



Laboratoire RALI
Université de Montréal

Mémoire de DEA

Le Traitement des Invariants dans les Systèmes Statistiques de Traduction Automatique

Jérémy BONNET
jeremy.bonnet@polytech.univ-nantes.fr

Version : 1.0

17 août 2004

ENCADRANT	Guy Lapalme	Professeur	RALI
CO-ENCADRANT	Philippe Langlais	Professeur	RALI

DEA ECD - Université de Lyon 2

Remerciements

Je remercie tout d'abord le professeur Guy Lapalme pour m'avoir accueilli au sein du RALI, et ainsi m'offrir la possibilité de travailler dans un laboratoire de linguistique informatique de renommée mondiale. Il a su, par sa gentillesse, sa grande disponibilité et son soutien financier, rendre mon travail fort agréable. Ce stage fut une expérience enrichissante en tous points et je le dois en grande partie à M. Lapalme.

Je tiens de même à témoigner ma grande reconnaissance à mon co-encadrant, Philippe Langlais. Malgré de nombreuses occupations, il a toujours été disponible pour m'aider et m'orienter dans mon travail. De conseils judicieux, il a toujours été d'une aide précieuse et je lui en suis très reconnaissant.

Je remercie également le personnel du RALI, et plus particulièrement les habitués de la pause café qui ont contribué au fait que mon stage se déroule dans une ambiance vraiment agréable.

Je pense aussi à Mehdi, Julien et Richard, mes camarades de promotion, avec qui j'ai vécu de grand moments ici.

Enfin, je n'oublie pas mes parents, mes deux soeurs, mes deux petits neveux Dorian et Maël ainsi que mes amis en France, qui m'ont accompagné dans mes démarches pour partir à Montréal et qui m'ont soutenu pendant ces six mois.

Résumé

La traduction automatique est en plein essor actuellement. Notre travail s'intéresse uniquement aux systèmes probabilistes (*Statistical Machine Translation* en anglais). La SMT repose essentiellement sur l'apprentissage des paramètres de différents modèles à partir d'une grande quantité de textes bilingues (corpus d'entraînement). Naturellement, ce corpus ne contient pas tous les mots existants, et encore moins toutes les *entités nommées* (i.e. les noms de personnes, de lieux et d'organisation). En SMT classique, la tentative de traduction d'un mot inconnu pénalise la qualité de traduction de la phrase qui le contient. Partant du constat que les entités nommées constituent généralement des invariants de traduction, l'objectif ici est de modifier le système classique afin de conserver ces invariants dans le but d'améliorer la qualité des traductions.

Dans une première approche, nous entraînons les modèles sur un corpus modifié afin d'*apprendre* au système à conserver les entités nommées. Par l'étude des limites de cette démarche, nous en proposons une amélioration. Enfin, nous évaluons enfin les performances de ces deux approches.

Mots clés : traduction automatique, statistical machine translation, entités nommées, décodeur, évaluation de traduction.

Abstract

Nowadays Machine Translation is getting more and more important. The work here is only dealing with Statistical Machine Translation. The principle of SMT is to learn different model parameters from an important quantity of bilingual texts (training corpus). This corpus obviously doesn't contain all existing words nor all the named entities (i.e. persons, locations and organizations names). In classical SMT, the attempt at translating an unknown word makes the entire sentence translation quality collapse. Keeping in mind that the named entities are generally translation invariants, the goal here is to modify the traditional system to preserve these invariants so that the translations quality is improved.

In a first approach, we train the models on a corpus that we modify in order to *teach* the system to preserve the named entities. By stressing the problems, we propose next an improvement of this approach. Eventually we evaluate the performances of those two approaches.

Key words : machine translation, statistical machine translation, named entities, decoding, translation evaluation.

Table des matières

Remerciements	2
Résumé	3
Abstract	4
1 Introduction	7
2 Présentation de l'existant	9
2.1 La traduction automatique statistique	9
2.1.1 Canal bruité	9
2.1.2 Le modèle de langue	11
2.1.3 Les modèles de traduction	11
2.1.4 Décodage	12
2.2 Évaluation de la traduction	14
2.3 Gestion actuelle des invariants	16
3 Première approche	18
3.1 Modélisation de la conservation des invariants	18
3.2 Données d'entraînement et de test	20
3.3 Résultats	20
3.3.1 Repérage d'Entités Nommées	20
3.3.2 Mesure de qualité de traduction	21
3.3.3 Amélioration des scores	22
3.3.4 Limites	22
4 Deuxième approche	24
4.1 Modélisation	24
4.1.1 Bruit dans le modèle de traduction	24
4.1.2 Entités nommées non invariants	24

4.2	Résultats	26
4.2.1	Amélioration des scores	26
4.2.2	Limites	27
5	Conclusion	28
5.1	Bilan de l'étude	28
5.2	Prolongements et améliorations	29
5.3	Conclusion générale	30

Chapitre 1

Introduction

La traduction automatique (TA) d'une langue humaine à une autre en utilisant les ordinateurs est désignée dans la littérature anglophone sous le terme de " Machine Translation " (MT). C'est un domaine 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 TA se concentrent principalement sur la traduction de textes scientifiques, techniques, commerciaux, officiels et médicaux. La traduction d'oeuvres littéraires reste assez marginale.

La traduction automatique a connu une évolution très importante depuis le début de son développement dans les années 1960. Il existe actuellement entre autres la traduction par règles (en anglais, Rule-Based Machine Translation RBMT), la traduction guidée par l'exemple (Example-Based Machine Translation EBMT) et la traduction statistique (Statistical Machine Translation SMT). 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. Une équipe d'IBM a publié les résultats de ses expériences sur CANDIDE [Berger et. al., 1994], un système de traduction purement statistique. C'est sur la SMT que porte ce travail.

Un problème récurrent en traduction automatique est la gestion des mots inconnus. Le terme de mot inconnu représente tous les mots que le système de traduction automatique ne connaît pas. Ainsi, cela regroupe les mots avec des fautes d'orthographe ou avec des erreurs de frappe ainsi que les noms propres. Cependant, la gestion de ces deux types de mots inconnus est très différente. En effet, les mots mal orthographiés ou les fautes de

frappe doivent être traduits par le système et la difficulté réside dans le fait de trouver la forme originale exacte du terme. Les noms propres, quant à eux, de la même manière que les nombres, constituent généralement des invariants de traduction, c'est-à-dire qu'ils ne doivent pas être modifiés par l'opération de traduction (eg. "Paris" ou "150" se trouvent inchangés lors du passage du français vers l'anglais). Le problème est donc très différent.

L'objectif de ce travail n'est pas de rendre le système statistique de traduction automatique plus robuste aux données bruitées en entrée, mais plutôt d'améliorer la qualité de la traduction de documents bien formés. C'est pourquoi nous faisons l'hypothèse de textes sans faute d'orthographe, de frappe, ni de grammaire en entrée. Notre gestion des mots inconnus comprend donc exclusivement les invariants de traduction qui englobent les nombres et les noms propres. Ces derniers peuvent être segmentés en trois sous-parties : les noms de *personnes*, de *lieux* et d'*organisations*. Dans le monde de la linguistique, on regroupe ces classes de mots sous le terme d'*entités nommées* (ou *named entities* en anglais).

Notre approche du problème est de modifier les textes bilingues (corpus d'entraînement) à partir desquels sont inférés les paramètres des modèles probabilistes dans le but d'*apprendre* au système à conserver les invariants de traduction. Cette tâche peut se diviser en trois sous-problèmes : la recherche des entités nommées; la conservation des invariants de traduction; et la mesure de l'impact des procédés mis en oeuvre en terme de qualité de traduction. Nous présentons donc une première approche et par l'étude de ses limites, nous proposons ensuite une amélioration permettant de résoudre un certain nombre de problèmes.

Chapitre 2

Présentation de l'existant

2.1 La traduction automatique statistique

Afin de poursuivre notre objectif, il est indispensable d'expliquer le principe de la traduction probabiliste ainsi que ses modèles mathématiques. Cette section présente donc le fonctionnement de ce type de traduction, ainsi que les différentes métriques utilisées pour son évaluation. Enfin, nous nous intéressons à la gestion actuelle des invariants en SMT, ce qui constitue notre base de départ.

2.1.1 Canal bruité

La traduction statistique repose sur la métaphore du canal bruité de Shannon qui a déjà fait ses preuves dans les systèmes de traitement de la parole. 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 2.1).

Le but pour R est de retrouver la phrase source à partir de la phrase reçue et de 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) . Pour ce faire, il faut déterminer les paramètres du canal en observant suffisamment de transmissions de phrases, c'est-à-dire de paires de phrases en relation de traduction.

Le problème général de la traduction statistique est de trouver la phrase e , étant donnée une phrase f^J , qui maximise $P(e^I|f^J)$ où I est le nombre de mots de la phrase anglaise

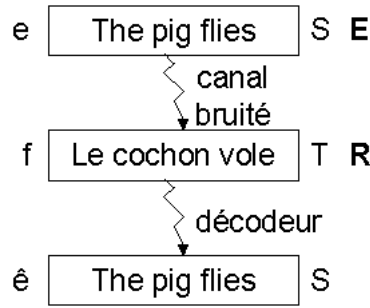


Fig. 2.1. Illustration du canal bruité. L'anglais est ici le langage source du canal et le français le langage cible.

et J le nombre de mots de la phrase française. De manière plus formelle :

$$\hat{e} = \arg \max_e [P(e^I | f^J)] \quad (2.1)$$

D'après le théorème de Bayes :

$$P(e^I | f^J) = \frac{P(f^J | e^I) \times P(e^I)}{P(f^J)} \quad (2.2)$$

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

$$\hat{e} = \arg \max_e P(e^I | f^J) = \arg \max_e P(e^I) \times P(f^J | e^I) \quad (2.3)$$

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. La maximisation représente le décodage.

Cette équation 2.3 résume le problème de la traduction statistique qui comprend trois objectifs : le calcul des paramètres du modèle de langue ; le calcul des paramètres du modèle de traduction ; la réalisation d'un décodeur, c'est-à-dire d'un mécanisme capable d'effectuer l'opération de maximisation. Il y a donc deux distributions à modéliser. Les paramètres de ces modèles sont inférés à partir d'un corpus d'entraînement.

2.1.2 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 :

$$\sum_i P(e^i) = 1 \quad (2.4)$$

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

$$P(e^I) = \prod_{i=1}^I P(w_i | \underbrace{w_1 \cdots w_{i-1}}_h) \quad (2.5)$$

où h est appelé 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. 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 trop complexe. On peut le simplifier en conditionnant la probabilité d'un mot seulement par les deux derniers mots dans l'historique de w . Cette simplification est appelée un modèle *trigramme* :

$$P(e^I) \simeq P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1w_2)P(w_4|w_2w_3) \cdots P(w_I|w_{I-2}w_{I-1}) \quad (2.6)$$

2.1.3 Les modèles de traduction

Le calcul de $P(f^J|e^I)$, la probabilité d'une phrase f^J étant donnée une phrase anglaise e^I constitue le deuxième problème de la traduction automatique probabiliste. On appelle la méthode qui permet de calculer cette distribution *un modèle de traduction*.

Les paramètres de ce modèle sont calculés à partir d'un corpus constitué de deux textes alignés au niveau des phrases¹. Le découpage en "mots" est aussi connu. L'idée est que toute paire de mots (source/cible) rencontrée dans le corpus d'entraînement est un

¹La tâche qui consiste à mettre en correspondance dans un corpus bilingue les phrases qui sont en relation de traduction est appelée l'appariement automatique de textes.

paramètre du modèle, c'est-à-dire qu'on associe une probabilité à cette paire.

On appelle la sortie d'un modèle de traduction, les probabilités de transfert ou encore le lexique bilingue probabilisé. Le tableau 2.1 est un exemple de sortie.

the	(le,0.18)(la,0.15)(de,0.12)
minister	(ministre,0.8)(le,0.12)
people	(gens,0.25)(les,0.16)(personnes,0.1)
years	(ans,0.38)(années,0.31)(depuis,0.12)

Tab. 2.1. Exemple d'extrait d'un modèle de traduction.

L'alignement en entrée du modèle de traduction étant seulement au niveau des phrases, il est indispensable de considérer les alignements au niveau des mots afin de calculer les probabilités de transfert.

Les modèles de traduction proposés par IBM

[Brown et al, 1993], une équipe de chercheurs d'IBM, voit donc un modèle de traduction 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. Dans les modèles IBM, seuls les alignements où chaque mot cible est associé à un mot source (et un seul) sont considérés. On désigne l'ensemble des alignements considérés par les modèles de traduction d'IBM entre les deux phrases f^J et e^I par $A(e, f)$.

En fait, [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 $P(f|e)$.

2.1.4 Décodage

Nous abordons ici le problème du décodage en SMT. Dans la traduction automatique probabiliste et pour notre exemple, le but du décodeur est de chercher la phrase anglaise $e^I = e_1, \dots, e_I$ la plus probable étant donnée une phrase source française $f^J = f_1, \dots, f_J$ et des modèles (modèle de langue et modèle de traduction) où I et e_i ($i \in [1, I]$) sont des

inconnus (figure 2.2).

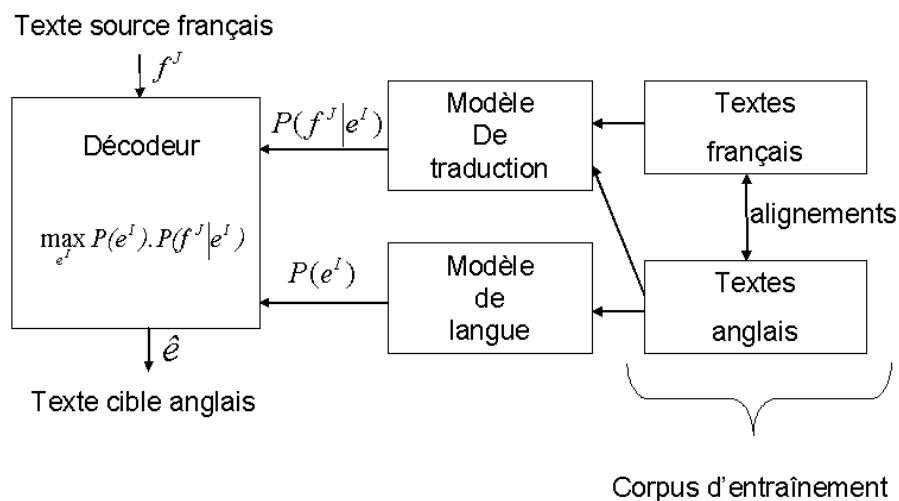


Fig. 2.2. L'architecture de la traduction probabiliste [Nießen et al., 1998].

Revenons à l'équation vue ci-avant :

$$\hat{e} = \arg \max_e P(e^I | f^J) = \arg \max_e [P(e^I) \times P(f^J | e^I)]$$

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 $P(e^I | f^J)$.

L'opération de maximisation est une opération complexe. En effet, [Knight K., 1999] a démontré sa NP-complétude. On utilise donc des simplifications qui permettent de factoriser certains calculs et donc de diminuer la complexité calculatoire du décodeur.

Le décodage suppose que tous les mots sont connus des modèles et aucun traitement particulier n'est effectué en cas de mot inconnu. L'opération de maximisation se déroulera exactement de la même manière.

2.2 Évaluation de la traduction

Une fois la traduction réalisée, il est indispensable de pouvoir l'évaluer. L'évaluation humaine est bien sûr une méthode pour déterminer la performance d'un système de traduction. Cependant, un des gros problèmes de l'évaluation humaine est le temps qu'elle nécessite, ce qui explique qu'elle soit presque exclusivement réservée à l'évaluation de systèmes stables. Dans notre cas, nous avons un besoin d'évaluation rapide de notre système en cours de développement. C'est pourquoi nous utilisons ici deux métriques bien connues : le Word Error Rate (WER) et le Sentence Error Rate (SER) d'une part, et BLEU et NIST d'autre part.

WER et SER

Le WER est calculé à partir de la distance d'édition au niveau du mot entre une phrase de référence x et une phrase traduite y . La distance d'édition entre x et y est définie à partir d'opérations d'édition qui permettent de transformer x en y . Classiquement, on définit trois opérations d'édition de base auxquelles sont associées un coût, indépendamment de la position : la *substitution*, l'*insertion* et la *délétion*. Le coût d'une transformation de x en y est obtenu en sommant le coût de chaque opération (il est fréquent d'associer le même coût aux trois opérations d'édition). Le nombre minimum d'opérations à appliquer pour transformer une phrase en une autre est appelée distance de Levenshtein [Levenshtein, 1966].

Le SER mesure le pourcentage de phrases pour lesquelles la traduction n'est pas identique à celle de référence.

Dans l'exemple suivant (tableau 2.2), on traduit du français vers l'anglais. SRC désigne 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. Les scores WER et SER sont présentés dans le tableau 2.3.

On remarque que SER est très sévère parce que cette métrique accorde un taux d'erreur maximal aux phrases qui ne sont pas parfaitement exactes, même si, comme dans notre exemple, la traduction se rapproche beaucoup de celle de référence. Cependant, WER est moins sévère, sa mesure est plus nuancée et n'accorde que 20% d'erreur sur notre exemple. De ce fait, WER est plus juste que SER.

SOURCE	: tuesday , april 13 , 2004
REFERENCE	: le mardi 13 avril 2004
CANDIDAT	: mardi 13 avril 2004

Tab. 2.2. Exemple de phrases à évaluer.

Insertion	: 0
Délétion	: 1
Substitution	: 0
Exact	: 4
WER	: 20%
SER	: 100%

Tab. 2.3. Résultats de l'évaluation sur les phrases du tableau 2.2.

Il est à noter que le WER et le SER sont des mesures d'erreur. La phrase mesurée est donc d'autant mieux traduite que le taux est faible.

BLEU et NIST

BLEU (BiLingual Evaluation Understudy) est une méthode pour évaluer une traduction automatique présentée par [Papineni et al, 2002]. L'idée de BLEU est de comparer les phrases de traduction et de référence en se basant sur les séquences n-grams (calcul pondéré sur les unigrammes, bigrammes, trigrammes et quadrigrammes). Une traduction est d'autant meilleure qu'elle partage un grand nombre de n-grams avec une ou plusieurs traductions de référence. BLEU donne un score entre 0 et 1. Plus le score est élevé et meilleure est la traduction.

[Papineni et al, 2002] ont montré que BLEU est cohérente avec l'évaluation humaine. Nous accorderons donc beaucoup d'importance à cette mesure.

Tout comme BLEU, la mesure NIST [NIST Speech Group] est basée sur les n-grams au niveau du corpus. Ces deux mesures dominent la plupart des travaux en traduction automatique et ont déjà prouvé leur efficacité.

Le score BLEU obtenu lors de l'évaluation de la phrase candidate de l'exemple précédent est de 0.78 et le NIST est de 1.88 .

2.3 Gestion actuelle des invariants

Comme nous venons de le voir, la SMT s'appuie sur des modèles probabilistes (modèles de langue et de traduction) dont les paramètres sont inférés à partir d'un corpus d'entraînement. Un mot est dit inconnu s'il ne fait pas partie du corpus d'entraînement, et donc s'il n'est pas présent dans les modèles. La traduction en sortie du décodeur est le fruit d'une maximisation globale de probabilités, c'est pourquoi il est difficile de prévoir un résultat localement. Cependant, on observe que les mots inconnus sont souvent traduits par un mot fréquent du vocabulaire, par exemple "the" si on considère l'anglais comme langue cible. Si les modèles connaissent la notion de 'mot inconnu', il peut arriver de trouver ce résultat en sortie. Ou encore, la rencontre d'un tel mot augmente l'incertitude en sortie du décodeur. Mais dans tous les cas, les entités nommées ne sont jamais considérées comme invariants de traduction.

Considérons l'exemple suivant et comparons les résultats (tableau 2.4).

SOURCE	:	des agriculteurs de la vendée sont venus à mon bureau la semaine dernière .
CIBLE	:	some farmers to come to my office last week .

Tab. 2.4. Exemple de SMT d'une phrase contenant un mot inconnu.

Dans cet exemple, le mot *vendée* ne fait pas partie du corpus d'entraînement. On observe plusieurs problèmes. Pour commencer, la rencontre d'un mot inconnu augmente l'incertitude en sortie du décodage et donne une phrase syntaxiquement erronée, ce qui constitue un premier problème. De plus, l'entité nommée *vendée* a complètement disparu de la cible, ce qui est extrêmement pénalisant au niveau de la sémantique. Pour bien réaliser l'impact du phénomène, considérons l'exemple similaire du tableau 2.5.

Ici, le décodeur ne rencontre pas de mot inconnu et donne donc une traduction de bien meilleure qualité. On n'observe aucun problème au niveau syntaxique ni sémantique. Cette non gestion des mots inconnus donne en SMT de mauvais résultats. Le problème se

SOURCE	:	des agriculteurs de la région sont venus à mon bureau la semaine dernière .
CIBLE	:	some farmers in the area came to my office last week .

Tab. 2.5. Exemple de SMT d'une phrase ne comprenant pas de mot inconnu.

pose principalement sur les entrées dont le sujet est éloigné de celui du corpus d'entraînement, c'est-à-dire qui comprennent de nombreux mots inconnus vis-à-vis des modèles.

Chapitre 3

Première approche

Ce chapitre présente notre première approche de résolution du problème. Nous commençons par modéliser notre solution et décrire nos données de travail. Nous présentons ensuite nos résultats et leur analyse.

3.1 Modélisation de la conservation des invariants

L'entraînement des modèles sur un corpus d'entraînement crée un vocabulaire. Ce vocabulaire va contenir certaines entités nommées, mais ne peut pas évidemment pas les contenir toutes. L'idée est de modifier le corpus d'entraînement à partir duquel les paramètres des modèles de langue et de traduction sont inférés afin d'*apprendre* au système à conserver les entités nommées. On effectue donc un *prétraitement* sur le corpus d'entraînement qui consiste à rechercher les entités nommées puis à les remplacer par un terme spécial. Les modèles de langue et de traduction sont ré-entraînés sur ce corpus prétraité afin d'incorporer ce nouveau terme.

Le tableau 3.1 présente un extrait du modèle de langue ainsi entraîné.

<i>entité nommée</i>	:	(<i>entité nommée</i> , 0.78) (le , 0.13)
beach	:	(plage , 0.24) (<i>entité nommée</i> , 0.14)

Tab. 3.1. Extrait du modèle de traduction entraîné sur un corpus prétraité.

En ce qui concerne le décodage, on réalise le même prétraitement sur le texte source (i.e. recherche des entités nommées et remplacement de celles-ci). Cependant, les entités nommées sont stockées avant d'être remplacées, afin de pouvoir reconstituer le texte cible. Le décodage s'effectue ensuite en utilisant les modèles ré-entraînés. On obtient une traduction brute. On effectue finalement un *post-traitement* qui consiste à replacer les entités nommées en utilisant la sauvegarde.

Cette approche peut être schématisée par la figure 3.1 comme une variante de la figure 2.2.

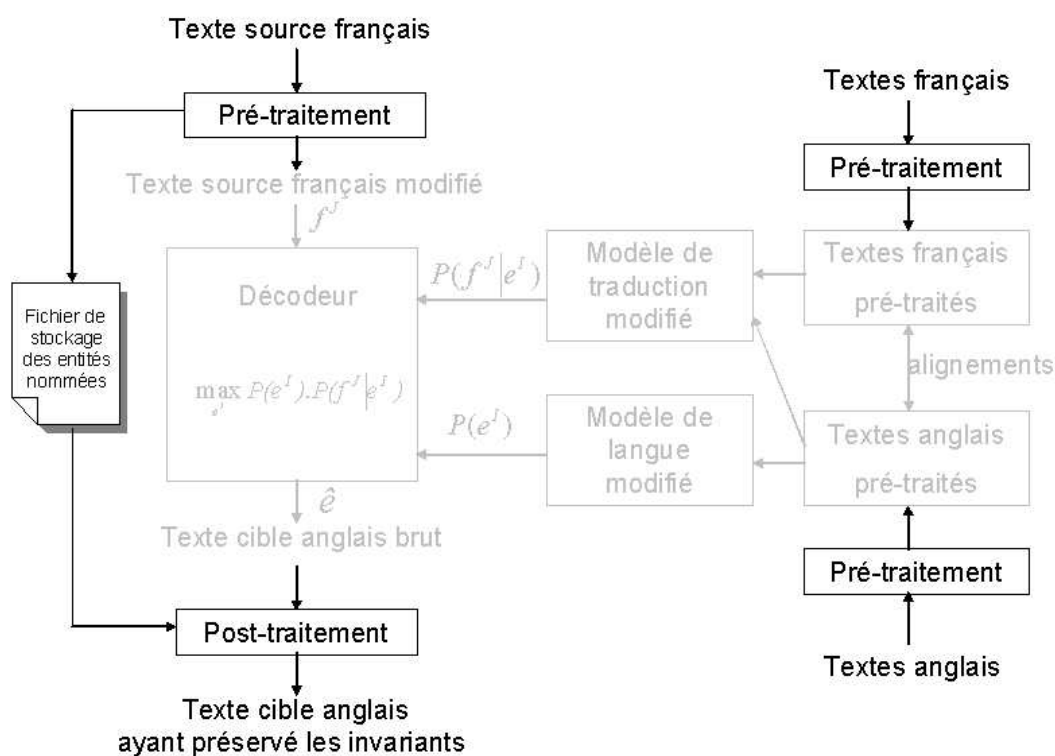


Fig. 3.1. L'architecture de la traduction probabiliste [Nießen et al., 1998] présentée à la figure 2.2 mais modifiée pour conserver les invariants de traduction.

Cette approche a l'avantage d'être complètement indépendante du décodeur, et même de toute la chaîne de traitement actuelle. On réalise simplement un prétraitement et un post-traitement, ce qui permet d'assurer une certaine portabilité. De plus, la tâche de reconnaissance des entités nommées est elle aussi indépendante et peut se faire par l'intermédiaire de différents modèles (arbres de décision, entropie maximum, réseaux de

neurones...).

3.2 Données d'entraînement et de test

Dans notre étude, nous avons utilisé le corpus des débats parlementaires canadiens, connu sous le nom de Hansard, pour entraîner nos différents modèles. Ce corpus est constitué de 1 639 250 paires de phrases, de 31 826 112 mots français et de 29 547 933 mots anglais. Les tailles des vocabulaires français et anglais sont respectivement de 103 830 et de 83 106 mots différents.

Le décodeur utilisé est celui développé par M. Langlais du RALI. Il s'appuie sur le modèle de traduction IBM2. Le système de reconnaissance des entités nommées est un taggeur développé par le RALI.

Nous réalisons nos expériences en traduisant du français vers l'anglais. Pour réduire les temps de calcul, nous n'avons considéré que les phrases ne dépassant pas 25 mots. Le document de test est un extrait du Hansard 2003 composé de 2 101 phrases et de 1 447 entités nommées identifiées par le taggeur.

3.3 Résultats

3.3.1 Repérage d'Entités Nommées

Sur la figure 3.1, on voit bien la place centrale du système de repérage des entités nommées. Cette tâche a donc dû être évaluée afin de valider le système utilisé.

Protocole d'évaluation

L'évaluation d'un système de REN consiste au calcul de sa précision, de son rappel et enfin de sa F-mesure équilibrée sur un document de test. La précision est en fait le pourcentage d'entités nommées repérées par le système s'avérant effectivement être entités nommées. Le rappel mesure le pourcentage de véritables entités nommées reconnues par le système. Quant à la F-mesure équilibrée comme la présente [Rijsbergen, 1979], il s'agit d'un calcul simple à partir de la précision et du rappel dans le but de donner le meilleur compromis entre ces deux mesures. Mathématiquement,

$$F - \text{mesure} = 2(P \times R)/(P + R)$$

L'évaluation a porté sur un document de test extrait du corpus d'entraînement. Il a fallu au préalable réaliser le repérage manuellement pour obtenir la référence. Les résultats sont présentés dans le tableau 3.2.

Précision	Rappel	F-mesure
93.71%	93.39%	93.55%

Tab. 3.2. Résultats de l'évaluation de la tâche de repérage d'entités nommées.

Ces scores sont très élevés. En comparaison, un étiquetage manuel donne des résultats entre 90% et 92% [Sundheim, 1995]. Nous validons donc notre système de repérage d'entités nommées.

3.3.2 Mesure de qualité de traduction

La mesure de l'apport de notre système se fait par l'évaluation de la différence des mesures de qualité de la traduction entre le même texte traduit par les deux méthodes. Une traduction *classique*, utilisant l'architecture de [Nießen et al., 1998], d'un texte est comparée à la traduction du même texte utilisant l'architecture modifiée.

Les différents scores sont présentés dans le tableau 3.3.

Système	WER	SER	NIST	BLEU
classique	68.96%	97.21%	1.6522	0.0598
avec REN	66.99%	97.48%	1.7212	0.0667

Tab. 3.3. Premiers résultats

Globalement, les scores de traduction se voient améliorés.

3.3.3 Amélioration des scores

Dans un premier temps, on constate une bonne augmentation du BLEU et du NIST et une diminution du WER de presque 2%. Afin de mieux expliquer ce résultat, reprenons l'exemple utilisé ci-avant (tableau 3.4).

SOURCE	des agriculteurs de la vendée sont venus à mon bureau la semaine dernière .
REF	the famers of vendée were in my office last week .
CIBLE CLAS	some farmers to come to my office last week .
CIBLE REN	ome farmers of vendée came to my office last week .

Tab. 3.4. Différences entre la traduction classique et le système modifié.

CIBLE CLAS est la traduction obtenue par le système classique et *CIBLE REN* est obtenue par le système utilisant la reconnaissance d'entités nommées. L'entité nommée *Vendée* est conservée par le système modifié alors qu'elle est perdue par le système classique. De plus, le fait que la phrase comporte un mot inconnu pour le système classique nuit au décodeur qui donne une phrase grammaticalement éloignée de la référence. Cette diminution du WER est donc due d'une part à la préservation des invariants de traduction, et d'autre part à la diminution du nombre de mots inconnus des modèles, ce qui permet de baisser l'indéterminisme au niveau du décodeur.

3.3.4 Limites

Dans un second temps, on remarque que le SER a très légèrement augmenté. Ceci peut s'expliquer par le fait que le terme spécial désignant une *entité nommée* dans le corpus d'entraînement est devenu très fréquent, en fait le troisième mot le plus fréquent (1 million d'occurrences) derrière le point . (1,5 millions) et *the* (2 millions). De ce fait, le terme *entité nommée* est très fréquent dans le modèle de traduction et y introduit du bruit. Il apparaît une fois tous les 18 mots dans ce modèle. Il en résulte un nombre important d'*entités nommées* dans le texte cible brut là où aucune entité n'est présente dans le texte source.

Un autre problème est mis en évidence. Dans notre système, toutes les entités nommées reconnues par le tagueur sont considérées comme des invariants de traduction. Cependant,

certaines noms de personnes, de lieux et d'organisation varient par l'opération de traduction, eg. *Ben Laden* en français devient *Bin Laden* en anglais, *états-unis* devient *united states* ou encore l'*ONU* devient l'*UNO*. Il peut être bénéfique de distinguer les entités nommées invariables de celles qui ne le sont pas, afin d'éviter les erreurs comme celle présentée dans le tableau 3.5.

SOURCE	:	les états-unis ont toujours respecté les règles de la guerre .
CIBLE REN	:	the états-unis have always respected the rule of war .

Tab. 3.5. Exemple d'entité nommée considérée à tort comme invariant de traduction.

Afin de pallier à ces deux problèmes, nous avons expérimenté une seconde approche du problème.

Chapitre 4

Deuxième approche

Par l'étude de la sortie du décodeur de la première approche, nous avons mis en exergue deux problèmes principaux. Cette deuxième approche tente d'y remédier afin d'améliorer encore les scores de traduction.

4.1 Modélisation

4.1.1 Bruit dans le modèle de traduction

Nous avons vu que le repérage des entités nommées introduit du bruit dans le modèle de traduction. Ce bruit entraîne que le décodeur génère des entités nommées dans le texte cible là où aucune entité nommée n'est présente dans le texte source. Pour pallier à ce problème, nous avons entraîné le modèle de traduction sur le corpus duquel les entités nommées ont été supprimées. De plus, la probabilité $P(\text{entité nommée}|\text{entité nommée}) = 1$ a été ajoutée. Ceci permet de supprimer le bruit lié à la reconnaissance d'entités nommées dans le modèle de traduction.

4.1.2 Entités nommées non invariants

Notre second problème est de distinguer les entités invariants de celles qui ne le sont pas. Pour ce faire, nous avons divisé le corpus d'entraînement en deux parties égales. Une moitié n'a pas été modifiée et l'autre a subi le prétraitement. De plus, seules les entités nommées n'appartenant pas au corpus d'entraînement ont été remplacées dans le texte

à traduire. Ainsi, les entités nommées n'étant pas des invariants de traduction et que le système était capable de traduire sont bien traduites, eg. *états-unis* se voit bien traduit en *united states*. Il est à noter que la fréquence des entités nommées dans le corpus d'entraînement s'en trouve diminué.

La figure 4.1 présente l'architecture de ce nouveau système.

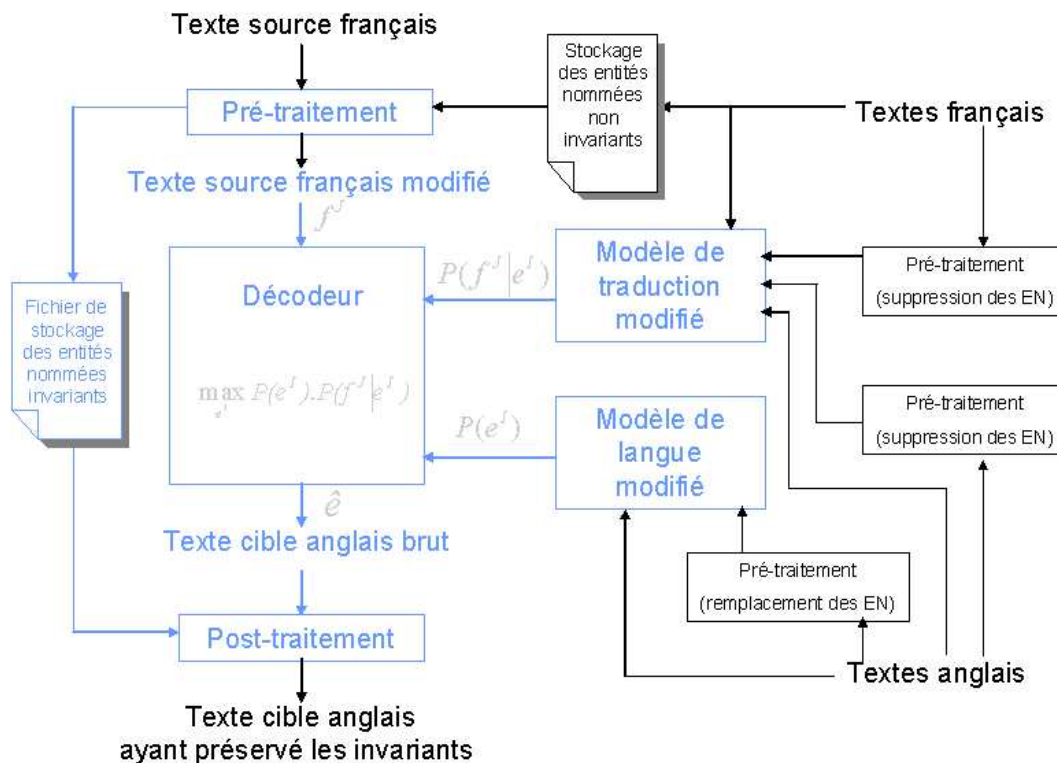


Fig. 4.1. Amélioration du système précédent (figure 3.1) permettant de supprimer le bruit lié à la reconnaissance d'entités nommées dans le modèle de traduction et de considérer certaines entités comme non invariants.

Données d'entraînement et de test

Afin de pouvoir comparer les résultats entre nos deux approches, les données d'entraînement des modèles et de test sont les mêmes ici que pour la première approche.

4.2 Résultats

Le tableau 4.1 présente les résultats de l'architecture classique de [Nießen et al., 1998] et ceux de nos deux approches.

Système	WER	SER	NIST	BLEU
CLAS	68.96%	97.21%	1.6522	0.0598
REN	66.99%	97.48%	1.7212	0.0667
REN MOD	67.28%	96.69%	1.7660	0.0688

Tab. 4.1. Résultats de la seconde approche

Les résultats vont encore dans le même sens, à savoir vers un gain de qualité de traduction.

4.2.1 Amélioration des scores

Les scores BLEU et NIST sont encore améliorés. Cette modification permet de bien traduire les entités nommées n'étant pas des invariants de traduction (évidemment, à partir du moment où elles appartiennent au corpus d'entraînement). Nous illustrons ce phénomène en reprenant l'exemple précédent et en observant la sortie du système (tableau 4.2).

SOURCE	:	les états-unis ont toujours respecté les règles de la guerre .
CIBLE REN	:	the états-unis have always respected the rule of war .
CIBLE REN MOD	:	the americans have always respected the rules of war .

Tab. 4.2. Illustration de l'apport du système modifié pour le traitement des entités nommées n'étant pas invariants de traduction.

La sortie du système modifié (CIBLE REN MOD) traduit bien l'entité nommée non invariant *états-unis*. La traduction gagne donc en qualité.

4.2.2 Limites

Cependant, certains problèmes subsistent. Par exemple, lorsque le prénom d'une personne est connu du modèle de langue et que son nom ne l'est pas, alors le modèle de langue assigne en sortie le nom de famille le plus fréquent du corpus d'entraînement associé à ce prénom (tableau 4.3).

SOURCE	:	c' est bien m. peter .
CIBLE REN MOD	:	it is mr. peter .
SOURCE	:	c' est bien m. peter baret .
CIBLE REN MOD	:	it is mr. peter milliken baret .

Tab. 4.3. Exemple de défaut du système.

Dans cet exemple, *Peter* est connu du modèle de langue alors que *baret* ne l'est pas ; on voit alors apparaître le nom *milliken* en sortie. Ceci peut expliquer la légère augmentation du WER constatée dans le tableau 4.1.

Chapitre 5

Conclusion

5.1 Bilan de l'étude

La gestion des entités nommées dans les systèmes de traduction automatique statistique pose problème. En effet, la plupart de ces entités ne sont pas présentes dans les modèles et les performances du système s'effondrent lors de la rencontre d'un mot inconnu. Ce travail vise donc à proposer une meilleure gestion dans le but d'améliorer la qualité des traductions.

Le principe de notre approche du problème est de modifier le corpus d'entraînement afin d'*apprendre* au système à conserver les entités nommées. Les résultats obtenus sont encourageants, mais l'étude de la sortie du décodeur nous permet de mettre en évidence un certain nombre de problèmes dans nos modèles. Nous proposons donc une modification de cette approche dans le but de supprimer le bruit lié à la reconnaissance des entités nommées dans le modèle de traduction et à faire une distinction entre les entités nommées invariants de celles qui ne le sont pas. Les performances vont encore dans le sens d'une amélioration de la qualité de traduction.

Nous pouvons donc dire que l'objectif d'améliorer les performances du système en proposant une meilleure gestion des invariants de traduction a été atteint.

5.2 Prolongements et améliorations

Etant donnés ces résultats encourageants, nous avons complété cette étude par l'utilisation du décodeur *Pharaoh* développé par [Koehn, 2003] et utilisant les Phrase-Based Models. La théorie de la traduction statistique basée sur les phrases est décrite dans [Zens et al., 2002]. Ce décodeur est intéressant pour nous car il autorise une entrée avec un balisage XML permettant de forcer une traduction en sortie, ce qui est exactement ce que nous désirons réaliser. Nos modèles de langue et de traduction ont été modifiés afin d'être compréhensibles par ce nouveau décodeur. Les résultats sont présentés dans le tableau 5.1.

Système	WER	SER	NIST	BLEU
classique	68.96%	97.21%	1.6522	0.0598
avec REN	66.99%	97.48%	1.7212	0.0667
avec REN MOD	67.28%	96.69%	1.7660	0.0688
Pharaoh	65.60%	96.62%	5.071	0.214

Tab. 5.1. Comparaison des scores avec Pharaoh.

Les scores sont très bons et dépassent même largement (en terme de NIST et de BLEU) ceux de notre deuxième approche.

Toute cette étude nous a permis de légitimer le fait de changer complètement notre approche du problème. Afin de se rapprocher de *Pharaoh*, nous avons modifié notre décodeur SMT afin de conserver les invariants. Le principe ne réside plus dans un traitement en amont et en aval de l'architecture de [Nießen et al., 1998], mais il s'agit maintenant d'agir directement sur le décodeur afin de considérer tous les mots inconnus des modèles comme invariants de traduction. Notre décodeur a donc été modifié en conséquence. Le tableau 5.2 présente ces scores.

Les résultats obtenus sont proches de ceux de *Pharaoh* et en tout cas, sont meilleurs que ceux de nos deux approches initiales. Toute cette étude nous a montré que les scores obtenus par la modification de l'architecture classique allaient dans le bon sens et donc, a légitimé le fait de proposer une meilleure gestion des invariants directement dans le décodeur.

Système	WER	SER	NIST	BLEU
classique	68.96%	97.21%	1.6522	0.0598
avec REN	66.99%	97.48%	1.7212	0.0667
avec REN MOD	67.28%	96.69%	1.7660	0.0688
Pharaoh	65.60%	96.62%	5.071	0.214
décodeur modifié	64.05%	95.48%	4.974	0.195

Tab. 5.2. Comparaison des scores avec le décodeur modifié.

5.3 Conclusion générale

Ce stage a été très enrichissant, et ce à plusieurs niveaux.

Il m'a permis de découvrir le fonctionnement d'un laboratoire et de son équipe. D'autre part, ce premier travail concret de recherche m'a permis d'acquérir une rigueur intellectuelle indispensable dans ce domaine.

J'ai pu apprendre à travailler au sein d'une équipe de recherche, bénéficier des compétences de ses membres et, à l'inverse, faire part de mes connaissances. De plus, certaines contraintes, de temps et d'organisation, m'ont été imposées, notamment pour la présentation finale de mes travaux lors du séminaire du RALI.

D'autre part, cette expérience à l'étranger sera un atout très important pour ma carrière professionnelle, et j'en suis pleinement satisfait. Au niveau personnel, c'est une chance extraordinaire que d'évoluer pendant six mois dans un tel contexte, de découvrir au quotidien une culture nord-américaine.

Bibliographie

- [1] Brown P. F. Cocke J. Della Pietra S. Della Pietra V. Jelinek F. Mercer R. Roossin P., *A Statistical Approach to Machine Translation*, Computational Linguistics. 16(2). pp. 79-85, 1990.
- [2] Brown P. F. Pietra S. A. D. Pietra V. J. D. Mercier R. L., *The mathematics of statistical machine translation : Parameter estimation*, Computational Linguistics. 19(2). pp. 263-311, 1993
- [3] Knight K., *Decoding complexity in word-replacement translation models*, Computational Linguistics. 25(4). pp. 607-615, 1999
- [4] Nießen S. Vogel S. Ney H. Tillmann C., *A DP Based Search Algorithm for Statistical Machine Translation*, *Proceedings of COLING-ACL '98*, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Montreal Quebec Canada. volume 2. pp. 960-967, 1998
- [5] Papineni K. Roukos S. Ward T. Zhu W., *BLEU : a Method for Automatic Evaluation of Machine Translation*, *Proceedings of COLING-ACL '02*. Philadelphia. USA. pp. 311-318, 2002
- [6] Berger et al., *The Candide system for machine translation*, *Proceedings of the ARPA Workshop on Human Language Technology*. pp. 152-162, 1994
- [7] Vauquois B., *A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation*, IFIP Congress-68 Edinburgh pp.254-260, 1968 - reprinted in Ch. Boitet (ed.), *Bernard Vauquois et la TAO : Vingt-cinq ans de Traduction Automatique - Analectes*, Grenoble. pp. 201-213, 1988
- [8] Levenshtein V. I., *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Physics Doklady. volume 10. pp. 707-710, 1966
- [9] NIST Speech Group, *Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics*, NIST web site, <http://www.nist.gov/speech/tests/mt/>

- [10] Koehn P., *A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models*, University of Southern California Information Sciences Institute - Los Angeles, 2003
- [11] Zens R. Och F. J. Ney H., *Phrase-Based Statistical Machine Translation*, Proceedings of the 25th Annual German Conference on AI. pp. 18-32, 2002