



Laboratoire RALI
Université de Montréal

Rapport ingénieur

Le Traitement des Invariants dans les Systèmes Statistiques de Traduction Automatique

Jérémy BONNET

`jeremy.bonnet@polytech.univ-nantes.fr`

Version : 1.0

19 août 2004

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

Ecole Polytechnique de l'Université de Nantes

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. Une seconde démarche consiste à modifier directement le code source de l'opération de traduction (appelée *décodage*) sans modifier le corpus d'entraînement. 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. One second step consists in directly modifying the source code of the translation operation (called *decoding*) without modifying the training corpus. 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 Avant-propos	8
1.1 Présentation du document	8
1.2 Glossaire et abréviations	9
2 Présentation du RALI	10
2.1 Historique	10
2.2 Domaines d'étude	10
2.3 Organisation	11
3 Bibliographie	12
3.1 Brève introduction à la MT	12
3.2 Introduction à la SMT	13
3.2.1 Canal bruité	13
3.2.2 Le modèle de langue	15
3.2.3 Les modèles de traduction	15
3.2.4 Décodage	18
3.3 Évaluation de la traduction	19
3.3.1 WER et SER	20
3.3.2 BLEU et NIST	21

4	Présentation du sujet et de l'existant	22
4.1	Notions de mots inconnus et d'entités nommées	22
4.2	Gestion actuelle en SMT	23
4.3	Objectif du stage	24
4.4	Outils du RALI	24
4.4.1	Corpus d'entraînement	24
4.4.2	Architecture classique de Nießen	24
4.4.3	Système de Repérage d'Entités Nommées	25
4.4.4	Évaluation de la traduction	25
5	Démarches adoptées	27
5.1	Pré et post traitements	27
5.2	Niveau du décodeur	29
6	Réalisation	30
6.1	Choix du langage	30
6.2	Description des scripts	31
6.2.1	Évaluation du système de REN	31
6.2.2	Pré et post traitements	31
6.2.3	Modification du décodeur	32
7	Résultats	35
7.1	REN	35
7.2	Première approche : pré et post traitements	36
7.3	Seconde approche : modification du décodeur	37
8	Difficultés rencontrées	39
8.1	Découverte du monde de la linguistique	39
8.2	Repérage d'Entités Nommées	39
8.3	Décodeur Pharaoh	40
9	Conclusions	41
9.1	Bilan de l'étude	41
9.2	Conclusion générale	42

Liste des tableaux

3.1	Exemple d'extrait d'un modèle de traduction.	16
3.2	Exemple de phrases à évaluer.	20
3.3	Résultats de l'évaluation sur les phrases du tableau 3.2.	21
4.1	Exemple de SMT d'une phrase comprenant un mot inconnu.	23
4.2	Exemple de SMT d'une phrase ne comprenant pas de mot inconnu.	24
4.3	Exemple d'étiquetage Part-Of-Speech	26
5.1	Extrait du modèle de traduction entraîné sur un corpus prétraité.	27
7.1	Résultats de l'évaluation de la tâche de repérage d'entités nommées.	35
7.2	Comparaison architecture classique et première approche.	36
7.3	Exemple d'amélioration de la qualité de la traduction.	36
7.4	Exemple de problème de la première approche.	37
7.5	Comparaison des scores des deux approches.	38

Chapitre 1

Avant-propos

1.1 Présentation du document

Ce document est le rapport final du stage que j'ai effectué au sein du Laboratoire RALI, de février à août 2004. Cette période de six mois termine mon cycle d'ingénieur informatique à l'École Polytechnique de l'Université de Nantes.

J'ai été encadré par le professeur Guy Lapalme travaillant sur le résumé automatique et l'extraction d'information et co-encadré par le professeur Philippe Langlais travaillant sur les modèles probabilistes.

Le travail ici porte sur le système de traduction automatique statistique développé par Philippe Langlais (professeur) et George Foster (chercheur).

Après une brève présentation du laboratoire, le document présente les notions indispensables à la conduite de ce stage, à savoir la théorie de la traduction automatique statistique ainsi que différentes métriques d'évaluation de la qualité des traductions. Nous pouvons alors présenter le sujet et les objectifs du stage, ainsi que l'existant comme base de départ. Nous distinguons ensuite deux approches différentes pour atteindre cet objectif dont nous détaillons la réalisation. Dans une dernière partie, nous présentons les résultats de ces deux méthodes et explicitons les difficultés rencontrées lors de ce stage.

1.2 Glossaire et abréviations

Nous expliquons ici les termes spécifiques utilisés dans ce rapport.

BLEU	<i>BiLingual Evaluation Understudy</i> , métrique d'évaluation de qualité des traductions proposée par [Papineni, 2002] et se basant sur les co-occurrences de uni- bi- tri- et quadrigrammes.
GATE	<i>General Architecture for Text Engineering</i> , plate-forme développée en JAVA par l'Université de Sheffield et offrant de nombreuses possibilités en TALN, tel que le REN par exemple.
NIST	<i>National Institute of Standards and Technology</i> , groupe d'étude travaillant entre autres sur le langage et proposant une métrique d'évaluation de qualité des traductions proche de BLEU.
Perl	Langage de programmation de haut niveau étant très adapté aux systèmes de TALN (expressions régulières).
POS	<i>Part-Of-Speech</i> , ensemble d'étiquettes grammaticales qui permet d'étiqueter un texte.
REN ou NER	<i>Repérage d'Entités Nommées</i> ou <i>Named Entity Recognition</i> , tâche qui identifie les noms de personnes, de lieux et d'organisation dans un texte.
SER	<i>Sentence Error Rate</i> , métrique d'évaluation de qualité des traductions se basant sur les phrases.
SMT	<i>Statistical Machine Translation</i> , système de traduction automatique probabiliste.
TA ou MT	<i>Traduction Automatique</i> ou <i>Machine Translation</i> , terme général qui désigne un système de traduction automatique.
TALN	<i>Traitement Automatique de la Langue Naturelle</i> , domaine d'étude qui englobe tous les traitements sur la langue que l'homme cherche à informatiser.
WER	<i>Word Error Rate</i> , métrique d'évaluation de qualité des traductions se basant sur les mots.

Chapitre 2

Présentation du RALI

2.1 Historique

Le laboratoire de Recherche Appliquée en Linguistique Informatique (RALI) a été fondé en 1997 lorsque le Département d'Informatique et de Recherche Opérationnelle (DIRO) de l'Université de Montréal a obtenu du Ministère de l'industrie du gouvernement canadien l'impartition du programme de recherche en Traduction Assistée par Ordinateur (TAO) poursuivi depuis 1985 au Centre d'Innovation en Technologie de l'Information (CITI). Sept chercheurs du CITI se sont joints à deux professeurs du DIRO avec une longue expérience dans le traitement automatique de la langue naturelle. Il est le plus grand laboratoire au Canada dans le domaine du traitement de la langue naturelle et le seul avec une activité bien affirmée en traduction automatique ou assistée.

2.2 Domaines d'étude

Le RALI poursuit un programme de recherche vigoureux dans les domaines de la traduction assistée par ordinateur, la recherche d'information et le traitement automatique de la langue. Parmi les différentes réalisations du laboratoire, nous pouvons citer notamment *TransType*, un système d'aide à la traduction en temps réel, *Reacc*¹ permettant de restaurer automatiquement les accents dans les textes français, ou encore *SILC*² qui détermine automatiquement la langue dans laquelle un document est écrit, de même que le

¹<http://www-rali.iro.umontreal.ca/Reacc/>

²<http://www-rali.iro.umontreal.ca/SILC/>

jeu de caractères employé. Autant par les systèmes réalisés au laboratoire que par les publications de ses membres, le RALI est actuellement reconnu au niveau de la communauté internationale de la linguistique informatique.

2.3 Organisation

De nombreux changements d'organisation ont eu lieu depuis la fondation du laboratoire. Ces modifications témoignent du dynamisme de ses membres pour assurer le financement de l'équipe, que ce soit par des subventions de recherches, des contrats, des ventes de logiciels et même la mise sur pied d'un service payant sur le web³. Le RALI compte actuellement trois professeurs (Philippe Langlais, Guy Lapalme et Jian-Yun Nie), six chercheurs à temps plein et une vingtaine d'étudiants gradués. La coordination de toute l'équipe est assurée par Elliott Macklovitch.

³TSRali.com

Chapitre 3

Bibliographie

Afin de mener à bien ce stage, il est indispensable de faire point sur la traduction automatique, puis sur le cas de la SMT qui nous intéresse. Enfin, nous présentons les différentes métriques d'évaluation de qualité des traductions que nous avons mis en jeu.

3.1 Brève introduction à la MT

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. Selon une étude menée par *Equipe Consortium Ltd*, le marché mondial de la traduction s'est élevé à plus de 2 milliards d'euros en 1998 et aurait doublé en 2000. Les besoins majeurs de la traduction automatique 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 1960¹. Il existe actuellement entre autres la traduc-

¹Pour plus d'informations sur l'histoire de la traduction automatique, voir John Chandioux, président John Chandioux experts-conseils inc., <http://www.univ-tlse2.fr/gril/TAL/TRAD/TRADAUTO1.htm>

tion 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.

3.2 Introduction à la SMT

Nous allons présenter la traduction statistique en introduisant le concept de canal bruité sur lequel elle repose, puis en explicitant les différents modèles mis en jeu à travers ce canal. Enfin, nous commenterons succinctement le problème complexe du décodage.

3.2.1 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 3.1).

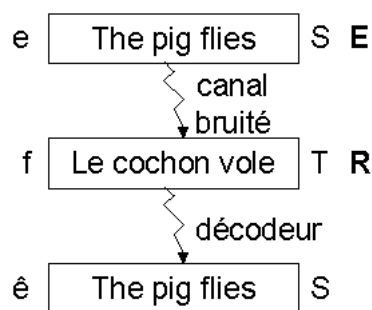


Fig. 3.1. Illustration du canal bruité. L'anglais est ici le langage source du canal et le français le langage cible.

Le but pour R est de retrouver la phrase source à partir de la phrase reçue et 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 \hat{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 et J le nombre de mots de la phrase française. De manière plus formelle :

$$\hat{e} = \arg \max_e [P(e^I|f^J)] \quad (3.1)$$

D'après le théorème de Bayes :

$$P(e^I|f^J) = \frac{P(f^J|e^I) \times P(e^I)}{P(f^J)} \quad (3.2)$$

Comme le dénominateur de l'équation 3.2 est indépendant de e^I , la maximisation devient alors :

$$\hat{e} = \arg \max_e P(e^I|f^J) = \arg \max_e P(e^I) \times P(f^J|e^I) \quad (3.3)$$

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 3.3 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.

3.2.2 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 :

$$\sum_i P(e^i) = 1 \quad (3.4)$$

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 :

$$P(e^I) = \prod_{i=1}^I P(w_i | \underbrace{w_1 \cdots w_{i-1}}_h) \quad (3.5)$$

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* :

$$P(e^I) \simeq P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1w_2)P(w_4|w_2w_3) \cdots P(w_I|w_{I-2}w_{I-1}) \quad (3.6)$$

3.2.3 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 phrases². 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 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 3.1 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. 3.1. 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 $A(e, f)$.

Modèle IBM1

On cherche à modéliser $P(F = f^J | E = e^I)$ où E est l'ensemble des phrases anglaises et F l'ensemble des phrases françaises. $e^I = e_1 \cdots e_I$ et $f^J = f_1 \cdots f_J$ sont deux phrases

²La 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.

particulières de E et F . $A(e^I, f^J)$ est l'ensemble des alignements liant une phrase anglaise donné à une phrase française. On note $P(F = f^J, A = a | E = e^I)$ la probabilité jointe de f^J et d'un alignement particulier a . Alors on a :

$$P(f^J | e^I) = \sum_a P(f^j, a | e^I) \quad (3.7)$$

Pour modéliser cette équation, il y a trois paramètres à estimer :

1. 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 f_j 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 .
3. Choisir un mot français f_j sachant cette position et toutes les autres informations.

Dans IBM1, 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 ($I+1$ car le mot NULL est considéré). De plus, si l'on somme tous les alignements possibles, alors [Brown et al., 1993] ont démontré que la probabilité d'une phrase f sachant une phrase e est :

$$P(f^J | e^I) = \frac{\epsilon}{(I + 1)^J} \cdot \prod_{j=1}^J \sum_{i=0}^I t(f_j | e_i) \quad (3.8)$$

où ϵ désigne la probabilité $P(J | e^I)$ et où $t(f_j | e_i)$ est la probabilité de transfert ou la probabilité lexicale, c'est-à-dire la probabilité que le mot e_i génère le mot français f_j .

Chaque mot cible est généré en consultant les probabilités de transfert de chaque mot source vers ce mot cible. IBM1 possède la propriété que l'on peut calculer $p(f | e)$ de manière exacte et efficace.

Autres modèles

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 $P(f|e)$.

3.2.4 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 3.2).

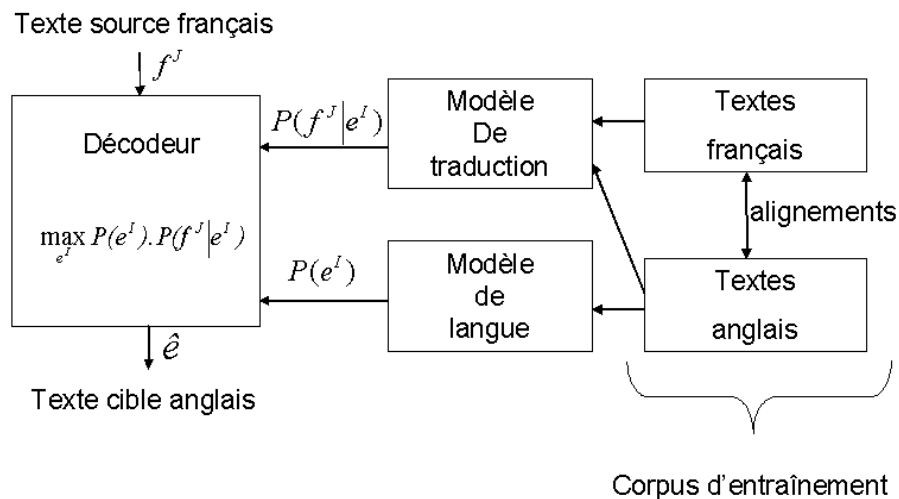


Fig. 3.2. L'architecture de la traduction probabiliste [Nießen et al., 1998].

Revenons à l'équation 3.2 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 heuristiques qui permettent de diminuer l'espace de recherche et donc de baisser 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.

3.3 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. Mais le problème principal de l'évaluation humaine est le temps qu'elle nécessite. Elle sera donc plus utilisée dans l'évaluation de systèmes stables. Cependant, il existe un besoin d'une évaluation plus rapide, même au détriment de la précision, pour évaluer les systèmes en cours de développement afin de vérifier très vite la validité ou non d'une hypothèse.

Nous allons présenter deux méthodes d'évaluation largement utilisées en traduction automatique : WER et SER d'une part et NIST et BLEU d'autre part. Ces métriques sont indépendantes de la paire de langues étudiée (dans la limite de l'existence de la notion de mot). L'idée est de comparer une ou plusieurs traductions automatiques à une traduction de référence, généralement réalisée par l'homme. L'acuité de ces métriques est discutable, mais les valeurs moyennes, mesurées sur de gros corpus de test, sont généralement suffisantes pour comparer plusieurs approches.

3.3.1 WER et SER

Ces métriques permettent d'évaluer la performance en terme de taux d'erreur mesurée à différents niveaux : au niveau de la phrase pour SER (Sentence Error Rate) et au niveau des mots pour WER (Word Error Rate).

SER mesure le pourcentage de phrases pour lesquelles la traduction n'est pas exactement celle de référence. Cette méthode est sévère car une traduction peut être bonne sans être identique à la référence. Une alternative pour pallier à ce problème est de considérer plusieurs traductions de référence.

WER se calcule par une distance de Levenstein qui comptabilise le nombre minimal d'opérations à effectuer pour passer de la traduction automatique à celle de référence. On considère trois opérations de base de même poids : l'insertion, la suppression et la substitution.

Dans l'exemple suivant (tableau 3.2), 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.

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

Tab. 3.2. Exemple de phrases à évaluer.

Les résultats sont présentés dans le tableau 3.3. 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.

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

Tab. 3.3. Résultats de l'évaluation sur les phrases du tableau 3.2.

3.3.2 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 et de NIST 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. 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 .

Chapitre 4

Présentation du sujet et de l'existant

Dans ce chapitre, nous introduisons les notions de mots inconnus et d'entités nommées nécessaires à la compréhension du sujet. Nous présentons ensuite l'existant, c'est-à-dire la gestion actuelle de ces entités nommées en SMT, ce qui nous amène à la formulation de l'objectif du stage. Dans un dernier point, nous détaillons les outils utilisés et mis à disposition par le RALI.

4.1 Notions de mots inconnus et d'entités nommées

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'il 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. Un traitement peut

être effectué en amont pour satisfaire cette hypothèse, mais ce n'est pas le sujet ici. 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).

4.2 Gestion actuelle en SMT

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. Dans cette approche, 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.

Prenons un exemple afin de mieux comprendre le phénomène (tableau 4.1).

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

Dans cet exemple, non seulement l'entité nommée a complètement disparu de la cible - ce qui est très gênant au niveau de la sémantique - mais la structure de la sortie est modifiée. Évidemment, le mot *vendée* ne fait pas partie du corpus d'entraînement. Pour bien réaliser l'impact du phénomène, considérons l'exemple similaire du tableau 4.2.

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

Ici, le décodeur ne rencontre pas de mot inconnu et donne donc une traduction de bien meilleure qualité. Cette non gestion des mots inconnus donne en SMT de mauvais résultats. Le problème se 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.

4.3 Objectif du stage

Gérer les entités nommées est un problème en traduction statistique. L'objectif ici est de considérer ce phénomène en expérimentant des techniques qui conservent ces invariants de traduction dans le cadre du système de traduction probabiliste du RALI afin d'améliorer la qualité de la traduction. Cet objectif 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.

4.4 Outils du RALI

4.4.1 Corpus d'entraînement

Tout d'abord, le RALI utilise de nombreux bitextes alignés permettant d'entraîner les différents modèles. Le corpus utilisé ici est le Hansard de 1994 qui traite des débats parlementaires canadiens.

4.4.2 Architecture classique de Nießen

Le RALI dispose d'un certain nombre d'outils permettant de mettre en oeuvre la traduction probabiliste :

- *ibm* est un ensemble de scripts *csH* et *c++* écrits par George Foster et Philippe Langlais permettant d’entraîner des modèles de traduction IBM1 et IBM2 à partir d’un bitexte aligné.
- *glm* est aussi un ensemble de scripts *csH* et *c++* écrits par les mêmes personnes et permettent d’entraîner des modèles de langue à partir d’un monotexte d’entraînement.
- *fe-translate* est un décodeur basé sur l’architecture de [Nießen et al., 1998] écrit en *c++* par Philippe Langlais permettant de réaliser réellement la traduction d’un texte source en utilisant les modèles entraînés par les deux scripts précédents.

A l’aide de ces trois outils, nous sommes capables d’implémenter totalement l’architecture classique de la traduction statistique.

4.4.3 Système de Repérage d’Entités Nommées

Le RALI a développé un outil (*rالي-tag*) permettant de réaliser l’étiquetage Part-Of-Speech sur un texte en entrée. Concrètement, cet outil associe à chaque mot une étiquette grammaticale du type *nom commun*, *adverbe*, *adjectif* ou encore *nom propre*. Cette dernière étiquette nous permet de repérer les entités nommées. Le REN est donc vu ici comme une sous-tâche de l’étiquetage POS. Le tableau 4.3 présente un extrait de sortie de *rالي-tag*.

Dans cet exemple, on voit que le mot *canada* s’est vu associé l’étiquette *NomP* ; on va donc considérer ce mot comme entité nommée.

4.4.4 Évaluation de la traduction

Un script permettant d’évaluer la traduction d’un texte en fonction d’une traduction de référence est aussi disponible. Les scores calculés sont le WER, le SER, le BLEU et le NIST. C’est donc à l’aide de ce script que nous pouvons évaluer les apports de nos systèmes.

Mot	Tag
le	: Dete-dart-ddef-masc-sing
canada	: NomP-masc-sing-pgeo
est	: Verb-indpré-sing-p3
un	: Dete-dart-dind-masc-sing
acteur	: NomC-masc-sing
dynamique	: AdjQ-femi-sing
sur	: Prep
la	: Dete-dart-ddef-femi-sing
scène	: NomC-femi-sing
de	: Prep
l'	: Dete-dart-ddef-femi-sing
économie	: NomC-femi-sing
mondiale	: AdjQ-femi-sing
.	: Punc-pcst

Tab. 4.3. Exemple d'étiquetage Part-Of-Speech

Chapitre 5

Démarches adoptées

Deux démarches bien différentes ont été utilisées pour tenter de conserver les invariants de traduction. Une première approche consiste en la modification du corpus d'entraînement afin d'*apprendre* au système à ne pas modifier les entités nommées, alors que le principe de la seconde est d'agir directement sur le décodeur pour réaliser cette tâche.

5.1 Pré et post traitements

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'approche préconisée ici ne consiste pas en la ré-écriture des modèles, mais plutôt en leur ré-entraînement sur un corpus modifié. L'idée est de rechercher les entités nommées du corpus d'entraînement puis de les remplacer par un terme spécial. Ceci constitue ce que nous appellerons le *prétraitement*. Les modèles de langue et de traduction sont ré-entraînés sur ce corpus prétraité afin d'incorporer ce nouveau terme.

Le tableau 5.1 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. 5.1. Extrait du modèle de traduction entraîné sur un corpus prétraité.

En ce qui concerne le décodage, on réalise le même prétraitement (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 modélisée par la figure 5.1 comme une variante de la figure 3.2.

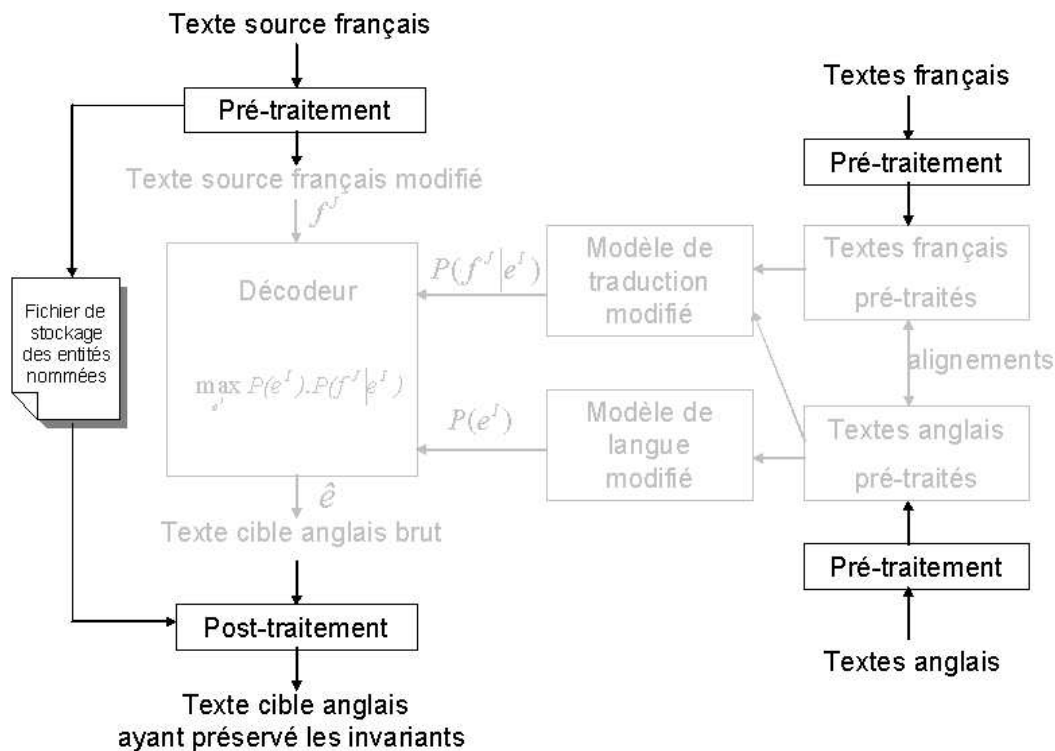


Fig. 5.1. L'architecture de la traduction probabiliste [Nießen et al., 1998] présentée à la figure 3.2 mais modifiée pour conserver les invariants de traduction.

Elle 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...).

5.2 Niveau du décodeur

Au vu des résultats encourageants (présentés plus loin) de la première approche, l'idée de s'intéresser directement au décodeur est apparue. On ne réalise ici plus aucune modification en amont ni en aval de la chaîne de traitements, mais on agit directement sur le décodeur de manière à ce que tout mot inconnu en entrée reste inchangé en sortie.

Le décodeur *fe-translate* utilisé au RALI est basé sur le papier de [Nießen et al., 1998]. Il s'appuie sur le modèle de traduction IBM2. Le principe est le suivant : on avance le long du texte cible (search inversé), et on essaie de couvrir progressivement les positions sources. On suppose également qu'un mot cible e_i est aligné avec l mots sources consécutifs. On appelle ce nombre de mots source couverts par un mot cible la *fertilité*. Par exemple, le mot cible anglais *update* a une fertilité de 3 lorsqu'il est la traduction de *mise à jour* en français. On part de toutes les hypothèses possibles considérant tous les mots, à toutes les positions et ayant toutes les fertilités. Les hypothèses les moins probables sont supprimées. On effectue le calcul des scores des hypothèses restantes pour finalement proposer la meilleure en sortie.

Certaines modifications ont été apportées dans cette démarche. Dans un premier temps, on effectue une recherche de tous les mots inconnus des modèles. Une fois ces mots repérés, on supprime toutes les hypothèses proposant une traduction pour ces mots. On génère ensuite la phrase cible où les mots inconnus sont manquants. Enfin, on remplace les mots manquants à la meilleure place en fonction du modèle de langue.

Chapitre 6

Réalisation

Ce chapitre présente l'implémentation concrète des deux approches précédentes. Nous commençons par justifier le choix de notre langage de programmation, puis nous décrivons les scripts que nous avons été amenés à coder.

6.1 Choix du langage

En TALN, certains langages s'avèrent beaucoup plus adéquats. On préfère naturellement les langages qui facilitent le traitement des fichiers textes et des chaînes de caractères. Pour cela, nous nous sommes tournés vers le *Perl* qui possède un certain nombre d'avantages. Le principal atout est la puissance des expressions régulières pour manipuler les chaînes de caractères, ce qui fait de *Perl* le langage de prédilection en linguistique. Outre cet aspect, *Perl* permet la programmation fonctionnelle aussi bien qu'orientée objet ; c'est un langage interprété (licence *GNU GPL*) pré-compilé à l'exécution disponible sur 87 plateformes, ce qui assure une portabilité maximale. Une autre richesse de *Perl* est le nombre impressionnant de modules¹ qui regroupent des fonctionnalités telles que la gestion de documents XML, le chiffrement ou encore les services réseaux. Tous ces modules facilitent grandement la tâche du programmeur. Pour toutes ces raisons, et aussi du fait que ce langage est largement utilisé au RALI, notre choix s'est naturellement tourné vers le *Perl*.

¹Modules disponibles sur <http://www.cpan.org>

6.2 Description des scripts

6.2.1 Évaluation du système de REN

Sur la figure 5.1, 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,

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

Ce programme a été codé en *Perl*. Il a ensuite été exécuté sur un document de test extrait du corpus d'entraînement. Il a fallu au préalable réaliser le repérage manuellement pour obtenir la référence.

6.2.2 Pré et post traitements

L'implémentation de la première approche passe par la programmation et l'utilisation de nombreux scripts. Une fois l'architecture classique fonctionnelle, reste à implémenter la modification de cette architecture, c'est-à-dire les différents pré- et post-traitements. Tout d'abord, en ce qui concerne le corpus d'entraînement, son prétraitement comprend deux opérations :

1. Le repérage des entités nommées.
2. Le remplacement de celles-ci par un terme spécial (n'appartient pas déjà au corpus d'entraînement).

Le prétraitement du texte source, quant à lui, comprend trois opérations bien distinctes :

1. Le repérage des entités nommées.
2. Le stockage de ces entités nommées dans un fichier (afin de pouvoir les replacer après la traduction).
3. Le remplacement de celles-ci par un terme spécial (n'appartenant pas déjà au corpus d'entraînement).

Enfin, le post-traitement comprend deux opérations :

1. La recherche dans le document cible brut des termes spéciaux correspondant aux entités nommées.
2. Le remplacement de ces termes spéciaux par les véritables entités nommées stockées dans le fichier.

Ces trois scripts ont été réalisés en *Perl* et appellent l'étiqueteur POS du RALI (*rality*) pour effectuer le repérage des entités nommées.

6.2.3 Modification du décodeur

Le décodeur *fe-translate* est écrit en *c++* afin de minimiser le temps d'exécution. Voici son algorithme simplifié en pseudo-code :

```
Entrée : une phrase française
Sélectionner la longueur maximale de la phrase cible
Sélectionner le vocabulaire actif (selon l'entrée et le modèle de traduction)
Rechercher les mots inconnus dans la source
for all mot du vocabulaire actif do
  for all position dans la phrase source do
    for all fertilité do
      // construction de l'ensemble des hypothèses
```

```

        Hyp ← Hyp ∪ h
    end for
end for
end for
// élagage des mauvaises solutions
for all h ∈ Hyp do
    if evalRapide(h) ≥ seuil then
        if aucun mot de h ne vient d'un mot inconnu then
            aliveHyp ← aliveHyp ∪ h
        end if
    end if
end for
// recherche de la meilleure solution
for all h ∈ aliveHyp do
    if evalTotal(h) ≥ evalTotal(bestHyp) then
        bestHyp ← h
    end if
end for
// Replacement des mots inconnus
for all mot inconnu mi do
    for all position p dans bestHyp do
        currHyp ← makeHyp(bestHyp, mi, p)
        if evalRapide(output) ≥ evalRapidecurrHyp then
            output ← currHyp
        end if
    end for
end for
return output

```

où : *evalRapide*(*h*) retourne le score de l'hypothèse *h* calculé rapidement en fonction du modèle de langue ; et *evalTotal*(*h*) retourne la probabilité de l'hypothèse *h* en fonction des paramètres du modèle de langue et du modèle de traduction.

La complexité de l'algorithme est : $\mathcal{O}(I_{max}^2 \cdot J^3 \cdot |\mathcal{E}|^2)$ où $|\mathcal{E}|$ 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 (contraintes) qui permettent d'accélérer l'algorithme pour une complexité finale en $\mathcal{O}(J^2 \times I_{max} \times |\mathcal{E}|)$.

Chapitre 7

Résultats

Nous présentons ici les résultats de nos différents traitements. Tout d'abord, nous décrivons les résultats du repérage des entités nommées de l'étiqueteur Part-Of-Speech du RALI. Nous analysons ensuite les scores de traduction de nos deux approches.

7.1 REN

L'évaluation de la tâche de repérage des entités nommées de l'étiqueteur Part-Of-Speech du RALI s'est faite sur un document extrait de corpus d'entraînement contenant approximativement 45 000 mots, 2 300 phrases et quelques 1 000 entités nommées. Les résultats en sortie du script sont présentés dans le tableau 7.1.

Précision	Rappel	F-mesure
93.71%	93.39%	93.55%

Tab. 7.1. 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 l'étiqueteur Part-Of-Speech du RALI comme notre système de repérage d'entités nommées.

7.2 Première approche : pré et post traitements

L'évaluation des gains des deux systèmes que l'on propose se fait par la comparaison des scores de la traduction utilisant l'architecture classique avec les scores de la traduction modifiée. L'entraînement se fait sur le corpus Hansard de 1994 constitué de 1.6 millions de paires de phrases, 30 millions de mots français et anglais et un vocabulaire de 90 000 mots différent dans chaque langue. Le document de test est un extrait du Hansard de 2003 contenant 2 000 phrases et environ 1 500 entités nommées. Les résultats sont présentés dans le tableau 7.2.

Architecture	WER	SER	NIST	BLEU
classique	68.96%	97.21%	1.652	0.059
première approche	67.28%	96.69%	1.766	0.068

Tab. 7.2. Comparaison architecture classique et première approche.

On observe que tous les scores évoluent de la même manière, à savoir dans le sens de l'amélioration des scores de traduction. Afin de mieux comprendre cette amélioration, reprenons l'exemple du tableau 4.1 et observons les différences dans le tableau 7.3.

SOURCE	des agriculteurs de la vendée sont venus à mon bureau la semaine dernière .
CLASSIQUE	some farmers to come to my office last week .
MODIFIEE	some farmers of vendée came to my office last week .

Tab. 7.3. Exemple d'amélioration de la qualité de la traduction.

Avec ce système, on diminue le nombre de mots inconnus des modèles à l'intérieur du document source. Ainsi, cela permet de diminuer l'indéterminisme au niveau du décodeur et donc, de proposer une traduction syntaxiquement de bien meilleure qualité. Le second point expliquant cette amélioration vient du fait que l'entité nommée *vendée* est conservée, ce qui est très bénéfique au niveau de la sémantique de la traduction. Pour ces deux raisons principales, nous avons amélioré la qualité du système de traduction en proposant

une meilleure gestion des invariants de traduction.

Cependant, on observe quelques problèmes, notamment lors de la traduction d'un nom de personne dans un contexte particulier. En effet, lorsque le prénom d'une personne est connu des modèles et que son nom de famille ne l'est pas, alors on voit apparaître en sortie le nom le plus fréquent dans le corpus d'entraînement associé au prénom. Ce problème est illustré dans le tableau 7.4.

SOURCE	c' est bien mr. peter .
MODIFIEE	it is mr. peter .
SOURCE	c' est bien mr. peter barrett .
MODIFIEE	it is mr. peter milliken barrett.

Tab. 7.4. Exemple de problème de la première approche.

Ici, *milliken* est le nom de famille du corpus d'entraînement le plus souvent associé au prénom *peter*. Cette situation est due à la très grande fréquence du bigramme *peter milliken* dans le corpus d'entraînement, ce qui entraîne une grande probabilité dans le modèle de langue du mot *milliken* quand le mot précédent est *peter*.

7.3 Seconde approche : modification du décodeur

Le protocole d'évaluation de cette seconde approche est identique au précédent. Nous utilisons les mêmes données d'entraînement ainsi que le même document de test. Les résultats sont présentés dans le tableau 7.5.

Tous les résultats sont encore améliorés, et le gain est important. Notre première approche a démontré que le fait de proposer une meilleure gestion des invariants est bénéfique pour notre système de traduction probabiliste. Partant de ce constat, nous avons complètement changé de méthode en modifiant directement le décodeur de [Nießen et al., 1998] afin de conserver ces invariants. De plus, nous avons élargi notre notion d'invariants puisque ne sont considérés comme tels non plus seulement les entités nommées, mais tous

Systeme	WER	SER	NIST	BLEU
classique	68.96%	97.21%	1.652	0.059
première approche	67.28%	96.69%	1.766	0.068
deuxième approche	64.05%	95.48%	4.974	0.195

Tab. 7.5. Comparaison des scores des deux approches.

les mots inconnus des modèles. Le gain en terme de qualité de traduction est alors conséquent.

Chapitre 8

Difficultés rencontrées

Plusieurs difficultés se situant à plusieurs niveaux sont apparues lors de l'avancement de ce stage. Nous commentons ici les principales.

8.1 Découverte du monde de la linguistique

La première difficulté rencontrée s'est faite ressentir dès le début projet. En effet, la découverte d'un domaine auparavant inconnu nécessite un long apprentissage ainsi que de nombreuses lectures sur le sujet afin d'acquérir le vocabulaire et les principes importants. Ainsi, une longue période de documentation sur la traduction statistique, le repérage des entités nommées et un travail de prise en main des outils mis à disposition par le RALI a été d'emblée nécessaire à la bonne conduite de ce stage.

8.2 Repérage d'Entités Nommées

Une autre difficulté a été de trouver un bon système de repérage des entités nommées. En effet, il existe une multitude de systèmes disponibles pour effectuer cette tâche. On distingue les approches par *knowledge engineering* développées par des linguistes des approches basées sur l'apprentissage qui ne nécessitent pas d'experts. Les premières utilisent l'intuition humaine, prennent du temps mais ne requièrent que peu de données d'entraînement. A l'inverse, les secondes ne nécessitent pas d'experts mais plutôt une grande

quantité de données d'entraînement. L'algorithme d'apprentissage peut être basé sur les chaînes de Markov, l'entropie maximale et encore les réseaux de neurones.

Nous avons dans un premier temps utilisé le système GATE¹. Ce système est en fait une plate-forme *JAVA* qui permet de réaliser la tâche de REN basée sur des règles. Cependant, les règles pour le français devaient être écrites² et le programme est très gourmand en mémoire. Après une évaluation rapide des performances, nous nous sommes donc tournés vers l'étiqueteur Part-Of-Speech du RALI, système utilisant les chaînes de Markov et étant entraîné sur des textes français et anglais.

8.3 Décodeur Pharaoh

Étant donné les bons résultats de la première approche, l'idée nous est venue d'agir directement sur le décodeur. Cependant, avant de se lancer dans ce travail fastidieux, nous avons voulu tester un décodeur existant. Ainsi, nous nous sommes tournés vers *Pharaoh*. *Pharaoh* est un décodeur développé par [Koehn, 2003] utilisant les Phrase-Based Models et qui peut prendre en entrée un texte avec un balisage XML afin de forcer une traduction en sortie (ce qui est exactement ce que nous désirons). Il nous a donc fallu prendre en main tous les toolkits nécessaires à l'implémentation de ce décodeur, traduire nos modèles en format compréhensible par *Pharaoh* et enfin évaluer la sortie. Les résultats étant meilleurs que ceux de la première approche, nous avons donc décidé de modifier notre décodeur.

¹General Architecture for Text Engineering : <http://gate.ac.uk>
réalisé par le Natural Language Processing Group de l'Université de Sheffield : <http://nlp.shef.ac.uk/>

²La tâche de REN est souvent dépendante de la langue considérée. Les systèmes indépendants de la langue donnent très souvent de moins bons résultats.

Chapitre 9

Conclusions

9.1 Bilan de l'étude

Dans ce rapport, nous avons présenté le problème posé par les entités nommées dans les systèmes de traduction automatique statistiques. Après avoir explicité les notions indispensables à la résolution du problème de leur conservation, nous explicitons une première démarche visant à améliorer les performances du système. Le principe est de modifier le corpus d'entraînement afin d'*apprendre* au système à conserver les entités nommées. Tous les traitements se font en amont et en aval de l'architecture classique de le SMT, ce qui assure une certaine portabilité.

Les résultats étant très encourageants, nous nous sommes tournés vers une seconde approche consistant à agir directement sur le décodeur. L'objectif maintenant de considérer tout mot inconnu des modèles comme invariants de traduction, sans modifier le corpus d'entraînement. Les résultats obtenus alors sont encore meilleurs que ceux de la première approche.

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.

9.2 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 au métier d'ingénieur.

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. 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
- [2] Knight K., *A statistical machine translation workbook*, USC/ISI - available at <http://www.clsp.jhu.edu/ws99/projects/mt/mt-workbook.htm>, 1999
- [3] 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
- [4] van Rijsbergen, *Information Retrieval*, Butterworths - London. pp. 112-140, 1979
- [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] Koehn P., *A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models*, University of Southern California Information Sciences Institute - Los Angeles, 2003
- [8] Sundheim B., *Overview of results of the MUC-6 evaluation*, *Proceedings of the Sixth Message Understanding Conference (MUC-6)* - Columbia - MorganKaufmann Publishers, 1995