	2	0,113953	les	0.0988315
déclaration	0.0115531	3	0,00031607	droit	0.0307268
aux	0.0066727	4	0,00021108	déclaration	0.0106583
ceux	0.00609777	5	0,000135305	la	0.00914235
leurs	0.00471072	6	0,00011211	personne	0.00824262
charte	0.00280203	7	0	leurs	0.00505768
les	0.0024573	8	0	aux	0.00494197
respecter	0.00185838	9	0	de	0.00433437

Tableau 4: Les dix premières probabilités de transfert du mot *"rights"* (modèle 2 et 3) entraîné sur un corpus de 1.6 millions paires de phrases. Soit n(Φ|e) la probabilité que le mot e a une fertilité *Φ*.

Le modèle 2 traduit le mot "*rights*" de cette phrase (*canadian charter of rights and freedoms*) par les deux mots français "*des droits*", cependant le modèle 3 traduit ce mot par un seul mot "*droits*". On peut expliquer la différence entre ces deux modèles par le fait que le modèle 3 concentre une masse de probabilités sur "*droits*" plus que le modèle 2, et ceci, grâce aux fertilités intégrées dans l'équation du modèle 3 (tableau 4). Noter que la probabilité de fertilité est nulle à partir de 7 car on a autorisé une fertilité maximale de 5 durant l'entraînement.

4.6.3. Les fertilités

Le mot anglais *« update »* se traduit habituellement par le terme « *mise à jour »* ou « *mettre à jour* ». Les probabilités de transfert peuvent capturer cette information en associant ces mots français au mot anglais avec une forte probabilité. Les distributions de

probabilités nous permettent d'apprécier sur cet exemple l'information que capture le modèle 3. Ici, le modèle découvre que « *update* » se traduit de manière très probable par 3 mots.

f	***t(f|e)***		***La fertilité Φ***	***n(Φ|e)***
jour	0.202269		0	0.0963158
à	0.128631		1	0.24774
mettre	0.110322		2	0.242352
mise	0.0673782		3	0.362265
moderniser	0.0540367		4	0.0313557
dire	0.0164854		5	0.0192533
courant	0.01618		6	0.000218907
modernisation	0.0137019		7	0
mis	0.0134613		8	0
actualiser	0.0107915		9	0

Tableau 5: Les probabilités de transfert et de fertilité du mot "update".

Le tableau 5 nous montre que la fertilité 3 est la plus probable et les probabilités de transfert offrent aussi la plupart des traductions (*jour, à, mettre, mise*, etc.…)

Certains mots anglais n'ont pas de traduction exacte en français comme par exemple le mot "*nodding*" (tableau 6). On retrouve par exemple les traductions suivantes: (*Il fait signe que oui | He is nodding*), (*Il fait un signe de la tête | He is nodding*), (*Il fait un signe de tête affirmatif | He is nodding*) ou *(Il hoche la tête affirmativement | He is nodding).* La fertilité élevée de ce mot est capturée par le modèle 3 (voir tableau 6).

f	***t(f|e)***		***La fertilité Φ***	***n(Φ|e)***
signe	0.17294		0	0.0129514
tête	0.168938		1	0.218677
oui	0.0739652		2	0.212233
que	0.0736777		3	0.220905
fait	0.0735548		4	0.324023
hoche	0.0637272		5	0.00819289
hocher	0.0333808		6	0.00301704
un	0.0321521		7	0
assentiment	0.0289505		8	0
me	0.0178129		9	0

Tableau 6: Un exemple des mots anglais ayant une large fertilité "*nodding*".

4.7. Une comparaison entre RMTTK et GIZA.

Pour mettre à l'épreuve notre modèle, nous le comparons avec RMTTK (RALI Machine Translation Toolkit), un package offrant l'entraînement des modèles 1 et 2 implémenté au RALI depuis quelques années. Ainsi, nous avons exécuté les 2 logiciels du RMTTK et GIZA sur le même corpus d'environ 1.4 millions paires de phrases.

On présente brièvement les temps d'exécution, les résultats et enfin quelques distributions obtenues par les deux programmes. Le temps d'exécution des entraînements sous GIZA et RMTTK sont très différents (voir tableau 7). Pour le modèle 1, RMTTK est environ deux fois plus rapide. Pour le modèle 2, RMTTK est de 8 fois plus rapide. On a lancé les deux programmes sur deux machines clac (tableau 7).

GIZA		***RALI***	
Modèle	*Temps en minutes*	*Modèle*	*Temps en minutes*
IBM 1	270 min	IBM 1	86 min~ 1:26h
IBM 2	861 min	IBM 2	82 min~1:21h

Tableau 7: Les temps d'exécution d'un entraînement avec GIZA et RMTTK.

Noter qu'on n'a pas implémenté le modèle 3 au RALI jusqu'à présent.

L'espace mémoire requis par RMTTK est également moindre que celui nécessaire à GIZA, comme on a déjà expliqué 993MO pour le modèle 1 et cela implique une vitesse lente. Cependant RMTTK réserve seulement 559 MO de la mémoire pendant l'exécution. Une raison qui peut provoquer la grande différence entre ces deux outils d'entraînement est que RMTTK a de meilleures structures de données.

On présente maintenant une brève comparaison des modèles IBM2 obtenus par les deux systèmes d'entraînement en terme du nombre de paramètres, des probabilités, et quelques exemples de mots. RMTTK a un nombre de paramètres de 34 969 331, supérieur à celui de GIZA (8 846 847). Ce n'est pas un avantage pour lui, car dès que le nombre de paramètres augmente, alors les probabilités de transfert diminuent du fait que la somme de probabilités vaut 1.

RMTTK offre cependant un mécanisme qui permet de filtrer ces paramètres en fonction de leur gain estimé à la prédiction d'un corpus de test. Nous n'avons pas testé ce mécanisme dans notre travail.

Exemples :

La table 8 représente deux mots pris au hasard, on voit que les 10 premiers mots sont vraiment très proches. Les associations les plus probables obtenues par les deux packages sont assez proches. En revanche, les modèles obtenus divergent rapidement sur les probabilités les plus faibles. Ici, étant la résultante du seuillage effectuée par GIZA, que RMTTK n'effectue pas.

Compute		**Message**	
RMTTK	**GIZA**	**RMTTK**	**GIZA**
Nombre de mots associés			
126	34	5582	408
Mot et probabilité			
calculer 0.16	calculer 0.179	message 0.59	message 0.6
concordent 0.083	concordent 0.093	le 0.035	le 0.0305
comptabiliser 0.042	inventer 0.046	transmettre 0.025	transmettre 0.023
instaurant 0.042	comptabiliser 0.046	comprendre 0.022	comprendre 0.021
pleine 0.042	dus 0.046	transmis 0.016	transmis 0.015
volonté 0.042	volonté 0.046	compris 0.013	compris 0.0136
colle 0.042	colle 0.0466	clair 0.012	clair 0.012
dus 0.042	parfait 0.04664	que 0.01	nous 0.0083
inventer 0.042	correspond 0.0463	nous 0.0091	entendre 0.007
compter 0.041	tels 0.0454	entendre 0.0085	dire 0.0073

Tableau 8: Deux mots exemples de RMTTK et GIZA, on a seulement pris les dix premières probabilités pour chaque mot.

4.8. Conclusion :

Nous avons montré que GIZA, bien que plus gourmand en temps et en espace mémoire étant un package viable pour l'entraînement de modèles IBM. Des paramètres obtenus par GIZA sont légèrement différents de ceux obtenus par le package RMTTK, ce qui est la résultante du filtrage opéré par GIZA.

La performance, la simplicité, l'accès simple et rapide aux paramètres d'un modèle de traduction sont des facteurs principaux cherchés par les programmeurs ayant pour objectif de concevoir un algorithme efficace et performant qui traduit rapidement. D'après nos expériences, nous avons montré que GIZA répond à ces préoccupations. Nous utilisons donc les paramètres obtenus par l'entraînement de cet outil afin de développer nos décodeurs.

Chapitre 5

Décodage

Nous avons abordé dans les sections précédentes le problème de l'entraînement des modèles. Nous nous concentrons dans ce chapitre sur le troisième problème de la traduction statistique « le décodage ». Nous expérimentons, ici des décodeurs traduisant du français vers l'anglais, bien qu'en principe, les modèles utilisés sont indépendants de la paire de langues.

Dans la traduction automatique probabiliste, le but d'un décodeur est de chercher la phrase anglaise $e^I=e_1...e_I$ la plus probable étant donnée une phrase source française $f^J=f_1,...,f_J$ et des modèles (modèle de langue et modèle de traduction) où I et e_i $i\in[1,I]$ sont des inconnus (figure 18).

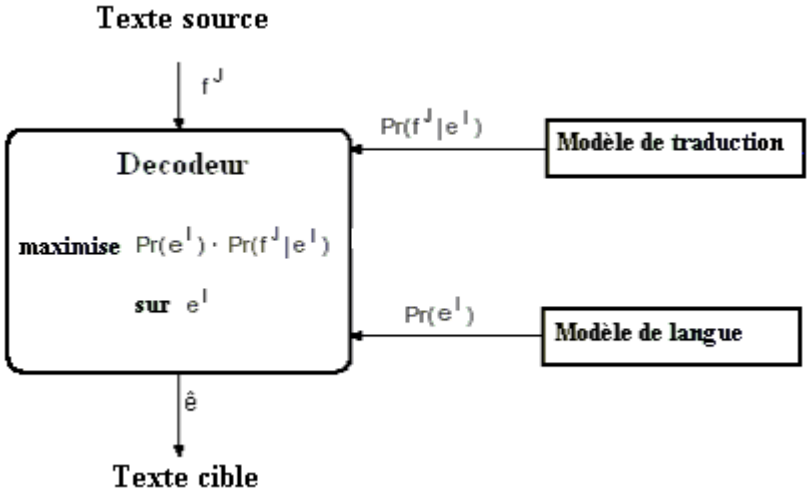

Figure 18: L'architecture de la traduction probabiliste [Nießen et al. 1998].

Chaque phrase anglaise est considérée comme une traduction possible de la phrase source française. On assigne à chaque paire de phrases (e^I,f^J) une probabilité $P(e^I|f^J)$. Il faut chercher un I_{opt} optimal et de même une phrase $\hat{e}^{I_{opt}}$ qui maximisent $P(e^I|f^J)$.

Revenons aux équations et modèles vus dans le chapitre 2, (2.1 et 2.2) :

$$\hat{e}_{Iopt} = \underset{I,e^I}{arg\,max}[\,P(e^I \mid f^J)\,] = \underset{I,e^I}{arg\,max}[\,P(e^I)\times P(f^J \mid e^I)\,] \qquad \textbf{(5.1)}$$

Bien qu'il soit possible d'écrire un décodeur pour le modèle 3, nous avons concentré nos efforts sur l'écriture de deux décodeurs (DP et Greedy et nous les présentons plus tard dans ce chapitre) pour le modèle 2. Ce modèle est en effet plus simple et le décodeur DP résultant de ce modèle est donc plus rapide.

L'équation du modèle 2 donnée par [Brown 1993] est la suivante :

$$P(f^J \mid e^I) = P(I \setminus J) \prod_{j=1}^{J} \sum_{i=0}^{I} P(f_j \mid e_i).P(i \mid j, J, I) \qquad \textbf{(5.2)}$$

Comme nous l'avons déjà mentionné dans les chapitres précédents, nous utilisons un modèle trigramme pour construire nos décodeurs, l'équation de la maximisation (5.1) devient alors :

$$\hat{e} = \max_{I}[P(J \mid I) \max_{e^I}\{\prod_{i=1}^{I} \max_{i}[P(e_i \mid e_{i-1}e_{i-2})P(i \mid j, J, I)P(f_j \mid e_i)]\}] \qquad \textbf{(5.3)}$$

Un modèle de longueur est également mis en jeu tel que spécifié dans l'équation (5.3). Dans cette étude nous avons fait l'hypothèse que la longueur (comptée en mots) d'une phrase française, traduction d'une phrase anglaise était normalement distribuée.

Notez que dans la dernière équation, la somme sur tous les alignements a été remplacée par une maximisation. Ceci correspond à une hypothèse sous-jacente (souvent faite) que la probabilité de l'alignement le plus probable domine la somme. On parle souvent de *maximum approximation*. Cette simplification permet de factoriser certains calculs et diminue donc la complexité calculatoire de décodeur. Il a en effet été démontré que l'opération de maximisation de l'équation 5.1 est NP-complète [Knight et al, 1999].

Dans ce chapitre, on présente et on compare deux algorithmes de recherche :

1- Le premier est une technique de programmation dynamique (*DP*) pour parcourir une partie importante de l'espace de recherche.
2- Le second est un algorithme *greedy* qui ne parcours de manière « adhoc » qu'un très petit sous-ensemble de l'espace de recherche.

La comparaison entre les deux méthodes de décodage étant notre première préoccupation.

5.1. L'algorithme DP[13]

5.1.1. Principe

Plusieurs décodeurs DP ont été proposés dans la littérature. Par exemple, [Tillmann et al., 1997] proposent un décodage où les alignements considérés obéissent à une hypothèse markovienne d'ordre 1: (l'alignement d'un mot au temps i est conditionné par l'alignement au temps i-1). Les auteurs font de plus l'hypothèse que les alignements sont monotones (pas de croisements). Sous ces contraintes, leur algorithme est capable de proposer de manière efficace une traduction.

Ce type d'algorithme contraint les traductions produites à être littérales, ce qui n'est en pratique pas souhaitable. C'est pour cela que [Nießen et al. 1998] ont proposé un décodeur où de telles hypothèses ne sont pas faites. Le prix à payer étant une complexité accrue, nous reviendrons sur ce point plus tard.

Ce décodeur basé sur la programmation dynamique examine les mots anglais séquentiellement. Pour qu'une phrase anglaise soit considérée comme une traduction complète de la phrase française, il faut qu'elle couvre toutes les positions françaises. L'idée de cet algorithme est d'étendre progressivement (mot à mot) les hypothèses de traduction, tout en couvrant progressivement les positions de la phrase française. Chaque mot français peut générer au plus un mot anglais à n'importe quelle position dans la phrase anglaise.

Nous étendons l'algorithme proposé par [Nießen et al. 1998] pour qu'un modèle trigramme soit utilisé à la place du modèle bigramme initial.

5.1.2. Description

Cet algorithme de recherche interprète la notion d'alignement de mots comme suit: chaque position i en e_1^I est assignée à une position $b_i=j$ en f_1^J.

À chaque position i de e_1^I, chaque mot du vocabulaire actif anglais peut être inséré. De plus, on permet qu'un mot anglais e_i soit aligné avec l mots français consécutifs (une sorte de fertilité non modélisée). Dans la plupart de cas, la fertilité optimale est égale

[13] Une description mathématique de l'algorithme est dans [Nießen et al. 1998].

à 1. Il est possible que le mot e_i ait la fertilité zéro, ce qui signifie que ce mot ne corresponde directement à aucun mot français (spurious).

D'un point de vue formel, ce que nous cherchons à optimiser se décrit par:

$$\hat{e} = \max_{I}[P(J \mid I) \max_{e^I} \prod_{i=1}^{I} [P(e_i \mid e_{i-1}e_{i-2}) \max_{j,l} \prod_{k=j-l+1}^{j} \{P(i \mid k,J,I)P(f_k \mid e_i)\}]] \quad \textbf{(5.4)}$$

Où l désigne la fertilité du mot e_i (on considère dans nos expériences une fertilité maximale de 3 mots français pour un mot anglais).

Cette équation ne garantie pas que toutes les positions sources sont couvertes. En d'autres termes, on doit forcer l'algorithme à couvrir toutes les positions françaises. [Nießen et al. 1998] proposent plusieurs stratégies pour résoudre ce problème (comme par exemple l'introduction d'une pénalité pour chaque position française non couverte) mais ils suggèrent une solution basée sur l'introduction d'un nouveau paramètre dans le critère de DP. Soit $Q_l(c,i,j,e)$ la probabilité du meilleur chemin arrivant en position i dans e^I, en position j dans f^J et tel que $e_i=e$ et c positions sources ont été couvertes. (Pour tous les détails voir [Nießen et al. 1998])

À noter que les coordonnées d'une hypothèse sont déterminées par les quatre paramètres (c,i,j,e). De ce fait, l'espace peut-être codé par une matrice à quatre dimensions; chaque item dans cet espace de recherche contenant des informations de chaînage arrière (*backtracking*) ainsi que le score de l'hypothèse associée.

Cette quantité $Q_l(c,i,j,e)$ est définie récursivement : [Nießen et al. 1998] présentent deux cas :

Cas simple: le mot e_i n'a pas de correspondant dans le texte français (*skip*).

$$Q_I^S(c,i,j,e) = \max_{e'}\{P(e \mid e'e'')Q_I(c,i-1,j,e')\} \quad \textbf{(5.5)}$$

On recherche dans la table la meilleure façon d'arriver à (e,i,j) depuis une cellule précédante.

Cas général : Dans l'équation (5.3), chaque position j de la phrase française est exactement alignée à une position anglaise i. De cela, si e_i est associé à l mots français ($l>0$), on vérifie qu'aucune de ces positions françaises ne sont déjà couvertes : on définit

une fonction *v(c,l,j',j,e')* qui retourne 1 si les *l* positions de f_k à f_j sont libres dans la phrase française; 0 sinon.

$$Q_I^N(c,i,j,e) = \max_{l>0}[\prod_{k=j-l+1}^{j}\{P(i \mid k,J,I).P(f_k \mid e_i)\}\times$$

$$\max_{e',e''}\{P(e \mid e'e'')\times$$

$$\max_{j'}[Q_I(c-l,i-1,j',e').v(c,l,j',j,e')]\}] \quad \textbf{(5.6)}$$

En gros, on cherche la meilleure fertilité, et la meilleure position française libre (via *v*).

Note : les choix *l, j'* et *e'* sont mémorisés pour pouvoir à la fin reconstruire la traduction « optimale ».

Nous avons alors tous les ingrédients de notre récursion:

$$Q_I(c,i,j,e) = max\{Q_I^S(c,i,j,e), Q_I^N(c,i,j,e)\} \quad \textbf{(5.7)}$$

La meilleure traduction qu'on peut trouver est en maximisant la longueur de la phrase anglaise *I* et recouvrant toutes les positions françaises *J*. Ainsi :

$$\max_I\{P(J \mid I).\max_{j,e} Q_I(J,I,j,e)\} \quad \textbf{(5.8)}$$

Chaque traduction possible a un score (le score de dernier mot de la phrase) calculé par l'accumulation des scores des items précédents (factorisation afin de réduire le temps de calcul). On choisit alors la phrase ayant le meilleur score et couvrant toutes les positions sources. La traduction est alors déterminée par le retour en arrière (*backtracking*).

5.1.3. Filtrage

Si nous parcourons l'ensemble de l'espace de recherche, nous trouvons l'alignement optimal au sens des modèles. Cependant, puisque le nombre des hypothèses augmente exponentiellement avec la longueur de la phrase, il est impraticable d'énumérer toutes les hypothèses possibles. On sacrifie alors l'optimalité pour la rapidité.

En effet, La figure 19 montre que la longueur de la phrase a une influence importante sur le temps de traduction. D'après [Nießen et al. 1998], la complexité de l'algorithme est:

$O(I_{max}^2.J^3.|\varepsilon|^2)$ où |ε| est la taille du vocabulaire cible, J est la longueur de la phrase source, I_{max} est la longueur la plus grande envisagée pour la phrase cible.

[Nießen et al. 1998] proposent cependant des optimisations qui permettent d'accélérer l'algorithme au détriment de la "souplesse" de son critère pour une complexité finale en $O(I_{max}.J^2.|\varepsilon|^2)$.

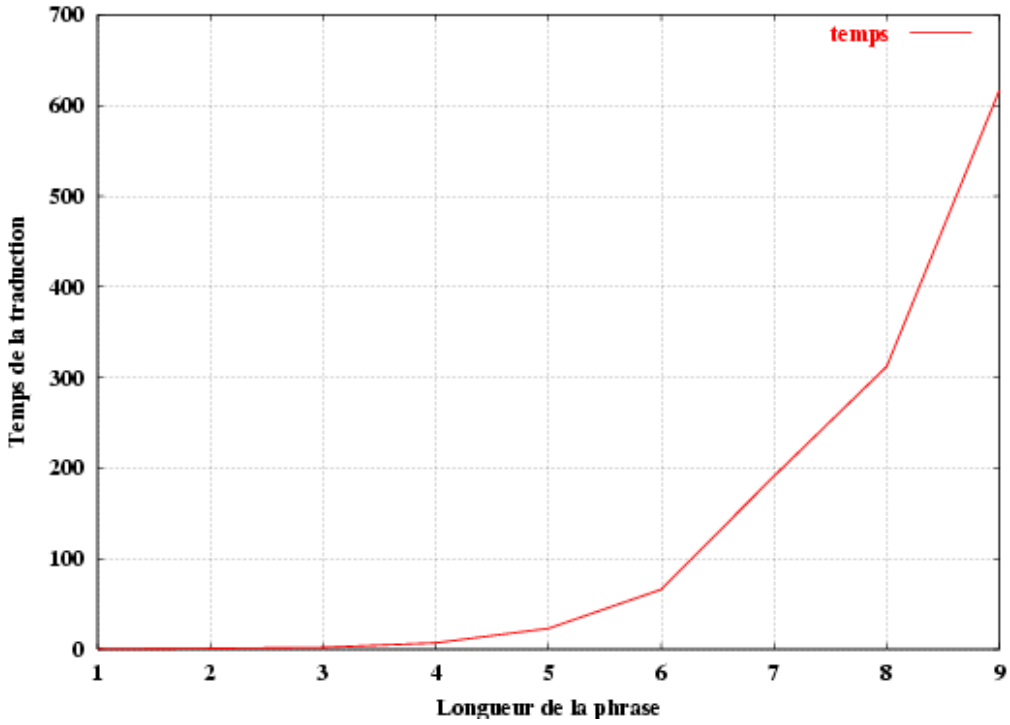

Figure 19: L'accroissement du temps de la traduction avec la longueur de la phrase.

Tout d'abord on peut limiter les couvertures sources par deux bornes majorante et minorante pour chaque niveau (ligne *i*) dans la matrice. En d'autres termes, on peut borner la recherche du meilleur *l* et du meilleur *k* (de l'équation 5.4) par un faisceau [$c_{min}(i)$, $c_{max}(i)$] centré autour de la diagonale que l'on observerait si l'ordre des mots dans les deux phrases était le même;

Avec $c_{min}(i) = [i\frac{J}{I}] - r$, $c_{max}(i) = [i\frac{J}{I}] + r$

Où r est une constante (fixée à 3 dans nos expériences et qui représente un degré de tolérance à cette hypothèse de synchronisation des traductions.

Dans l'équation (5.4), on maximise sur I, la longueur de la traduction produite. En pratique, ça signifie que l'on recommence la programmation dynamique depuis le début, car le critère dépend de I par les probabilités d'alignement *P(i|k,J,I)*. C'est pratiquement trop lourd pour notre algorithme qui doit être rapide. Ils proposent de remplacer dans le critère la valeur exacte de I par son estimée (autrement dit, on retire une dépendance directe

à I dans les probabilités d'alignement pour introduire l'estimée de I): $P(i|k,J,I_{max})$. Dans nos expériences, nous estimons I_{max} par |I + log(I+1)|.

Même après ces contraintes sur l'algorithme, dès que la phrase française arrive à une longueur de 7 mots, le nombre de mots dans le vocabulaire actif devient trop important et impose un filtrage.

Au moins trois options de filtrage sont possibles en prenant compte de :

1. Le nombre de mots anglais associés à chaque mot source français (**N**).
2. Le score de l'hypothèse et les probabilités (transfert et alignement).
3. Les positions de mots sources et ses traductions.

Les tableaux[14] 11 et 12 représentent une illustration en deux dimensions de la table de recherche, chaque case de tableau contient au moins **N** items. On peut remarquer sur les tableaux (11 et 12) l'effet de la contrainte sur le nombre de mots associés à chaque mot français. Après l'expérience, on trouve qu'à partir de 7 mots anglais associés, les résultats restent acceptables en terme de performance et de temps de calcul.

Les positions des mots anglais				
	5	348	348	348
	4	348	348	348
	3	348	348	348
	2	348	282	216
	1	150	100	50
		nous	**avons**	**vu**
		Les mots français		

Tableau 9: Une phrase de 3 mots avec 50 mots anglais associés à chaque mot français, 116 vocabulaires actifs.

Les positions des mots anglais				
	5	690	690	690
	4	690	690	690
	3	690	690	690
	2	690	560	430
	1	300	200	100
		nous	**avons**	**vu**
		Les mots français		

Tableau 10: Une phrase de 3 mots avec 100 mots anglais associés à chaque mot français, 230 vocabulaires actifs.

Une grande taille du vocabulaire actif |ε| amène à un nombre important d'hypothèses. Le grand nombre d'hypothèses augmente l'espace de mémoire occupé par la table de recherche, ce qui augmente les risques de saturation de la mémoire.

[14] **Les nombres notés dans les cases de ces deux tableaux représentent les nombres des items**

Comme on l'a déjà mentionné, la taille du vocabulaire actif |ε| introduit dans la complexité de l'algorithme ($O(I_{max}.J^2.|\varepsilon|^2)$), de ce fait le temps de traduction augmente exponentiellement avec la taille du vocabulaire actif (figure 20).

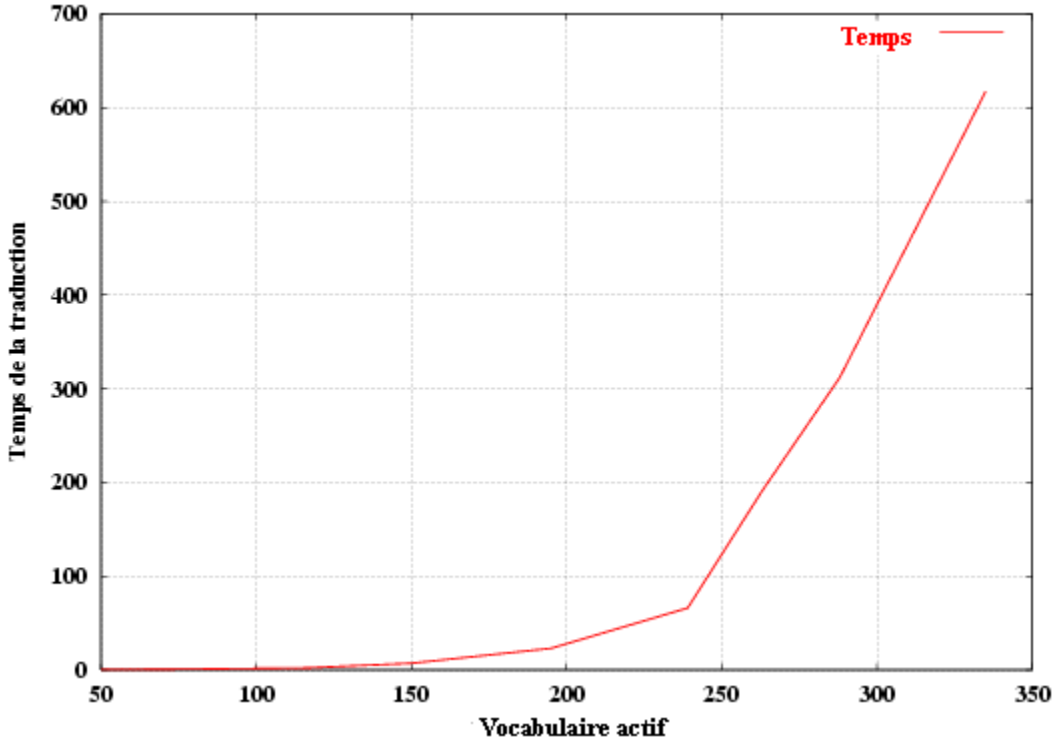

Figure 20: L'accroissement de temps avec la taille de l'ensemble de vocabulaire actif (une phrase de 6 mots français et sa traduction de 6 mots anglais).

Des critères additionnels sur les scores ont également été ajoutés. On ne considère que les hypothèses ayant des scores supérieurs à un seuil : (par exemple le score maximal multiplié par un nombre BETA). Dans certains cas, on perd des bonnes hypothèses et cela influence les résultats.

Un filtrage des probabilités de transfert $P(f_j|e_i)$ et d'alignements $P(i|j,J,I)$ est aussi utile. On ignore toutes les probabilités inférieures à un certain seuil (par exemple $Pr>10^{-3}$).

Évidement, en relaxant les contraintes sur les positions de mots (tous les positions sont possibles avec une probabilité d'alignement $P(i|j,J,I)$), le nombre d'alignements est important ($(I+1)^J$). Afin de réduire ce nombre, on introduit des contraintes sur les positions de mots. Le mot français de la position i peut se connecter aux mots anglais situés entre deux bornes inférieures et supérieures (par exemple $[i-3;i+3]$). Nous vérifierons l'impact des filtrages dans la section 5.3.

5.1.4. Implémentation

Entrée ***: $f_1 \ldots f_j \ldots f_J$***

Choisir une longueur maximale : ***I_{max}***

Sélectionner les vocabulaires actifs.

Initialiser la table de recherche ***Space***

Pour toutes les positions cibles ***$i=1,2,\ldots,I_{max}$***

. *Pour toutes les positions sources* ***$j=1,\ldots J$***

. . *Pour toutes les hypothèses* ***H*** *dans* ***Space(i,j)***

. . . ***cv****=couverture de l'hypothèse.*

. . . *Mise à jour des positions libres.*

. . . *Pour toutes les positions libres* ***k*** *varie de* ***1 à J****.*

. . . . *Pour tous les mots associés* ***e*** *au mot source* ***f_k***

. . . . . ***S=0***

. . . . . *Pour les fertilités* ***f*** *de* ***1*** *à* ***min(J,MAX_FERTILITE)***

. . . . . ***S*** *+=* ***score H+log $(P(f_k|e_i))$+log$(P(e_i|e_{i-1},e_{i-2}))$+log(a(i\|k,l,m))***

. . . . . *Mise à jour dans* ***space(k,j+1)*** *l'hypothèse(****mot,cv,f,S,Mot Précédant=i,j****)*

. . . *Pour tous les mots de l'ensemble de vocabulaire actif*

. . . . ***S*** *=* ***score H+log(P(e_i\|e_{i-1},e_{i-2}))+log(a(0\|k,l,m))***

. . . . *Mise à jour dans* ***space (i, j+1)*** *l' hypothèse(****mot, cv, f, S, Mot Précédant=i,j****)*

Chercher le score maximum et *backtracking***:**

Pour tous les ***i*** *de* ***1*** *à* ***I_{max}***

. *Pour toutes les hypothèses* ***H*** *dans* ***Space(i,j)***

. . *S=****Score H+P(J\|i)***

. . *Si((****c==J****) et (****s>max_s****) alors*

. . . ***max_s =s***

. . . ***(max_i, max_j, ,max_e)****=(i,j,e)*

retourner l'hypothèse ***H_{max}, max_i, max_j.***

Sortie: ***$e_1,\ldots e_i,\ldots e_{maxi}$***

Figure 21: L'algorithme de recherche DP.

Nous présentons l'algorithme qui implémente l'idée de [Nießen et al. 1998]. L'entrée de l'algorithme est le texte source à traduire et la sortie est le texte traduit dans la langue cible.

L'algorithme mémorise toutes les informations nécessaires dans une table que nous illustrons dans un exemple (section 5.1.5). Il est divisé en trois phases:

-Dans la première phase, il s'agit de la préparation de données pour initialiser la table de recherche et sélectionner le vocabulaire actif.

-La seconde partie est la principale qui remplit la table de recherche. Pour remplir la case (i,j), l'algorithme cherche dans toutes les cases de la ligne *i-1* et dans les positions sources libres *k*.

-La troisième partie de l'algorithme implémente la méthode du chaînage en arrière (*backtraking*) qui sert à chercher la meilleure traduction. Enfin, construire le texte traduit.

5.1.5. Les problèmes rencontrés durant l'implémentation

Dans cette section, on parle des principaux problèmes rencontrés durant l'implémentation de cet algorithme:

- Estimer la longueur de la phrase cible parce que dans la plupart des cas la phrase source n'a pas la même longueur que sa traduction.
- Sélectionner les traductions possibles de chaque mot source.

Une phrase française de 10 mots peut aussi bien être traduite par une phrase anglaise de 8 que par une phrase de 13 mots. Sur un corpus de 66 paires de phrases (français/anglais) telles que les phrases françaises sont de 10 mots, la longueur des phrases anglaises se varie entre 6 et 14 mots anglais, la moyenne est de 9.64 mots. (Voir la distribution, figure 22)

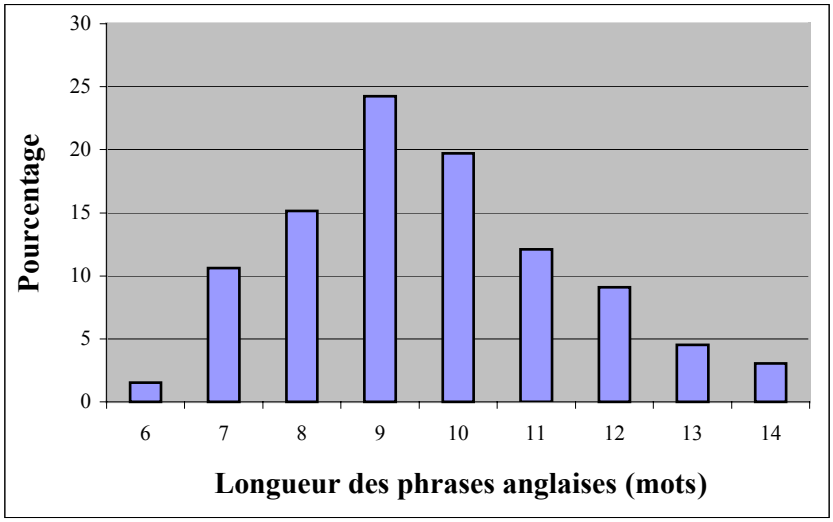

Figure 22: La distribution de longueur des traductions anglaises des phrases françaises de 10 mots.

À ce fait, on trouve plusieurs problèmes lors des calculs des probabilités *P(I|J)* et *P(i|j,J,I)* de l'équation (5.4) ainsi que la détermination d'une des dimensions de la table de recherche. Nous avons abordé ce problème dans la section (5.1.2).

Fixant une longueur maximale (I_{max}) pour la traduction, le modèle de longueur utilisé dans l'équation (5.8) peut accorder une probabilité *P(I|J)* à chaque paire de phrases de longueur I et J où I varie entre 1 et I_{max}.

Nous nous intéressons maintenant au problème de la sélection du vocabulaire actif qui peut être associé à chaque mot français. Nous utilisons pour cela le modèle de transfert $P(e_i|f_j)$. Formellement nous cherchons :

$$\hat{e} = \arg\max_{ei} P(e_i|f_j) = \arg\max_{ei} \; P(f_j|e_i) \times P(e_i) \qquad \textbf{(5.9)}$$

Où $P(f_j|e_i)$ est la probabilité de transfert donnée par le modèle de traduction IBM2 et $P(e_i)$ est la probabilité donnée à e_i par un modèle unigramme.

Exemple :

Étant donnée cette phrase à traduire **: *nous avons vu***

Selon notre modèle de transfert, On trouve que ***nous*** est associé à 6874 mots anglais, ***avons*** à 3608 mots et ***vu*** à 2176 mots. On voit clairement à partir de cet exemple

que chaque mot français est associé à un nombre important de mot anglais alors un filtrage est nécessaire pour réduire ce nombre de mots. D'habitude on sélectionne les meilleures **N** traductions.

En appliquant la formule (5.9), voici les 7 mots anglais les plus probablement associés à chaque mot français. Nous comparons ce score à celui obtenu en ne tenant pas compte du modèle unigramme. En pratique nous observons qu'il est préférable d'utiliser l'équation (5.9). (les mots sont classés selon les scores de probabilités par ordre décroissant).

$P(f_j|e_i)$ *nous :* we us ourselves canting beckoning radiator the
$P(f_j|e_i)\times P(e_i)$ *nous :* we us our the this it that

$P(f_j|e_i)$ *avons :* our have need because saw we heard
$P(f_j|e_i)\times P(e_i)$ *avons :* we have our is was been heard

$P(f_j|e_i)$ *vu :* seen saw amputee cartoonist watched seely given
$P(f_j|e_i)\times P(e_i)$ *vu :* seen saw see given had been because……

5.1.6. Exemple

Dans cette section, on présente un exemple afin d'illustrer l'algorithme, son fonctionnement, la modalité de recherche, le meilleur score et enfin comment construire la traduction par le *backtracking*.

La phrase française à traduire est : ***Nous avons vu***

Noter que dans nos expériences, les valeurs des probabilités sont remplacées par leur logarithme (à base deux) (Log[P]). Ceci nous permet, et de simplifier nos calculs et de gérer les problèmes de précisions dûes à la multiplication des termes entre 0 et 1.

On commence par faire le choix d'une longueur maximale de la traduction. la phrase française contient trois mots, nous décidons arbitrairement que la traduction n'en

contiendra pas plus de 5. Un modèle de longueur est appliqué pour déterminer la longueur cible étant donnée la longueur source.

On décide ensuite d'un nombre maximum de mots associable à chaque mot français. Ceci n'est pas commandé par l'équation (5.4) mais permet en pratique de réduire les temps de calcul. Dans cet exemple, nous prenons comme valeur 10.

La table de recherche est donc dans notre exemple de dimensions 5 par 3. Chaque case (*i,j*) mémorise le dernier mot de l'hypothèse e_i aligné au(x) mot(s) français f_j^{j+l-1}. Une information dans une case est appelée un *item.*

Un sous-ensemble des items construits lors de la recherche est illustré en figure 20. Chaque item est décrit par un 6-uplet (e_i,c,f,s,j,e_{i-1}) où e_i est le mot anglais, c est le nombre de mots français couverts, f est la fertilité de e_i, s est le score de l'hypothèse, j et e_{i-1} sont respectivement la position et le mot précédent, ce sont des informations de *backtraking*.

Par exemple : *(saw, 3, 2, -31.6374, 0, we)* ; *saw* est le mot, 3 est la couverture, 2 est la fertilité, -31.6374 est le score de l'hypothèse, cette hypothèse est une extension d'une autre hypothèse se terminant par we. L'information de l'hypothèse avant extension se trouve en ligne précédente (c'est toujours le cas) et colonne 0 (information de retour en arrière). L'espace de recherche est organisé sous la forme d'une table à 4 dimensions T. L'hypothèse *(saw, 3, 2, -31.6374, 0, we)* s'y trouve à la case T[1,1,saw,3]. Les cellules contenues dans T[1,1] contiennent toutes les hypothèses en cours dont le premier mot cible est aligné avec le premier mot source.

La table est remplie en étendant chaque hypothèse valide en position *i-1* par un nouveau mot en position *i*. Un mot peut être aligné à 0,1 ou … *l* mots français dans ce cas le score associé à une hypothèse h est étendu par le score du mot précédent, le score donné par le trigramme $P(e_i|e_{i-1}\ e_{i-2})$ et le modèle de traduction IBM2 en n'alignant e_i à aucun mot à *1 ou… l* mots (fertilité modélisée).

Plus formelle: pour le mot e_i de fertilité de l le score S de l'hypothèse est :

$$S = \underbrace{Score(e_{i-1})}_{score\_du\_mot\_précedent} + \underbrace{Log_2[P(e_i \mid e_{i-1}e_{i-2})]}_{Trigramme} + \underbrace{\sum_{k=j-l+1}^{j}\{Log_2[P(i \mid k,J,I)] + Log_2[P(f_k \mid e_i)]\}}_{Modèle\_de\_traduction\_IBM2\_fertilité\_modélisée}$$

(be, 1, 0, -118.821, 0, to) (tell, 1, 0, -124.36, 0, to) *(we, 2, 1, -75.7717, 2, 4)* (forced, 2, 0, -101.34, 0, we) (tell, 2, 1, -85.749, 2, to) *(have, 3, 1, -75.8361, 1, we)* (situation, 3, 0, -104.003, 0, a) (tell, 3, 1, -85.9551, 1, we)	*(be, 1, 0, -140.719, 1, to)* (tell, 1, 0, -146.258, 1, to) *(we 2, 1, -78.7303, 2, ,)* (forced, 2, 0, -105.497, 1, we) (tell, 2, 0, -109.247, 1, we) *(have, 3, 1, -63.4354, 2, we)* (situation, 3, 0, -95.7367, 1, this) (tell, 3, 0, -92.1114, 1, we)	*(to, 1, 0, -85.4202, 2, going)* (tell, 1, 0, -92.0437, 2, to) *(of, 2, 0, -68.5801, 2, 1411)* (situation, 2, 0, -76.4878, 2, the) (forced, 2, 0, -79.8977, 2, of) ***(the 3, 0, -58.9385, 2, in)*** (situation, 3, 0, -64.3124, 2, the)
(to, 1, 0, -90.6244, 0, going) (forced, 1, 0, -98.8266, 0, have) *(we, 2, 1, -64.7879, 2, time)* (tell, 2, 1, -74.6024, 2, to) *(we, 3, 1, -64.8327, 1, time)* (situation, 3, 0, -100.527, 0, our) (tell, 3, 1, -76.9989, 1, us)	*(to, 1, 0, -108.929, 1, going)* (tell, 1, 0, -115.02, 1, to) *(have, 2, 1, -64.7508, 2, we)* (tell, 2, 0, -86.8764, 1, to) *(we, 3, 1, -49.9875, 2, that)* (situation, 3, 0, -79.4187, 1, a) (tell, 3, 0, -81.5409, 1, seen)	*(, , 1, 0, -70.1086, 2, time)* (forced, 1, 0, -77.7886, 2, have) *(the 2, 0, -53.5531, 2, in)* (situation, 2, 0, -59.2954, 2, the) ***(the 3, 0, -44.0328, 2, seen)*** **(in, 3, 0, -47.2669, 2, seen)**
(have, 1, 0, -63.3408, 0, could) (forced, 1, 0, -71.1479, 0, are) *(us, 2, 1, -51.8878, 2, to)* (forced, 2, 0, -60.5926, 0, are) *(we, 3, 2, -55.6962, 2, what)* (situation, 3, 0, -82.2798, 0, the) (forced, 3, 0, -75.8484, 0, we)	*(to, 1, 0, -78.3212, 1, have)* (forced, 1, 0, -83.708, 1, have) *(have, 2, 1, -42.2313, 0, could)* (forced, 2, 0, -55.564, 1, have) *(seen, 3, 1, -38.2634, 2, have)* (forced, 3, 0, -74.0786, 1, have)	*the 1, 0, -53.954, 2, to)* (forced, 1, 0, -64.2729, 2, was) *(the 2, 0, -37.656, 2, saw)* (situation, 2, 0, -51.7445, 2, see) ***(seen, 3, 1, -28.5164, 1, have)*** (forced, 3, 1, -44.5886, 1, have)
(have, 1, 0, -35.2632, 0, we) (forced, 1, 0, -44.4243, 0, we) *(we, 2, 1, -23.9373, 1, we*) (situation, 2, 0, -54.0669, 0, we) (tell, 2, 1, -32.5504, 1, we) *(we, 3, 2, -38.1071, 2, when)* (situation, 3, 0, -83.3688, 0, we) (tell, 3, 1, -55.5457, 1, saw)	*(have, 1, 0, -46.3813, 1, we)* (forced, 1, 0, -57.7353, 1, we) ***(have, 2, 1, -18.2373, 0, we)*** (forced, 2, 0, -81.2807, 1, saw) ***(saw, 3, 2, -31.6374, 0, we)*** (saw, 3, 2, -31.6374, 0, we) (are, 3, 2, -56.904, 0, we)	*(the 1, 0, -39.3, 2, since)* (saw, 2, 1, -23.5968, 0, we) *(situation, 2, 1, -38.1285, 0, we)* (saw, 2, 1, -23.5968, 0, we) *(saw, 3, 1, -31.9204, 0, we)* (saw, 3, 1, -31.9204, 0, we)
(we, 1, 1, -11.458, 0, Début) (give, 1, 1, -26.3494, 0, Début) (must, 1, 1, -30.0417, 0, Début) *(we, 2, 2, -19.7816, 0, Début)* (give, 2, 2, -54.4216, 0, Début) (must, 2, 2, -58.114, 0, Début) ***(we, 3, 3, -49.0835, 0, Début)*** (now, 3, 3, -64.9999, 0, Début) (give, 3, 3, -83.7236, 0, Début)	*(we, 1, 1, -19.1016, 0, Début)* (found, 1, 1, -37.7142, 0, Début) (have, 1, 1, -38.1074, 0, Début) (seen, 1, 1, -33.8516, 0, Début) *(saw, 2, 2, -40.8197, 0, Début)* (found, 2, 2, -51.2493, 0, Début) (have, 2, 2, -67.4093, 0, Début) (intend, 2, 2, -65.8712, 0, Début) (seen, 2, 2, -41.7503, 0, Début)	*(since, 1, 1, -29.419, 0, Début*) (see, 1, 1, -40.2709, 0, Début) (seen, 1, 1, -38.7929, 0, Début) (noticed, 1, 1, -41.1871, 0, Début) (saw, 1, 1, -41.4us, 0, Début)
Nous	Avons	Vu

Figure 23: Le format de la matrice (mot, couverture, fertilité, score, position précédent, mot précédent), l'accès à une hypothèse se fait par (mot, couverture, ligne, colonne).

Une fois la table complètement remplie, on cherche l'hypothèse de score maximal qui couvre l'ensemble des positions sources (3 mots dans notre cas). Cette hypothèse est décrite par l'item (seen, 3, 1, -28.5164, 1, have). Pour obtenir la solution, il faut alors reculer (*backtraking*). On cherche l'antécédent qui est (have, 2, 1, -18.2373, 0, we) puis la

précédente (we, 3, 3, -49.0835, 0, Début) qui est le premier mot parce que cet item a un antécédent (0,Début), alors on s'arrête et la traduction est construite.

Si on ne prend pas en compte le meilleur score dans la matrice et on cherche les meilleures traductions de différentes longueurs telle qu'elles sont couvrantes tous les mots sources on obtient :

Longueur 1:

Dans la ligne 1 de la matrice, on trouve cette hypothèse, ***(we, 3, 3, -49.0835, 0, Début),*** a le meilleur score et une couverture 3, c'est-à-dire elle couvre tous les mots sources, on la considère comme une solution et le *backtracking* s'arrête au mot *Début.*

Alors traduction est d'un seul mot : *we*

Longueur 2:

Dans la ligne 2*,* ***(saw, 3, 2, -31.6374, 0, we)*** est la meilleure hypothèse. Elle vient de l'item identifié par *(mot= we, couverture= 1, ligne= 0, colonne= 0)* qui est au *Début*.

La traduction est de deux mots : *we saw*

Longueur 3:

Dans la ligne 3, ***(seen, 3, 1, -28.5164, 1, have)*** est la meilleure hypothèse. Elle vient de l'item identifié par **(have, 2, 1, 1),** en reculant (*backtracking),* on arrive à celui-ci ***(have, 2, 1, -18.2373, 0, we),*** et à son tour, ce dernier vient de celui qui est identifié par **(we, 1,0,0).** On trouve cet item ***(we, 1, 1, -11.458, 0, Début)*** à la fin de *backtracking.*

Alors la traduction est de trois mots : *we have seen*

On note que la traduction se construit de la fin de la phrase jusqu'au début.

Longueur 4:

De la même manière, ***(the 3, 0, -44.0328, 2, seen) -> (seen, 3, 1, -28.5164, 1, have)-> (have, 2, 1, -18.2373, 0, we)-> (we, 1, 1, -11.458, 0, Début)***

La traduction est de quatre mots *: we have seen the*

Longueur 5:

(the 3, 0, -58.9385, 2, in) ->(in, 3, 0, -47.2669, 2, seen)-> (seen, 3, 1, -28.5164, 1, have)-> (have, 2, 1, -18.2373, 0, we)-> (we, 1, 1, -11.458, 0, Début)

La traduction est de cinq mots : *we have seen in the*

Exemple de traduction et comparaison après et avant le filtrage:

1-Chaque mot français a 50 traductions en anglais (N=50) et un seuil BETA=10^6.

	Phrase source	Traduction	Temps (sec.)	VCB actifs
1	nous	We	~0	50
2	nous avons	we have	1	84
3	nous avons vu	we have seen	2	116
4	nous avons vu des	we have seen some	7	150
5	nous avons vu des solutions	we have seen some solutions	23	195
6	nous avons vu des bonnes solutions	we have seen some positives solutions	66	239
7	nous avons vu des bonnes solutions.	we have seen some positive alternatives .	191	263
8	nous avons vu des bonnes solutions avec vous	we have seen some positive solutions with you	312	288
9	nous avons vu des bonnes solutions avec vous .	we have seen some positive alternatives with you.	619	335

Tableau 11: Les résultats de décodeur sans filtrage strict (N=50; BETA=10^6)

2-Chaque mot français a 15 traductions en anglais (N=15) et un seuil BETA=1.9.

	Phrase source	Traduction	Temps (Sec.)	VCB actifs
1	nous	We	~0	15
2	nous avons	we have	~0	24
3	nous avons vu	we have seen	~0	33
4	nous avons vu des	we have seen some	~0	43
5	nous avons vu des solutions	we have seen some solutions	1	58
6	nous avons vu des bonnes solutions	we have seen some ideas	4	73
7	nous avons vu des bonnes solutions.	we have seen some positive alternatives.	9	88
8	nous avons vu des bonnes solutions avec vous	we have seen some positive solutions with you	15	93
9	nous avons vu des bonnes solutions avec vous.	we have seen some positive alternatives with you.	25	106

Tableau 12: Les résultats de décodeur avec un filtrage sur les nombres de mots anglais associés à chaque mot français (N=15) et un seuil (BETA=1.9) sur les hypothèses.

3-Chaque mot français a 7 traductions en anglais (N=7) et un seuil BETA=1.25.

	Phrase source	Traduction	Temps (Sec.)	VCB actifs
1	nous	We	0	7
2	nous avons	we have	0	12
3	nous avons vu	we have seen	0	18
4	nous avons vu des	we have seen	0	24
5	nous avons vu des solutions	we have seen some solutions	1	31
6	nous avons vu des bonnes solutions	we have seen some kind	1	38
7	nous avons vu des bonnes solutions.	we have seen some good answers.	2	45
8	nous avons vu des bonnes solutions avec vous	we have seen some good answers with you	5	49
9	nous avons vu des bonnes solutions avec vous.	we have seen some good answers with you.	8	55

Tableau 13: Les résultats de décodeur avec un filtrage sur les nombres de mots anglais associés à chaque mot français (N=7) et un seuil (BETA=1.25) sur les hypothèses.

Regardons les tableaux et comparons les résultats. Au niveau du temps et de nombres de vocabulaires; les temps ont été diminués d'une façon remarquable.

1- Pour la phrase de 9 mots, le temps de traduction est réduit de 619 secondes à 25 secondes pour la même réponse et à 8 secondes pour une autre réponse qui est aussi considérée comme une traduction acceptable.
2- Pour 5 mots français, le temps est réduit de 23 secondes à une seconde pour les mêmes réponses.
3- La phrase de 4 mots (*nous avons vu des*), dans les deux premiers tableaux, a été traduite par (*we have seen some*), cependant elle est traduite par (*we have seen*) avec les contraintes strictes dans le troisième tableau, c'est un exemple de perte de certaines hypothèses importantes. En examinant la matrice de toutes les hypothèses, on trouve que ce résultat est favorisé parce qu'il a un score de -37.7094, par contre (*we have seen some*) a un score -37.8372.

5.1.7. Exemples de résultats obtenus

Les phrases sont prises de HANSARD, corpus test, c'est-à-dire l'entraînement n'a pas été fait sur ces phrases. On peut avoir une traduction parfaite (phrase 1) et d'autres acceptables (phrases 2, 3 et 4) et un peu loin (dernière phrase).

Phrase	mots	Les phrases sources et les traductions
Source	5	**le jeudi 17 avril 1986**
Décodeur	6	thursday , april 17 , 1986
Humain	6	*thursday , april 17 , 1986*
Source	7	**la charte canadienne des droits et libertés**
Décodeur	7	the Canadian charter of rights and freedoms
Humain	6	*Canadian charter of rights and freedoms*
Source	12	**m. nunziata : monsieur le président , j' invoque le règlement .**
Décodeur	12	mr. nunziata : mr. speaker , i rise on a settlement .
Humain	12	*mr. nunziata : mr. speaker , on a point of order .*
Source	17	**les pétitionnaires demandent que la loi canadienne sur la santé soit inscrite dans la constitution canadienne .**
Décodeur	16	the petitioners ask that the canadian act of health be put into the canadian constitution .
Humain	16	*these petitioners ask that the canada health act be enshrined in the constitution of canada .*
Source	20	**ils doivent engager des frais importants pour assister et participer aux audiences de l' office national de l' énergie .**
Décodeur	16	they should hire some substantial costs to participate and assist people in the national energy .
Humain	17	*they are faced with substantial costs to attend and to participate in national energy board hearings .*
Source	19	**je n' ai pas l' intention de faire une longue déclaration , mais je voudrais faire valoir quelques points .**
Décodeur	17	I do not have the intention to make a long statement and i would points .
Humain	19	*i do not want to make a long statement but i would like to make a few points .*
Source	17	**la douleur doit être encore plus vive lorsque l' enfant a été victime d'un meurtre.**
Décodeur	16	the pain be even more intense when the child has been a victim of murder .
Humain	19	*the pain these parents feel is even greater knowing they have lost a child as a murder victim .*
Source	19	**étant moi-même mère , je peux imaginer à quel point cela doit être dur de perdre un enfant .**
Décodeur	15	I , as my homeland cannot conceive how be in a baby out tough .
Humain	16	*as a parent I can imagine how difficult it would be to lose a child .*

Tableau 14: des exemples de corpus test de Hansard et une comparaison avec la traduction humaine. Un filtrage a été appliqué N=10, BETA=1.5.

5.2. L'algorithme de recherche "Greedy".

5.2.1. Principe

L'idée de l'algorithme "*greedy*" est d'appliquer un ensemble d'opérations en vu d'améliorer, au sens des modèles, une traduction initiale. L'idée a été proposée par [Germann U. et al, 2001] dans le cadre d'un modèle IBM4. Cette idée sacrifie l'exhaustivité de l'exploration faite par l'algorithme DP avec un pari que la solution trouvée ne s'éloignera pas trop de la solution optimale (au sens des modèles).

On peut argumenter que cet algorithme de recherche est plus souple à adapter à un nouveau modèle que les formulations par programmation dynamique. Il suffit en effet de redéfinir l'ensemble des opérations à considérer.

L'algorithme greedy tente de manière systématique toutes les opérations partout où elles s'appliquent et retient l'opération et son emplacement qui améliorent plus la probabilité $P(e|f)$. On itère alors le processus sur la nouvelle solution jusqu'à ce qu'aucune application d'une opération n'améliore la solution courante.

Plus formellement,

Soit f la phrase source, e sa traduction initiale et a l'alignement associé.

Fonction A

$\forall$ o $\in$ **O,** l'ensemble des opérations valides

$\forall$ c, le contexte d'application de o.

Soit (e',a')=l'application de $(o,c,(e,a))$.

Si $P(e',a'|f)>P(e,a|f)$.

$Max\_o=o$

$Max\_c=c$

$Max\_s=P(e',a'|f)$

Fin de A.

Initialisation :

e' est la première solution.

a' est l'alignement entre f et e'.

Do

a ← *a'* , *e* ← *e'*

Max_S =P(e,a|f)

Fonction A.

While *(Max_s !=P(e,a|f))*

e' **est la traduction de** *f* **avec la probabilité** *Max_s***.**

Dans notre cas, *P(e,a|f)* est calculé à l'aide d'un modèle de traduction IBM2 et un modèle trigramme (les même modèles que nous avons utilisés dans l'algorithme DP.)

Outre sa rapidité de convergence (vers une solution locale), l'algorithme greedy est peu consommateur de mémoire (en comparaison avec l'algorithme précédent). En contre-partie, il est très possible que la solution trouvée par l'algorithme ne soit pas la solution optimale au sens des modèles mis en jeu.

5.2.2. Description

Les hypothèses sont stockées dans une table de deux dimensions (*I×4 cases*) où I est la longueur de la phrase française et 4 est le nombre de paramètres de chacune des hypothèses (*mot, position du mot source, fertilité, score*).

On initialise l'algorithme avec une traduction simplifiée où chaque mot français f_i est aligné avec le mot anglais e_i le plus probable au sens du modèle de transfert $e_i = \arg\max_e P(e \mid f_i) \ \forall i$ (les valeurs de $P(e_i|f_i)$ sont obtenues par l'application de Bayes).

Une fois l'alignement initial créé, le décodeur tente de l'améliorer en cherchant l'alignement le plus probable par l'application d'opérations parmi l'ensemble : *substitution, insertion, permutation ou suppression*. L'algorithme itère ces opérations sur toutes les hypothèses de l'alignement actuel. À chaque itération, le décodeur choisit le meilleur

alignement jusqu'à ce qu'aucune opération ne puisse être appliquée avec un gain. Ces opérations que nous décrivions dans la section suivante, ont été choisies pour deux raisons :

Elles sont tout d'abord peu coûteuses au niveau du temps de calcul, elles possèdent de plus la propriété souhaitée de perturber de manière non triviale les alignements considérés, ce qui offre à l'algorithme d'étudier divers types d'alignements.

À la limite, l'algorithme DP que nous avons vu possède en pratique ce problème car pour réduire les temps de calculs, nous filtrons une portion non négligeable de l'espace de recherche.

5.2.3. Implémentation des opérations

Nous utilisons les logarithmes (à base deux) pour remplacer les probabilités du score IBM2 (voir équation 5.7) et nous faisons des factorisations afin de simplifier les calculs et éviter certains termes à chaque modification d'alignement.

$$Score = P(J \mid I)\prod_{i=1}^{I}[P(e_i \mid e_{i-1}e_{i-2})P(i \mid j,J,I)P(f_j \mid e_i)] \qquad \textbf{(5.10)}$$

La première opération que nous avons considérée est la substitution d'un mot par un autre. Cette opération peut se produire à tout endroit dans la traduction en cours. Nous limitons la nature des mots "remplaçables" à ceux qui appartiennent au vocabulaire actif (section 5.1.5) calculé pour chaque mot.

Substituer()

Pour toutes les positions cibles ***i=1,…,I***

Pour tous les mots e, les traductions du mot f_i (parmi le vocabulaire actif).

Substituer le mot e_i *par e.*
Calculer le score de la phrase.
Garder le meilleur score, la position et le mot e correspondants.

Retourner le score, la position et le mot e.

Comme nous l'avons déjà vu, le modèle 2 introduit une probabilité d'alignement *P(i|j,J,I)* qui est la probabilité qu'un mot anglais en position *i* soit associé à un mot français en position *j*.

La fonction permuter() nous permet de permuter deux mots anglais en positions *i* et k dans la traduction. La probabilité d'alignement *P(i|j,J,I)* de l'équation (5.10) aide à déterminer la position optimale (au sens du modèle) du mot e dans la traduction.

Permuter()

Pour toutes les positions cibles ***i=1,2,…,I-1***
  Pour toutes les positions cibles ***k=i+1,…I***
    Permuter les mots e_i *et* e_k
    Calculer le score de la phrase.
    Garder le meilleur score, les positions correspondantes i et k.
Retourner le score et les positions i et k.

Le modèle 2 ne gère pas la notion de fertilité, mais il recèle indirectement une indication de cette fertilité: le nombre de mots connectés à un mot est un indice de sa fertilité. La fonction suivante essaie les différentes fertilités possibles (1,2,… *Max_f*) pour les mots anglais dans la phrase proposée comme solution.

Autoriser fertilité()

Pour toutes les positions cibles i=1,…,I-1

  Pour tous les fertilités f=1,..Max_f

      Calculer le score de la phrase.
      Garder le meilleur score, la position et la fertilité correspondante.
Retourner le score, la position et la fertilité du meilleur alignement.

Il arrive qu'un mot anglais n'ait pas d'équivalent en français. Nous appelons ces mots "mots spurious" ([Brown et al, 1993]). Nous proposons donc une fonction qui tente d'insérer un mot anglais (parmi les vocabulaires actifs) après chaque mot dans la phrase anglaise proposée.

Insérer spurious();

Pour toutes les positions cibles ***i=1,2,…,I***
  Pour tous les mots e des vocabulaires actifs

      Insérer le mot e dans la position i+1;
      Calculer le score de la phrase;
      Garder le meilleur score, la position et le mot e correspondants;
Retourner le score, la position et le mot e;

5.2.4. Exemple

Nous illustrons sur quelques exemples le déroulement de l'algorithme (les itérations et les opérations).

1-Phrase source : nous avons vu des résultats.

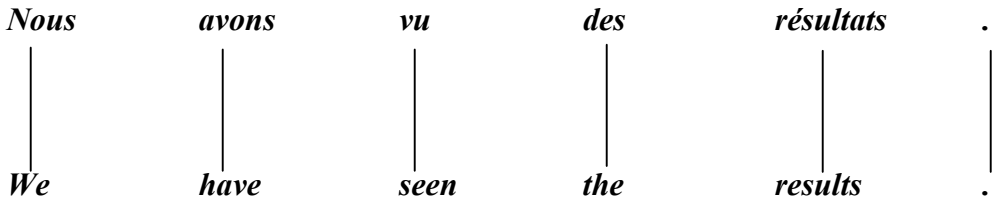

Figure 24: La traduction initiale par alignement un à un des mots à leur traduction la plus probable selon le modèle de transfert.

Dans cet exemple, la solution est atteinte dès l'initialisation : aucune opération n'améliore la traduction.

2- Phrase source: nous avons vu des outils importants.

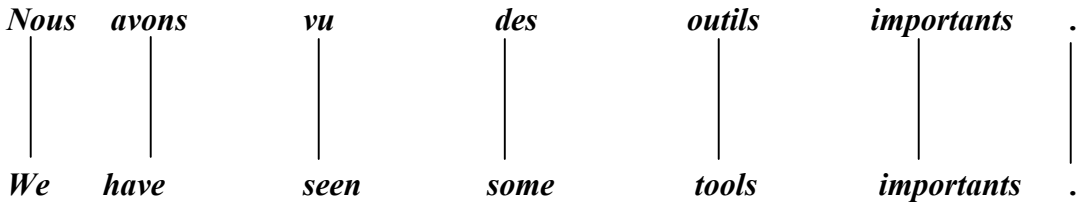

Figure 25: L'initialisation.

Le score de l'alignement initial est -68.1119. L'algorithme tente d'appliquer les opérations possibles pour trouver le meilleur alignement. Deux opérations améliorent ce score : substituer (the par some) pour un score –65.217 et permuter (les positions 5 et 6) pour un score de -64.681, la deuxième opération est donc choisie.

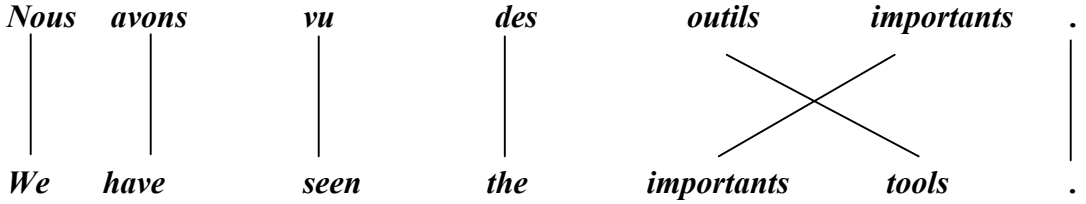

Figure 26: Itération 1, une permutation à la position 5 et 6.

À la deuxième itération, seule la substitution (de the par some) améliore le score de l'étape 1.

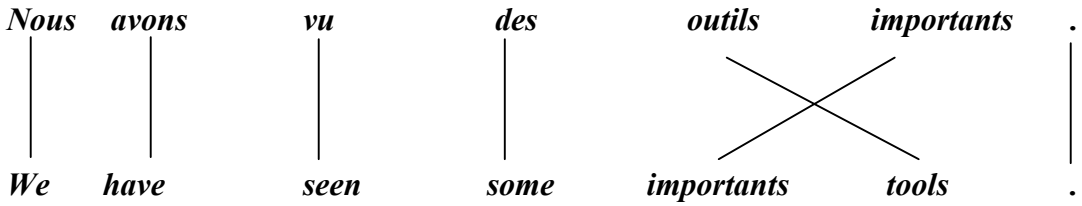

Figure 27: Itération 2, une substitution à la positin 4.

À l'étape 3, aucune autre transformation n'apporte de meilleur score. La traduction est donc celle résultant de l'étape précédente.

3-Phrase source: nous avons vu des solutions remarquables avec d'autres mesures.

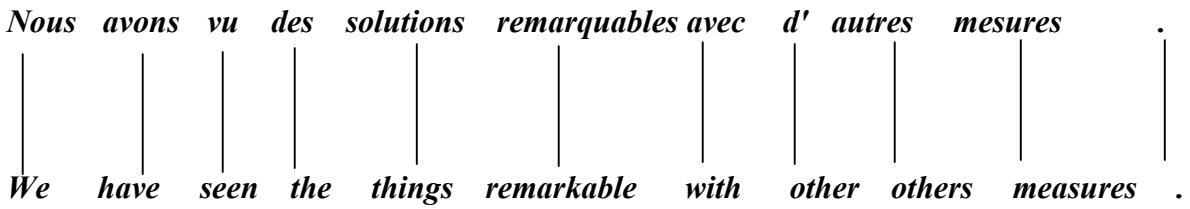

Figure 28: À l'initialisation, le score d'alignement est -133.518.

Les opérations suivantes améliorent l'alignement : Autoriser fertilité 2 pour le mot *other* avec un score de -115.237, Substituer(*the* , *some*) avec un score -129.697 et Permuter(les mots anglais des positions 5 et 6) avec un score –128.362. L'algorithme choisi donc la première des trois.

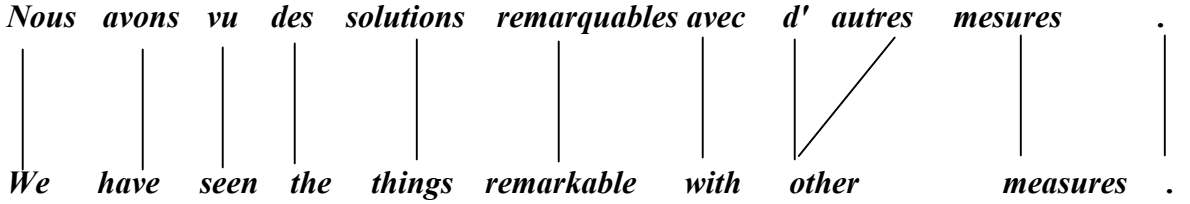

Figure 29: Itération 1: Le mot (*other*) à la position 8 est aligné avec 2 mots (*d' autres*) (fertilité 2);

À l'itération 2, la substitution du mot *the* par *some* amène au meilleur alignement.

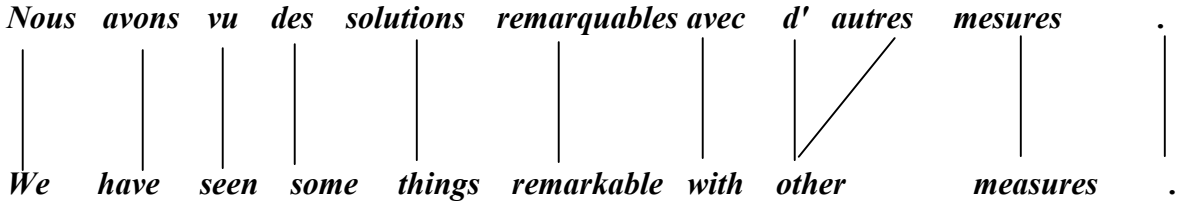

Figure 30: Itération 2: Substitution du mot (*the*) à la position 4 par (*some).*

À l'itération 3, le meilleur alignement trouvé est d'appliquer une permutation sur les positions 5 et 6.

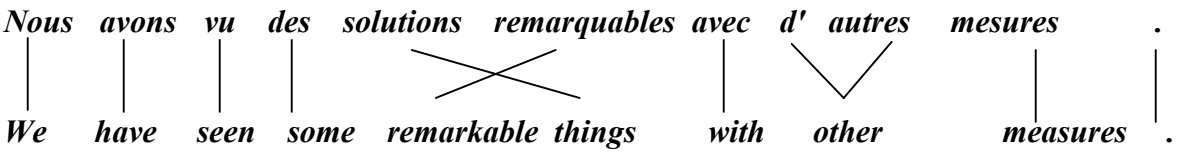

Figure 31: Itération 3: Permutation des mots à la position 5 et 6.

Enfin, la dernière itération amène à la substitution de *things* par *solutions*.

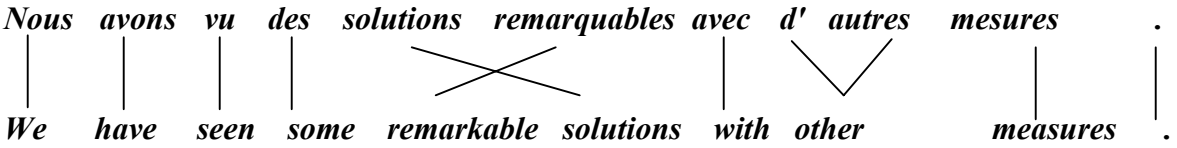

Figure 32: Itération 4: Substitution du mot (*things*) à la position 6 par (*solutions*).

5.2.5. Le nombre d'itérations et les temps de traduction.

Nous avons utilisé notre algorithme pour traduire 2376 phrases dont la longueur n'excédait pas 30 mots (français). La figure 33 montre le nombre d'itérations (moyen) effectué en fonction du nombre de mots de la phrase à traduire. L'accroissant du nombre d'itérations suit une courbe linéaire.

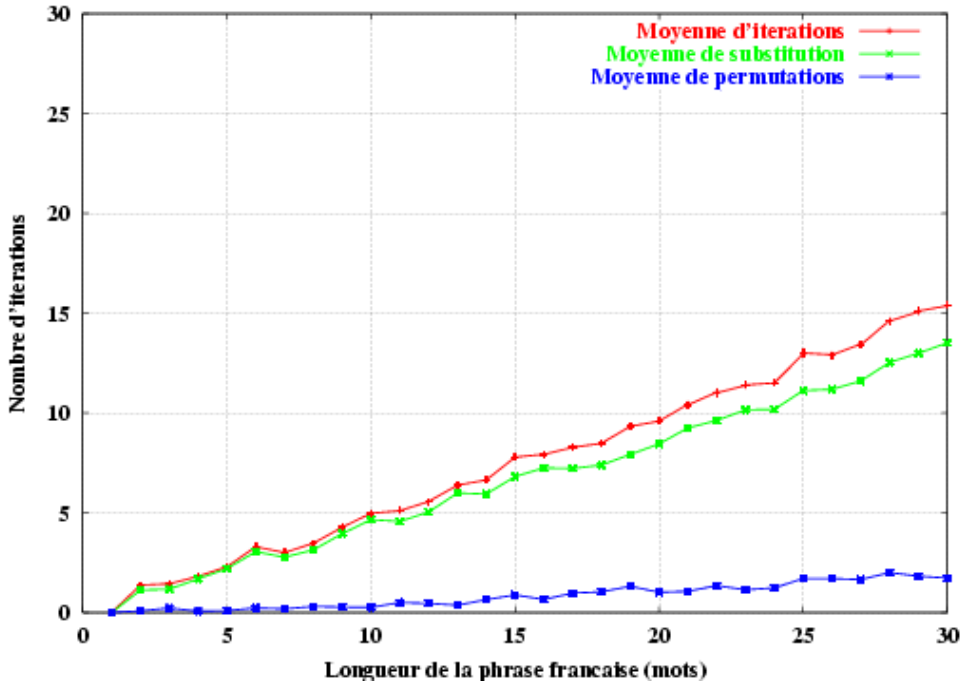

Figure 33: La moyenne d'itérations, le nombre de substitutions et permutations augmentent linéairement avec la longueur de la phrase à traduire.

On observe, d'après la figure 33 que la plupart des itérations sont des substitutions (72%), viennent au deuxième rang les permutations (19%); cependant les deux autres opérations s'appliquent moins souvent (6% pour l'opération de la fertilité et 3% pour l'insertion). Par exemple, une phrase française de dix mots est traitée par notre algorithme en 5 itérations (en moyenne). 4 de ces itérations font intervenir une substitution. (figure 34)

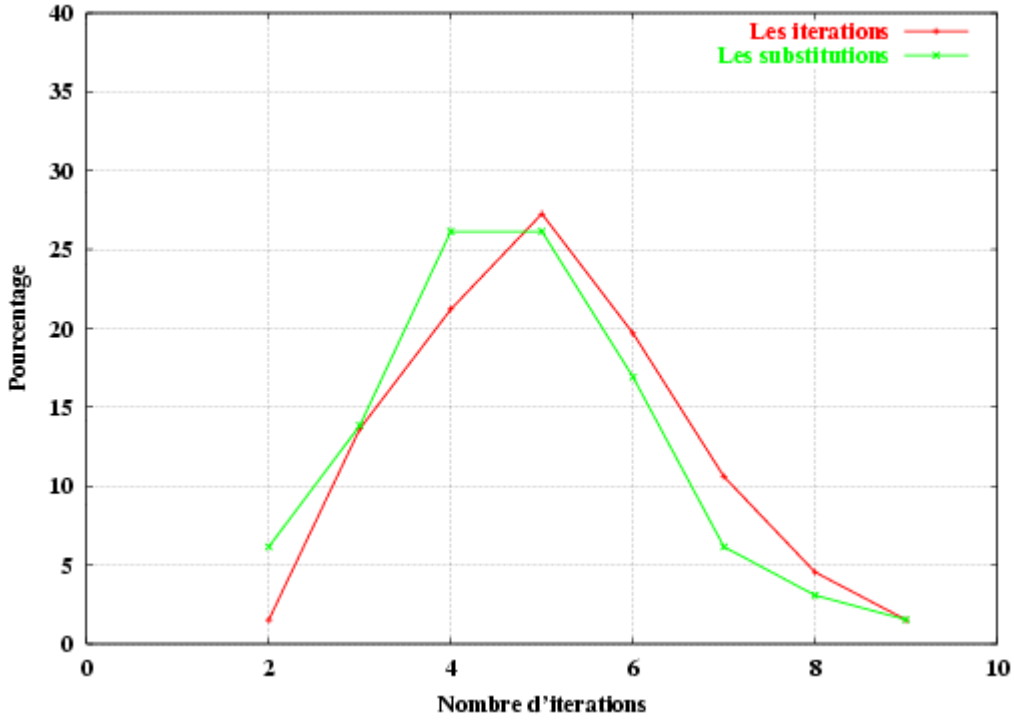

Figure 34: La distribution du nombre d'itérations et de substitutions.

L'expérience sur un sous-ensemble du corpus précédent de 66 phrases (figure 34) dont les phrases françaises sont de 10 mots, montre que la moyenne du nombre d'itérations est 5, que l'algorithme itère au moins deux fois et que le nombre maximal d'itérations est 9. Pour la substitution, l'opération la plus fréquente, est appliquée au moins deux fois.

L'entraînement du modèle 3 qui intègre la notion de la fertilité montre que 90% des mots anglais ont de fertilité 1 alors ceci explique que les opérations de fertilité et d'insertion de mots spurious s'appliquent rarement (9%). D'autre part, la position de l'adjectif en français est après le nom cependant en anglais c'est l'inverse. De ce fait, on comprend pourquoi la permutation est assez fréquente. Le fait que chaque mot anglais

possède différentes traductions explique de manière naturelle que les substitutions soient nombreuses.

D'après l'expérience sur le même corpus de 2376 phrases dont les phrases françaises sont constituées d'au plus 30 mots, on observe que la traduction des phrases de longueur d'au plus 10 mots sources prend au plus une seconde; les phrases constituées de 10 à 20 mots prennent entre une et deux secondes et les phrases dont les longueurs plus de 20 mots prennent au plus 4 secondes. Ces temps sont inférieurs à ceux de l'algorithme DP décrits dans la section (5.1). À titre d'exemple, traduire une phrase de 10 mots prenait avec cet algorithme environ 9 secondes. (la différence sur des phrases plus longues serait encore plus parlante).

5.2.6. Exemples de résultats obtenus

Nous reportons ici quelques exemples de traductions produites par notre algorithme pour des phrases du Hansard non présentes dans le corpus d'entraînement.

Phrase	mots	Les phrases sources et les traductions
Source	5	**Le jeudi 17 avril 1986**
Décodeur	5	On february 17 april 1986
Humain	6	*thursday , april 17 , 1986*
Source	7	**La charte canadienne des droits et libertés**
Décodeur	7	the canadian charter of rights and freedoms
Humain	6	*Canadian charter of rights and freedoms*
Source	12	**m. nunziata : monsieur le président , j' invoque le règlement .**
Décodeur	12	mr. nunziata : mr. speaker , i rise to of order .
Humain	12	*mr. nunziata : mr. speaker , on a point of order .*
Source	17	**les pétitionnaires demandent que la loi canadienne sur la santé soit inscrite dans la constitution canadienne .**
Décodeur	17	the petitioners ask that the bill on canadian health to be placed in the canadian constitution .
Humain	16	*these petitioners ask that the canada health act be enshrined in the constitution of canada .*
Source	20	**ils doivent engager des frais importants pour assister et participer aux audiences de l' office national de l' énergie .**
Décodeur	20	they must commit some important tuition to attend and participate the hearings of the board of national the energy .
Humain	17	*they are faced with substantial costs to attend and to participate in national energy board hearings .*
Source	19	**je n' ai pas l' intention de faire une longue déclaration , mais je voudrais faire valoir quelques points .**

Décodeur	20	i have listened not the intention of doing a long statement , but i would make points few points .
Humain	19	*i do not want to make a long statement but i would like to make a few points .*
Source	19	**cependant , en tant que membre de ce comité , je ne m'opposerais pas à réexaminer cet usage .**
Décodeur	20	however , in both the member of this committee , i cannot convince me not to review this practice .
Humain	17	*as one member of the committee i would certainly not object to revisiting that particular rule .*
Source	17	**La douleur doit être encore plus vive lorsque l' enfant a été victime d'un meurtre.**
Décodeur	17	the pain will be even more intense when the child has been victims of a murder .
Humain	19	*the pain these parents feel is even greater knowing they have lost a child as a murder victim .*
Source	19	**étant moi-même mère , je peux imaginer à quel point cela doit être dur de perdre un enfant .**
Décodeur	19	having myself mother , i can imagine at what point that must be tough to lose a child .
Humain	16	*as a parent I can imagine how difficult it would be to lose a child .*

Tableau 15: Exemples de traduction, extraits d'un corpus test (N=10). Humain est la traduction produite par un traducteur.

5.3. Greedy initialisé par la traduction produite par DP

On a vu dans les sections précédentes que le décodeur DP est lent mais qu'il parcourt une portion importante de l'espace de recherche. Néanmoins, il y a des filtres pour rendre les temps de réponse « acceptables ». Nous avons voulu voir si le greedy ne pouvait pas éventuellement trouver une solution meilleure que celle de DP en étant initialisée par DP. Dans ce but, on propose dans cet algorithme que la solution initiale du greedy soit la solution obtenue par le décodeur DP.

Les paramètres des hypothèses de l'algorithme DP (la fertilité, la position source) sont réutilisés pour le décodeur greedy et les mêmes opérations sont appliquées sur les résultats de DP. Nous appelons cette variante de l'algorithme greedy+.

L'expérience lancée sur un corpus de 2376 phrases a pris 218 secondes pour tout le corpus. Une partie de corpus de 403 phrases (~ 17%), aucune itération n'a été possible et la plupart de phrases non modifiées sont les phrases courtes (longueur inférieure à 10 mots).

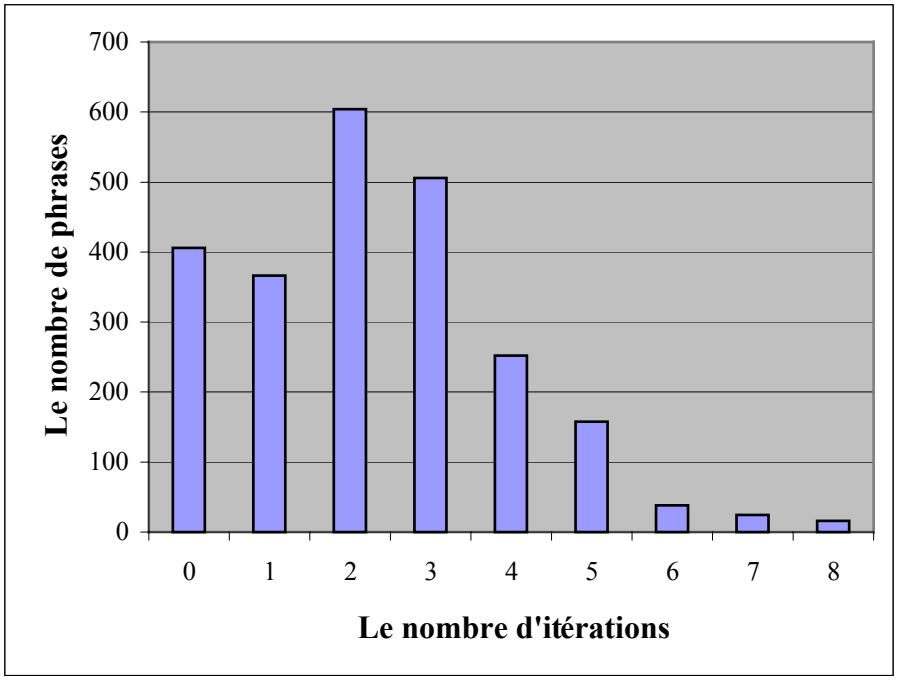

Figure 35: Les nombres de phrases itérées par greedy+.

La figure 35 montre que 62% des alignements optimaux sont atteints après au plus 3 itérations. Pour 16 phrases seulement, l'algorithme produit une solution après 8 itérations. Les opérations appliquées sont des substitutions.

Les critères de filtrage ne permettent donc pas d'atteindre toujours l'optimum au sens des modèles (dans 83% des cas). Donc on filtre trop. Nous discutons d'une façon détaillée les performances des décodeurs et nous présentons la perte de qualité de la traduction provoquée par le filtrage dans le chapitre 6 mais nous présentons dans la section suivante des exemples pour argumenter les résultats.

5.3.1. Exemples de résultats obtenus

On prend quelques exemples pour commenter les résultats.

1)

Source : le jeudi 17 avril 1986

DP : thursday , april 17 , 1986

Greedy+: *thursday , april 17 , 1986*

Humain : thursday , april 17 , 1986

Dans cet exemple, on remarque que le décodeur greedy+ ne peut pas améliorer la traduction de DP alors il n'y a pas de perte au sens des modèles causée par le filtrage.

2)

Source : adoption des motions portant présentation et première lecture .

DP : concurrence motions for introduction and first reading .

Greedy+: *concurrence motions introduction and first reading* .

Humain : ***motions for introduction and first reading deemed adopted .***

Le décodeur greedy+ donne une fertilité 2 à *motions* (élimination du mot *for*) de la traduction produite par DP. Dans cet exemple, la traduction produite par greedy+ éloigne de la traduction humaine (en terme de nombre des mots communs entre la référence et la traduction) mais s'améliore au sens des modèles.

3)

Source : je dépose aujourd'hui une autre pétition qui porte des centaines de signatures.

DP : i now table another petition which concerns hundred of signatures .

Greedy+: *i present another petition which concerns hundred of signatures* .

Humaine:***i would like to introduce another with several hundred signatures today .***

Les évaluateurs humains trouvent que la traduction produite par DP est meilleure que celle produite par greedy+. Cependant au sens des modèles, la traduction de greedy+ est la meilleure.

4)

Source : il ne désigne pas nécessairement un homme .

DP : he does not necessarily signal a man .

Greedy+: *it does not necessarily signal a man* .

Humaine: ***it does not refer to a man .***

Dans cet exemple, geedy+ améliore la traduction au sens des modèles d'autant plus que cette traduction au point de vu humain est la meilleure. Ainsi le filtrage du DP avait de mauvais effet sur la traduction.

Nous l'avons vu sur ces exemples, chaque algorithme propose sa traduction. Il arrive que les traductions soient les mêmes et peut-être non, dans ce dernier cas, il n'est pas facile de choisir la meilleure traduction : un meilleur score d'alignement n'est pas nécessairement garant d'une meilleure traduction (ceci traduit les faiblesses des modèles sous-jacents utilisés). Nous aborderons dans le chapitre suivant les problèmes d'évaluation.

Chapitre 6

Évaluation des résultats

Les algorithmes de recherche constituent une partie cruciale de la traduction automatique probabiliste. Leur performance affecte directement la qualité de la traduction. Sans un décodeur fiable et efficace, un système de traduction automatique statistique peut manquer la traduction d'une phrase source, même si elle était une phrase du corpus d'entraînement.

Il existe plusieurs questions que l'on doit étudier lorsqu'on réalise un décodeur :

1- Optimalité : L'algorithme de décodage peut-il trouver la traduction optimale au sens du modèle?
2- Rapidité : En combien de temps la traduction est-elle proposée?

Il est à noter qu'un bon décodeur, c'est-à-dire celui qui propose rapidement la traduction optimale au sens des modèles, n'est appréciable à l'utilisateur que dans la limite où les modèles sont bons. Nous étudions dans ce chapitre le comportement des décodeurs que nous avons implémentés.

L'évaluation humaine est une méthode pour déterminer la performance d'un système de traduction. Les évaluations humaines de la traduction automatique s'intéressent à plusieurs aspects de la traduction, comme l'adéquation, la fidélité, et la maîtrise de la traduction. L'évaluation humaine est très discutée, en témoigne le nombre des travaux sur le sujet ([White et al, 1994]). En fait, il y a plus de publications sur l'évaluation que sur la modélisation de la traduction.

Le problème majeur de l'évaluation humaine (nous faisons abstraction ici des nombreux protocoles d'évaluation proposés) est le temps qu'elle nécessite. Elle correspond donc d'avantage à une situation où l'on souhaite évaluer un système stable. En phase de développement, on souhaite cependant appliquer des protocoles plus rapides au sacrifice éventuel de leur précision. L'idée étant de vérifier rapidement la validité d'une hypothèse faite dans le processus de développement.

Dans ce travail, on utilise deux méthodes d'évaluation bien connues dans le monde de la traduction automatique WER et BLEU. Ces métriques sont indépendantes de paires de langues étudiées (dans la limite de l'existence de la notion de mots). L'idée est d'utiliser une ou plusieurs traductions de référence auxquelles on compare la traduction produite automatiquement.

L'acuité des ces métriques est très discutable, l'on peut argumenter que les valeurs moyennes, mesurées sur de gros corpus de tests, sont indicateurs suffisants pour valider comparativement plusieurs approches.

Nous décrivons l'essence de ces deux métriques et les appliquons pour évaluer les décodeurs décrits dans le chapitre précédent. Nous contrastons de plus les valeurs observées par une évaluation humaine d'un sous-ensemble de traduction produite par chaque système.

6.1. WER et SER

WER et SER sont deux métriques souvent utilisées pour l'évaluation de traductions automatiques. On évalue la performance en terme de taux d'erreur mesurés au niveau de la phrase (***SER Sentence Error Rate***) et des mots (***WER Word Error Rate***).

Le premier taux (SER) mesure le pourcentage de phrases pour lesquelles la traduction n'était pas exactement celle de la référence. Cette méthode est sévère car une traduction peut-être bonne sans pour autant être identique à la référence. Considérer plusieurs traductions de référence permet de limiter jusqu'à un certain point le problème.

Le second taux (WER) est calculé par une distance de Levenstein qui comptabilise le nombre minimal d'opérations qu'il faut effectuer pour passer de la traduction produite à la traduction de référence. Les trois opérations considérées ici sont l'*insertion*, la *suppression* et la *substitution* qui reçoivent toutes le même poids.

Exemple:

Dans ces exemples les traductions sont du français vers l'anglais. On désigne par SRC la phrase source à traduire, REF la traduction de référence (humaine) et CAN la traduction candidate c'est-à-dire la traduction obtenue par le système de traduction que l'on souhaite évaluer automatiquement.

1) SRC : friday , march 15 , 2002

REF : le vendredi 15 mars 2002

CAN: le vendredi 15 mars 2002

insertion:0 deletion:0 substitution:0 exact:5 WER=0.00% ; SER=0.00%

Les deux phrases sont parfaitement identiques.

2) SRC: business of the house

REF: les travaux de la chambre

CAN: travaux de la chambre

insertion:0 deletion:1 substitution:0 exact:4 WER=20.00%; SER=100.00%

Élimination de *les*.

3) SRC: it is clearly wrong .

REF: c' est clairement répréhensible .

CAN: ce qui est clairement .

insertion:1 deletion:1 substitution:1 exact:3 WER=50.00%; SER=100.00%

Il faut insérer le mot *qui* dans REF, substituer *c'* par *ce* et supprimer *répréhensible*

4) SRC: canada

REF: canada

CAN: les canadiens

insertion: 1 deletion: 0 substitution: 1 exact: 0 WER=100.00%; SER=100.00%

On doit substituer le mot *canada* par *les* dans REF et insérer *canadiens*.

On remarque que SER est trop sévère parce que cette métrique accorde un taux d'erreur aux phrases qui ne sont parfaitement pas exactes (les exemples 2 et 3) cependant WER était moins sévère et elle accorde un taux d'erreur de 20% et 50% respectivement aux exemples précédents. De ce fait, WER est plus efficace que SER.

6.2. BLEU

[Papineni et al, 2001] ont présenté une méthode d'évaluation pour la traduction automatique (Bleu BiLingual Evaluation Understudy).

L'idée de Bleu est de comparer les phrases (traduction, référence) en se basant sur les séquences de mots (n-gram). Une traduction est d'autant meilleure qu'elle partage un grand nombre de n-gram avec une ou plusieurs traductions de référence.

Nous utilisons dans notre travail un package à la disposition de NIST, qui organise des compagnes d'évaluation sur la traduction automatique.

Exemple 1 :

Supposons que deux systèmes de traduction traduisent une telle phrase source par ces deux phrases anglaises Candidat1 et candidat2.

Candidat 1 : It is a guide to action which ensures that the military always obeys the commands of the party

Candidat 2: It is to insure the troops forever hearing the activity guidebook that party direct

Référence: It is a guide to action that ensures that the military will forever heed party commands

Pour le candidat 1 : La précision uni-gram = 14/18**.**
La précision bi-gram = 8/17.
…

Pour le candidat 2: La précision uni-gram = 8/14**.**
La précision bi-gram = 1/13.
…

Bleu donne un score entre 0 et 1 où 0 est le score des phrases complètement différentes des références.

[Papineni et al, 2001] ont montré que l'évaluation de BLEU est cohérente avec celle des évaluateurs humains.

6.3. Évaluations et comparaison des décodeurs

Nous mesurons la performance des algorithmes de recherche que nous avons décrits dans le chapitre précédent.

On utilise ici deux corpus de test. Chacun contient 1210 phrases françaises dont la longueur maximale est de 20 mots. L'un présent dans le corpus d'entraînement et l'autre est constitué de phrases qui n'ont jamais été vues à l'entraînement. Les deux corpus sont extraits des textes Hansard. Aucune stratégie particulière n'a été appliquée dans le but de sélectionner, dans le second corpus, des phrases proches des phrases vues à l'entraînement. Les mots inconnus, c'est-à-dire que le modèle de traduction n'a pas ces mots parmi ses vocabulaires, sont remplacés par UNK. (Le premier corpus contient 23 mots inconnus tandis que le second contient 197 mots inconnus)

Les métriques décrites dans les sections précédentes sont utilisées sur nos corpus et montrées dans la table 16. On observe que DP obtient la meilleure performance (pour les 3 scores).

Modèle	***Corpus***	***Nb de phrases***	***Bleu***	***WER***	***SER***	***Traductions parfaites***
DP	*test*	1210	0.19	58.7%	96.8%	39
	entraînement	1210	0.31	52.4%	90.1%	120
Greedy	*test*	1210	0.15	62.8%	98.1%	22
	entraînement	1210	0.25	54.8%	91.9%	98
Greedy+	*test*	1210	0.17	60.2%	97.2%	34
	entraînement	1210	0.28	53.5%	90.7%	112

Tableau 16: Les résultats de l'évaluation des décodeurs.

Nous en déduisons que DP, qui parcourt une partie de l'espace de recherche plus grande que la partie parcourue par greedy, est garant de meilleures traductions. Le décodeur greedy+ améliore les résultats au sens des modèles mais en pratique les traductions

s'éloignent des traductions humaines (la référence). De ce fait, les métriques automatiques notent une baisse de qualité par rapport à DP.

On remarque également sans surprise que les taux observés sur le corpus d'entraînement sont nettement supérieurs à ceux observés sur le corpus de test. Une raison qui explique les taux élevés d'erreur du corpus test est la présence de mots hors vocabulaire. De manière prévisible, la présence de mots inconnus a un impact direct sur les performances et en particulier, sur la couverture du vocabulaire actif à partir duquel les traductions sont construites. Et une autre raison est que les modèles utilisés ne sont pas parfaits.

Le décodeur DP a pris 34 216 secondes pour traduire le corpus de test (en moyenne 28 secondes par phrase), alors que greedy a traduit le corpus en 1574 secondes (en moyenne 1.3 secondes par phrase). Dans cette expérience, on a observé que le temps moyen de traduction d'une phrase de 20 mots avec le décodeur DP est 40 secondes (le temps accroît exponentiellement) cependant le temps avec le décodeur greedy ne dépasse pas 3 secondes (croissance linéaire). On observe une grande différence de temps et la raison majeure de cette marge de temps est le nombre d'alignements considérés.

Selon les métriques, il semble donc préférable d'utiliser l'algorithme DP, et c'est essentiel car les algorithmes considérés sont beaucoup plus nombreux que ceux que greedy observe. En revanche, il est clair que greedy est de l'air plus rapide. Nous avons donc voulu voir si la différence de qualité mise en évidence ici était vraiment confirmée par des évaluations humaines.

6.4. Évaluation humaine

Un corpus de test formé de 50 phrases, a été à cet effet constitué de 35 phrases non vues à l'entraînement prises au hasard du Hansard, et 15 autres phrases prises d'une toute autre source. On associe à ces paires de traductions la phrase traduite par un traducteur humain, donc supposée correcte (la référence.) Ces phrases sont présentées à plusieurs évaluateurs (6 personnes bilingues qui travaillent dans le domaine du traitement des langues), et les traductions automatiques sont présentées au hasard de façon à ce que les

évaluateurs n'en sachent pas la provenance. La phrase source est clairement indiquée ainsi que la traduction de référence.

Les évaluateurs comparent les traductions automatiques, et indiquent celle qu'ils préfèrent. Lors du calcul des scores, une préférence se traduit par un score de 1. Le score 0 indique que les systèmes sont équivalents ou bien qu'aucune préférence n'existe.

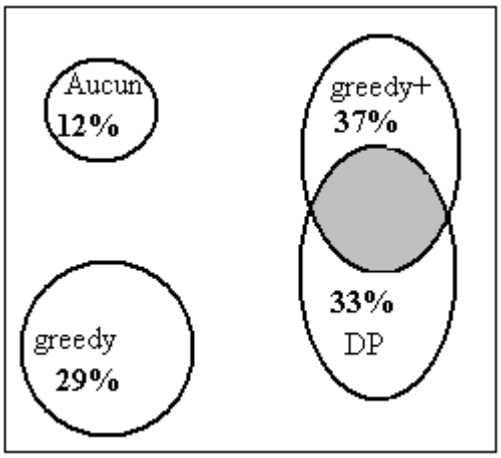

Figure 36: Les proportions d'acceptabilité de traduction des évaluateurs pour chaque décodeur.

Les évaluateurs n'ont pas trouvé une traduction acceptable dans 12% des cas, parmi les traductions proposées par les trois décodeurs (figure 36). Les traductions proposées par greedy ont été sélectionnées dans 29% des cas et les traductions des greedy+ et DP sont préférées dans 59%. 11% des évaluateurs ont jugé acceptables les traductions fournies par DP et greedy+ à la fois (i.e. le décodeur greedy+ n'a pas pu améliorer les traductions de 14 phrases ainsi les traductions de ces 14 phrases sont les communes entre DP et greedy+). 37% des évaluateurs ont trouvé que greedy+ est le meilleur décodeur parmi les trois algorithmes, cependant DP occupe le second rang par un pourcentage de 33%. (tableau 17)

Une contradiction entre l'évaluation automatique qui préfère DP et l'humaine qui préfère greedy+ est peut-être expliquée par le fait que les évaluations automatiques ont besoin de plusieurs références pour bien déterminer les préférences mais de toute façon DP et greedy+ avec les deux évaluations se sont proches l'un de l'autre.

Modèle	Personne1	Personne2	Personne3	Personne4	Personne5	Personne6	%
Greedy	8	15	17	18	17	13	**29%**
DP	8	14	15	11	9	8	**22%**
Greedy+	8	13	12	15	16	13	**26%**
DP et Greedy+	4	8	6	6	5	4	**11%**
Aucun	22	0	0	0	3	12	**12%**

Tableau 17: Les choix des évaluateurs et ses différents avis avec les pourcentages.

Le tableau 17 montre que les évaluateurs humains aussi n'ont pas le même jugement (dans 20% des cas, tous les évaluateurs sont d'accord, 22% des cas, cinq évaluateurs sont d'accord). Mais tous les évaluateurs préfèrent greedy+ (37%).

Toutes les phrases avec les évaluations humaines de ces expériences sont présentées dans l'annexe1.

6.5. Exemple des traductions évaluées

1)

REFERENCE : nous avons vu des résultats .
DP : we have seen the results .
Greedy+: we have seen some results .
Greedy : we have seen the results .

Tous les évaluateurs choisissent la traduction de greedy+.

2)

REFERENCE : il faut faire en sorte de modifier cette perception .

Greedy : we must make the kind of change that perception .

DP : there must be the kind of change that perception .

Greedy+: we must be the kind of change that perception .

Dans cet exemple, les évaluateurs ont plusieurs choix. (un ne trouve aucune traduction acceptable, un autre préfère celle de greedy+ , un choisi DP et d'autre greedy.)

3)

REFERENCE : les températures moyennes sont bien sûr plus tempérées .

Greedy+: canadians are obviously more temperate average temperatures .

DP: canadians are much more temperate course average temperatures .

Greedy: the temperatures averages are certainly well longer boreal .

(un pour aucune traduction, 2 greedy, un DP et un greedy+).

L'objectif premier de la traduction automatique est de comprendre et/ou de se faire comprendre. En conséquence, un critère efficace est l'opinion du lecteur : le document traduit est-il exploitable pour le besoin que l'utilisateur en a et retranscrit-il le contenu du texte de départ ? Est-ce que le document a été traduit rapidement ?

Pour un utilisateur qui s'intéresse à l'efficacité du traducteur automatique et il ignore le temps de traduction, nous lui suggérons le décodeur DP qui est le meilleur selon les évaluations humaines et automatiques parmi les deux décodeurs DP et greedy. Toutefois, l'utilisateur qui aimerait comprendre le sujet d'un document dans une autre langue rapidement, nous croyons que le décodeur greedy répond à ses besoins.

Chapitre 7

Conclusion

C'est aux abords des années 90, qu'une équipe de chercheurs d'IBM a proposé une approche statistique opérationnelle à la traduction automatique. Cette approche dénotait de celles qui étaient alors utilisées en traduction et qui consistaient essentiellement en l'écriture de règles et de lexiques structurés. L'approche statistique a depuis séduit de nombreux chercheurs du domaine.

Le problème de la traduction probabiliste est dual. Il convient en effet d'acquérir automatiquement les paramètres d'un modèle de traduction et de la langue dans laquelle ont traduit. C'est le problème de la modélisation que nous avons abordé dans la première partie de ce mémoire. Nous avons en particulier montré que le package GIZA mis au point par un groupe de travail de l'université de John Hopkins et disponible pour les universitaires répondait aux besoins de la modélisation. Le second problème consiste à trouver la meilleure traduction possible, étant donnés les modèles.

Le problème du décodage en traduction n'est pas simple et est en fait un problème NP-complet. Nous nous sommes donc intéressés dans une deuxième partie de ce mémoire à comparer deux techniques de décodages s'appuyant sur un modèle IBM2 que nous avons entraîné à l'aide du package GIZA. Ces deux techniques possèdent des caractéristiques à priori contraires. La première technique utilise la programmation dynamique (DP) pour factoriser des calculs répétitifs, et permet ainsi d'explorer une partie raisonnable (assez grande) de l'espace des traductions potentielles au prix cependant de temps de traitements non négligeables. La seconde technique (dite vorace), fait au contraire le pari qu'en perturbant de manière mineure un alignement de départ (que l'on peut obtenir de manière simple), on peut trouver une traduction pertinente (au sens du modèle) en ne parcourant qu'une petite partie de l'espace de recherche.

Nous avons évalué les forces et les faiblesses des deux décodeurs que nous avons implantés. Nous avons pour cela considéré deux approches. L'une consiste à comparer automatiquement les traductions produites à celle faite par un humain. Nous utilisons ici des mesures telles que la distance d'édition et d'autres plus spécifiques à la tâche de traduction. L'autre approche est plus lourde mais également plus fiable. Elle consiste en effet à demander à des humains de classer des traductions automatiques en ordre de préférence. Nous montrons entre autre que ces deux protocoles d'évaluation sont en faveur de l'approche DP. Cependant, les approches voraces proposent des traductions qui ne sont pas déméritantes et ce type de décodeur est en fait à favoriser dans toute situation où les temps de traductions sont un facteur à considérer (traduire une phrase d'une vingtaine de mots avec une approche DP peut prendre une quarantaine de secondes).

Ce travail a porté sur l'étude de la paire de langues français/anglais. Ces deux langues ont des structures proches, aussi aimerions-nous étudier la viabilité de nos décodeurs (et des modèles sous-jacents) à traduire des langues plus éloignées, comme par exemple l'anglais et l'arabe. De plus, il existe de nombreuses façons d'améliorer la qualité produite par une approche probabiliste. L'ajout de dictionnaires pour contraindre l'entraînement des modèles est un problème sur lequel nous aimerions travailler.

Bibliographie

BROWN P. F., PIETRA S. A. D., PIETRA V. J. D. & MERCER R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, **19**(2), 263–311.

Al-Onaizan Y, Curin J., Jahr M., Knight K., Lafferty J., Melamed D., Och F-J, Purdy D., Smith N. A., Yarowsky D. (1999), *Statistical Machine Translation: Final Report*, Johns Hopkins University 1999 Summer Workshop on Language Engineering, Center for Speech and Language Processing, Baltimore, MD.

Knight, K. 1999. *A Statistical Machine Translation Tutorial Workbook*. Tech. Rep.?, USC/ISI. (available at http://www.clsp.jhu.edu/ws/projects/mt/wkbk.rtf).

BROWN P. F., Coke J., PIETRA S. A. D., PIETRA V. J. D. Jelinek F., Lafferty J. Roosin P. S., MERCER R. L. (1993). *A Statistical Approach to Machine Translation.*

Nießen, S., Vogel, S., Ney, H., and Tillmann, C. (1998). A DP Based Search Algorithm for Statistical Machine Translation. In *Proceedings of COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 960–967, Montreal, Quebec, Canada, August.

Y. Wang and A. Waibel. 1997. *Decoding algorithm in statistical machine translation*. In *Proc ACL.*

Germann U., Jahr M., Knight K., Marcu D., and Yamada K., 2001. *Fast Decoding and Optimal Decoding for Machine Translation.*

Ismael Garcia and Fransisco Casacuberta. 2001. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proceedings of the 8th Machine Translation Summit,* pages 115–120, Santiago de Compostela, Galicia, Spain, September. IAMT.

Vogel, S., Franz, J. O., Tillman, C., Nießen S., Sawaf, H., and Ney H.,(2000) *Statistical Methods fo Machine Translation.*

Nießen, S., Och, F. J., Leusch, G., and Ney, H. (2000). An Evaluation Tool for Machine translation: Fast Evaluation for MT Research. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, 39–45, Athens, Greece, May.

Papineni K., Roukos S., Ward T., Zhu W., IBM T.J. Watson Research Center.(September 17, 2001) *BLEU: a Method for Automatic Evaluation of Machine Translation.*

C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga. 1997. A dp-based search using monotone alignments in statistical translation. In ACL-35 (ACL, 1997), pages 289–296.

Och, F. J. and Ney, H. (2000a). A comparison of alignment models for statistical machine translation. In *COLLING '00: The 18th Int. Conf. on Computational Linguistics*, pages 1086-1090, Saarbrücken, Germany.

Och, F. J. and Ney, H. (2000). *Statistical Machine Translation.*

White, J.S. et al. (1994). The ARPA MT evaluation methodologies: evolution, lessons, and future approaches. In: *Technology partnerships for crossing the language barrier: proceedings of the first conference of the Association for Machine Translation in the Americas,* 5-8 October 1994, Columbia, Maryland. [Washington, DC: AMTA, 1994], pp.193-205.

Philippe Langlais, Michel Simard, and Jean Véronis. 1998. Methods and practical issues in evaluating alignment techniques. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL),* Montréal, Canada, September.

Ph. Langlais. 1997. Alignement de corpus bilingues: intérêts, algorithmes et évaluations. In *Bulletin de Linguistique Appliquée et Générale, numéro Hors Série*, pages 245–254, Université de Franche-Comté, France, dec.

M. Simard and P. Plamondon. 1996. Bilingual sentence alignment: Balancing robustness and accuracy. In *Proceedings of the Second Conference of the Association for Machine Translation in the Americas (AMTA),* Montréal, Québec.

Van Slype, G. 1982. Conception d'une méthodologie générale d'évaluation de la traduction automatique. *Multilingua* 1(4): 221-237

Annexe1

On présente dans cette annexe les phrases avec les évaluations humaines comme l'exemple suivant.

1
REFERENCE: les comités de la chambre
GREEDY : the committees of the house
DP : the house committees
GREEDY+ : the reports house
REPONSES : 2 1 2 1 1 2

Dans cette phrase le premier évaluateur préfère le deuxième décodeur (DP), le deuxième évaluateur préfère le premier algorithme (GREEDY), ainsi de suite…

2
REFERENCE: nous avons vu des résultats .
GREEDY : we have seen the results .
DP : we have seen the results .
GREEDY+ : we have seen some results .
REPONSES : 3 3 3 3 3 1/2

3
REFERENCE: monsieur le président , je suis heureux de pouvoir présenter un projet de loi d' initiative parlementaire .
GREEDY : mr. speaker , i am pleased to introduce a piece of legislation a parliamentary initiative forward .
DP : mr. speaker , i am pleased to present a bill of private .
GREEDY+ : mr. speaker , i am pleased to introduce bill in private .
REPONSES : 0 2 1 3 1 1

4
REFERENCE: je dépose aujourd'hui une autre pétition qui porte des centaines de signatures .
GREEDY : i table today another petition which deals of hundreds of signatures .
DP : i now table another petition which concerns of hundred signatures .
GREEDY+ : i present another petition which concerns of hundred signatures .
REPONSES : 0 1 1 2 3 3

5
REFERENCE: le président : les autres questions restent elles au feuilleton ?
GREEDY : the chairman : the remaining questions stand they order paper ?
DP : the chairman : the remaining questions be allowed to stand ?
GREEDY+ : the chairman : the remaining questions be allowed to stand ?
REPONSES : 2/3 2/3 1 2/3 2/3 2/3

6
REFERENCE: cependant , en tant que membre de ce comité , je ne m' opposerais pas à réexaminer cet usage .
GREEDY : however , in both the member of this committee , i cannot convince me not to review this practice .
DP : however , in both the member of this committee , i am not opposed to review this practice .
GREEDY+ : however , in both the member of this committee i am not opposed to this custom review .
REPONSES : 3 3 3 3 3 3

7
REFERENCE: en toute honnêteté , je dois dire que certains progrès mineurs ont déjà été faits à cet égard .
GREEDY : in all honesty , i must say that some progress miners have already been made in this regard .
DP : in all honesty , i must say that some progress miners have been agreement .
GREEDY+ : in all honesty , i must say that some progress have been agreement .
REPONSES : 0 1 1 1 1 1

8
REFERENCE: des programmes comme ceux-là doivent habituellement compter sur du personnel bénévole .
GREEDY : some programs as people are usually rely on the personal bénévole .
DP : some programs as they usually rely on volunteers staff .
GREEDY+ : some programs as normal rely on bénévole staff .
REPONSES : 2 3 2 2 2 3

9
REFERENCE: il faut d' abord faire payer les criminels .
GREEDY : it must be first to pay the block-busting .
DP : first we need to pay block-busting .
GREEDY+ : first we have to pay block-busting .
REPONSES : 0 2/3 2/3 2/3 2/3 0

10
REFERENCE: les coûts de tout notre régime de justice pénale sont énormes .
GREEDY : the cost of everything our system of criminal justice is enormous .
DP : the cost of everything in our system of criminal justice is enormous .
GREEDY+ : the cost of everything in our system of justice is enormous .
REPONSES : 2 2 2 2 2 3

11
REFERENCE: les problèmes de criminalité qu' affronte notre société aujourd'hui ne sont pas dus à l' absence de lois .
GREEDY : the problems of crime than facing our society today is are not owed to the absence of legislation .
DP : the problems of crime that this is due non-funding laws .

GREEDY+ : the problems of crime that this problem is due non-funding laws .
REPONSES : 1 1 1 1 1 0

12
REFERENCE: comment cela protège il le canadien ordinaire ?
GREEDY : how that protects the ordinary canadian ?
DP : how that protects the ordinary canadian ?
GREEDY+ : how he would protect the ordinary canadian ?
REPONSES : 1 1 1 1 1 1

13
REFERENCE: il faut faire en sorte de modifier cette perception .
GREEDY : we must make the kind of change that perception .
DP : there must be the kind of change that perception .
GREEDY+ : we must be the kind of change that perception .
REPONSES : 0 1/3 2 1 1 0

14
REFERENCE: comme toujours , mon parti est prêt à proposer une solution de rechange constructive .
GREEDY : as always , my party is willing to propose an alternative to constructive alternatives .
DP : as always , my party is prepared to move an alternative manner .
GREEDY+ : as always , my party is prepared to offer constructive alternative .
REPONSES : 3 3 3 3 3 3

15
REFERENCE: qu' importe que le montant soit minime , il rappellera au contrevenant sa responsabilité .
GREEDY : that matter that the amount is minimal , he recalled the offender 's responsibility .
DP : that is important to small amount , it brings in its responsibility .
GREEDY+ : that is any minimal amount , it brings in its responsibility .
REPONSES : 0 2 1 1 1 0

16
REFERENCE: à long terme , le dédommagement obligatoire sous forme d' indemnité vise deux objectifs importants .
GREEDY : the long term , the required compensation under form of compensation is two major objectives .
DP : in the long term the required compensation form compensation purpose two important functions .
GREEDY+ : the long term , the voluntary form compensation two major objectives aimed .
REPONSES : 0 3 2 2 2 2

17
REFERENCE: premièrement , la victime reçoit une indemnité financière .

GREEDY : first , the victim receives a financial compensation .
DP : first , the victim to receive financial compensation .
GREEDY+ : first , the victim to receive financial compensation .
REPONSES : 2/3 1 1 1 1 1

18
REFERENCE: il serait difficile pour un député de faire abstraction d' une pétition portant deux millions et demi de signatures .
GREEDY : it is difficult for a member to make apart from a petition dealing two million and one-half of signatures .
DP : there is a difficult for him to ignore a petition carries two and one-half millions signatures .
GREEDY+ : it is a difficult for him to ignore a petition of two and one-half millions signatures .
REPONSES : 3 1 3 3 3 3

19
REFERENCE: j' ai maintenant parlé aux parents de trois victimes .
GREEDY : i have been spoken to parents of three victims .
DP : i have now spoken to the parents of three victims .
GREEDY+ : i have talked to the parents of three victims .
REPONSES : 3 2 2 2 3 2

20
REFERENCE: ce n' est pas plus facile d' une fois à l' autre .
GREEDY : what did is no longer easy for a 19.90 with one another .
DP : this is no longer not easy for some time in the other .
GREEDY+ : it is no longer not easy for some time in another .
REPONSES : 0 3 3 3 2 3

21
REFERENCE: deuxièmement , elle dit que les droits des criminels passent avant ceux des victimes dans notre société .
GREEDY : second , she said that the rights of criminals going ahead who are victims in our society .
DP : second , she said that the rights of the criminal to supersede the rights of the victim .
GREEDY+ : second , she said that the rights of block-busting donut are the victims before .
REPONSES : 0 2 2 2 2 1

22
REFERENCE: en 1975 , on a utilisé des armes à feu dans 42 p. 100 des vols .
GREEDY : in 1975 , it has taken some weapons to firearms in 42 per cent of flights .
DP : in 1975 , it are firearms in 42 per cent of flights .
GREEDY+ : in 1975 , has been using firearms in 42 per cent of flights .
REPONSES : 0 3 2 3 3 3

23
REFERENCE: nous examinerons aussi le problème de la violence faite aux femmes .
GREEDY : we will also the problem of the and violence to women .
DP : we also look to the problem of violence to women .
GREEDY+ : we look to the problem of violence to women .
REPONSES : 2 2 2 2 2 2

24
REFERENCE: je vais maintenant les aborder .
GREEDY : i am now in touch .
DP : i want to talk now .
GREEDY+ : i want to address now .
REPONSES : 3 3 3 3 3 0

25
REFERENCE: en fait , il n' a même pas besoin d' ouvrir la bouche .
GREEDY : in fact , he did was not even need an open mouth the .
DP : in fact , he did not even need the from opening .
GREEDY+ : in fact , it did not even need to open the mouth .
REPONSES : 0 3 1 1 3 3

26
REFERENCE: je ne pense pas que mon collègue veuille refuser ce droit aux prévenus .
GREEDY : i cannot think not think my colleague wants refuse are entitled to advance .
DP : i do not think that my colleague wants to deny this right to notice .
GREEDY+ : i do not think that my colleague wants to deny right to notice .
REPONSES : 2 2 2 2 2 3

27
REFERENCE: j' espère que non .
GREEDY : i hope that either .
DP : i hope not .
GREEDY+ : i only hope than .
REPONSES : 2 2 1 2 2 2

28
REFERENCE: deuxièmement , mon collègue a mentionné les conseils et la réhabilitation .
GREEDY : second , my colleague has mentioned them and councils the pardon .
DP : second , my colleague has referred to the advice and rehabilitation .
GREEDY+ : second , my colleague has mentioned the advice , and rehabilitation .
REPONSES : 2 2 3 3 3 2

29
REFERENCE: il est exact que nous essayons de punir , lorsque nous le pouvons .
GREEDY : it is true that we try to punish , when we the can .
DP : there is true that we try to punish the government when we can .
GREEDY+ : it is true that we try to punish , when we can .
REPONSES : 3 3 3 3 3 3

30
REFERENCE: en ce qui concerne les prisons , elles doivent être humaines .
GREEDY : in this regard who the prisons , they must be human .
DP : it in that regard , the prisons will be human .
GREEDY+ : in it that regard , the prisons will be human .
REPONSES : 0 1 2/3 1 2/3 2/3

31
REFERENCE: naturellement , nous privons les condamnés de leur liberté .
GREEDY : naturally , we deprive people convicted of their freedom .
DP : naturally , we are to deprive convicted their freedom .
GREEDY+ : naturally , we are to deprive convicted freedom .
REPONSES : 1 2 3 1 1 0

32
REFERENCE: les prisons sont le reflet des valeurs de la société .
GREEDY : the prisons are pale and values of the corporation .
DP : that the prisons reflect the society values .
GREEDY+ : the government prisons reflect the values society .
REPONSES : 2 2 2 2 2 2

33
REFERENCE: souvent , elles sont même très inconfortables .
GREEDY : often , they are even very uncomfortable .
DP : often , although they are very uncomfortable .
GREEDY+ : often , they are very uncomfortable .
REPONSES : 1 3 3 1 1 1

34
REFERENCE: faisons nous assez pour les victimes de crimes ?
GREEDY : do we enough for the victims of crime ?
DP : we doing enough for the victims of crime ?
GREEDY+ : we doing enough for the victims of crime ?
REPONSES : 1 2/3 1 2/3 1 1

35
REFERENCE: cependant , comme partout , il y a un équilibre à trouver .
GREEDY : however , like everywhere , it has ago a equilibrium at work .
DP : however , like everywhere there has to find a balance .
GREEDY+ : however , like everywhere there has to find a balance .
REPONSES : 2/3 2/3 2/3 2/3 2/3 2/3

36
REFERENCE: à l' heure actuelle , les choses sont sensiblement équilibrées .
GREEDY : at the present time , all things are significantly balanced .
DP : in the present time , the same thing is balanced .
GREEDY+ : at the moment , the same thing is balanced .

REPONSES : 0 3 1 1 1 1

37
REFERENCE: écoutez attentivement les directives .
GREEDY : listen carefully the guidelines .
DP : listen carefully the directives .
GREEDY+ : listen carefully the directives .
REPONSES : 1 1 1 1 1 1

38
REFERENCE: dès l' ouverture des portes , asseyez vous sur le plancher .
GREEDY : today the opening the door , you sit on the floor .
DP : the beginning of open door , you sit on the floor .
GREEDY+ : the beginning of open door , sit on the floor .
REPONSES : 0 2 3 3 0 0

39
REFERENCE: laissez vous glisser sur le sol ou utilisez l' échelle .
GREEDY : let me bootleg on the ground or use of scale .
DP : you let slip on soil or use the scale .
GREEDY+ : let slip on soil or use the scale .
REPONSES : 0 2 2 3 3 3

40
REFERENCE: ouverture manuelle des portes en cas d' urgence .
GREEDY : opening UNK on doors in cases of emergencies .
DP : coal-mining and opens doors in an emergency .
GREEDY+ : and opens doors in an emergency .
REPONSES : 0 1 2 3 3 3

41
REFERENCE: utilisez ce téléphone pour communiquer avec le responsable .
GREEDY : use the telephone to communicate with the responsibility .
DP : to use the telephone communication with the responsibility .
GREEDY+ : use the telephone communication with the responsible
REPONSES : 1 1 3 3 3 1

42
REFERENCE: appuyez et attendez qu' on vous répondre .
GREEDY : stand and watch what you please answer .
DP : answer you wait and support .
GREEDY+ : answer you wait and support .
REPONSES : 0 2/3 2/3 2/3 1 0

43
REFERENCE: si le train est en marche , il s' immobilisera à la prochaine station .
GREEDY : if the train is in march , it is UNK at the next station .
DP : if the train is the way , it is in the next station .

GREEDY+ : if the train is the way , it is in the next station .
REPONSES : 0 2/3 2/3 1 1 0

44
REFERENCE: notre planète est suffisamment chaude pour soutenir la vie .
GREEDY : our planet has enough hot in support of life .
DP : our planet is enough to support the warm .
GREEDY+ : our planet is enough to support the warm .
REPONSES : 0 1 1 1 0 1

45
REFERENCE: d' autres ne sont pas certains .
GREEDY : the others do are not there .
DP : not in others are not there .
GREEDY+ : not to others are not there .
REPONSES : 0 1 1 1 0 0

46
REFERENCE: une partie de l' information figurant sur ce site web vient de sources externes .
GREEDY : a part of the information on hereto what m-6 from tangled of sources factors .
DP : a part of the information contained in that site just tangled of external roots .
GREEDY+ : a part of the information contained on site just tangled of external roots .
REPONSES : 0 1 1 1 1 1

47
REFERENCE: vous quittez le gouvernement du canada .
GREEDY : you leave the government of canada .
DP : government please leave canada .
GREEDY+ : government please leave canada .
REPONSES : 1 1 1 1 1 1

48
REFERENCE: le québec constitue un lieu unique en cette terre d' amérique .
GREEDY : that quebec constitutes a rather unique in this land in america .
DP : the quebec government is a rather unique in this land in america .
GREEDY+ : the quebec is a rather unique in this land in america .
REPONSES : 3 3 3 3 3 0

49
REFERENCE: le québec présente un climat polaire dans l' extrême nord .
GREEDY : the quebec presents a climate in beads the extreme north .
DP : the quebec presents a polar climate in the extreme north .
GREEDY+ : the quebec presents a polar climate in the extreme north .
REPONSES : 2/3 2/3 2/3 2/3 2/3 2/3

50
REFERENCE: les températures moyennes sont bien sûr plus tempérées .

GREEDY : the temperatures averages are certainly well longer boreal .
DP : canadians are much more temperate course average temperatures .
GREEDY+ : canadians are obviously more temperate average temperatures .
REPONSES : 0 1 2 1 3 0