

Université de Montréal

**Amélioration a posteriori de la traduction automatique par
métaheuristique**

par Sébastien Lavoie-Courchesne

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

Mémoire présenté en vue de l'obtention du grade de maîtrise en
informatique

Mars 2016

©, Sébastien Lavoie-Courchesne, 2016

Résumé

La traduction automatique statistique est un domaine très en demande et où les machines sont encore loin de produire des résultats de qualité humaine. La principale méthode utilisée est une traduction linéaire segment par segment d'une phrase, ce qui empêche de changer des parties de la phrase déjà traduites. La recherche pour ce mémoire se base sur l'approche utilisée dans Langlais, Patry et Gotti 2007, qui tente de corriger une traduction complétée en modifiant des segments suivant une fonction à optimiser. Dans un premier temps, l'exploration de nouveaux traits comme un modèle de langue inverse et un modèle de collocation amène une nouvelle dimension à la fonction à optimiser. Dans un second temps, l'utilisation de différentes méta-heuristiques, comme les algorithmes gloutons et gloutons randomisés permet l'exploration plus en profondeur de l'espace de recherche et permet une plus grande amélioration de la fonction objectif.

Mots clés. traduction automatique statistique (SMT), méta-heuristique, recherche locale, collocations, modèle de langue

Table des matières

| | |
|--|-----------|
| Table des matières | 3 |
| Table des figures | 5 |
| Liste des tableaux | 6 |
| 1 Introduction | 8 |
| 2 Traduction Automatique Statistique | 9 |
| 2.1 Début | 9 |
| 2.2 Modèles probabilistes | 10 |
| 2.3 Fonction objectif complexe | 13 |
| 2.4 Création des hypothèses de traduction | 14 |
| 2.5 Vérification de la performance des systèmes de traductions | 15 |
| 3 Métaheuristiques | 17 |
| 3.1 Algorithmes | 17 |
| 3.2 Applications en traduction automatique | 19 |
| 4 Réalisations | 21 |
| 4.1 Chaîne de traitement | 21 |
| 4.2 Algorithmes | 25 |
| 4.3 Transformations | 26 |
| 4.4 Modèles | 27 |
| 5 Résultats | 32 |
| 5.1 Données | 32 |
| 5.2 Présentation des résultats | 32 |
| 5.3 Algorithmes et réglage | 34 |
| 5.4 Modèle de langue inverse | 38 |
| 5.5 Modèle de collocations | 43 |
| 5.6 Retour sur les résultats | 53 |
| 6 Perspectives | 55 |
| 6.1 Métaheuristiques | 55 |
| 6.2 Modèle de langue inverse | 58 |

| | | |
|----------|---|-----------|
| 6.3 | Modèle de collocations | 58 |
| 6.4 | Autres modèles possibles | 59 |
| 6.5 | Autres métriques pour l'analyse | 60 |
| 7 | Conclusion | 62 |
| A | Annexe | 63 |
| A.1 | Analyses manuelles | 63 |
| | Bibliographie | 74 |

Table des figures

| | | |
|---|---|----|
| 1 | Exemple de table utilisée dans un modèle de traduction | 11 |
| 2 | Exemple de table utilisée dans un modèle de langue | 12 |
| 3 | Chaîne de traitement du décodeur par recherche locale a posteriori | 22 |
| 4 | Exemples des transformations possibles | 27 |
| 5 | Différence entre un modèle de langue n -gramme ordinaire et notre modèle de collocations | 29 |
| 6 | Exemple de fichier du modèle de collocations | 30 |
| 7 | Exemple d'amélioration en utilisant le modèle de langue inverse | 40 |
| 8 | Exemple de détérioration en utilisant le modèle de langue inverse | 41 |

Liste des tableaux

| | | |
|----|---|----|
| 1 | Statistiques pour les corpus utilisés pour les tests | 32 |
| 2 | Comparaison des algorithmes et des réglages (newstest2009) . . | 35 |
| 3 | Comparaison des algorithmes et des réglages (newstest2010) . . | 35 |
| 4 | Comparaison des algorithmes et des réglages (newstest2011) . . | 35 |
| 5 | Statistiques pour le corpus newstest2009 avec les algorithmes de base | 36 |
| 6 | Comparaison humaine entre Moses et la recherche locale sans modèle ajouté sur un échantillon aléatoire | 38 |
| 7 | Utilisation du modèle de langue inverse (newstest2009) | 38 |
| 8 | Utilisation du modèle de langue inverse (newstest2010) | 39 |
| 9 | Utilisation du modèle de langue inverse (newstest2011) | 39 |
| 10 | Statistiques pour les différents corpus avec l'utilisation du mo- dèle de langue inverse ou non | 42 |
| 11 | Comparaison humaine entre Moses et la recherche locale avec le modèle de langue inverse sur un échantillon aléatoire | 43 |
| 12 | Statistiques pour les collocations vues dans le corpus europarl . | 44 |
| 13 | Effet de la taille de la fenêtre pour la méthode d'évaluation par compte du modèle de collocations (collc) | 45 |
| 14 | Effet de la distance et de l'ordre pour la méthode d'évaluation par compte du modèle de collocations (collc) | 47 |
| 15 | Effet de la taille de la fenêtre pour la méthode d'évaluation par somme du modèle de collocations (colls) | 48 |
| 16 | Effet de la distance et de l'ordre pour la méthode d'évaluation par somme du modèle de collocations (colls) | 49 |
| 17 | Effet de la taille de la fenêtre pour la méthode d'évaluation par probabilité du modèle de collocations (collp) | 50 |
| 18 | Effet de la distance et de l'ordre pour la méthode d'évaluation par probabilité du modèle de collocations (collp) | 51 |
| 19 | Comparaison humaine entre Moses et la recherche locale avec le modèle de collocations (collp) sur un échantillon aléatoire | 52 |

Liste des algorithmes

| | | |
|---|---|----|
| 1 | Pseudo-code de l'algorithme glouton | 25 |
| 2 | Pseudo-code de l'algorithme glouton randomisé | 26 |

1 Introduction

Le grand domaine de cette recherche est la traduction automatique statistique, un domaine très étudié, mais où il reste encore beaucoup d'améliorations possibles à découvrir. Nous nous penchons particulièrement sur l'amélioration de celle-ci par une recherche locale a posteriori. L'idée de base, soit modifier une phrase après sa traduction pour trouver une meilleure traduction, provient d'un article de Langlais et al. [7]. Ceci a également été exploré dans Monty [10].

Ce mémoire continue donc l'exploration de l'utilisation de la recherche locale pour améliorer la traduction automatique statistique tout en amenant de nouvelles idées dans l'espoir d'en améliorer la qualité.

Un premier chapitre (Chapitre 2) fera un retour rapide sur les méthodes utilisées en traduction automatique statistique, tant à ses débuts qu'après tant d'années de recherches. Le chapitre 3, de son côté, donnera une idée générale de la recherche courante en métaheuristiques, et comment appliquer ce domaine à la traduction automatique statistique. Le troisième chapitre (Chapitre 4) décrira ce qui a été implémenté en détail. La chaîne de traitement y sera détaillée, ainsi que les métaheuristiques, soit une recherche gloutonne et une recherche gloutonne randomisée, et les modèles implémentés, soit un modèle de langue inverse et un modèle de collocations. Le chapitre 5 sera réservé aux résultats obtenus lors des tests effectués et analysera plus en détail les traductions produites. Le dernier chapitre (Chapitre 6) sera réservé à une ouverture vers de nouvelles solutions qui pourraient être envisagées pour continuer dans la direction de l'amélioration a posteriori de la traduction automatique statistique.

2 Traduction Automatique Statistique

Considérant qu'il existe plus d'une vingtaine de langues qui sont parlées par au moins 50 millions de personnes [8], la traduction est un outil nécessaire à la communication mondiale. Malheureusement, la traduction est une tâche requérant beaucoup de temps et d'argent. C'est pourquoi, dans les années 50, certaines recherches ont commencé sur la possibilité d'automatiser la traduction.

2.1 Débuts

Les premiers systèmes de traduction automatique ont commencé par un modèle à base de règles. Cette méthode est souvent dénommée *Rule-Based Machine Translation* (RBMT). L'idée de base est d'utiliser un système de règles pour analyser morphologiquement, syntaxiquement et sémantiquement une phrase à traduire et en produire une traduction. Malheureusement, cette méthode a son lot de défauts :

- les règles sont généralement faites pour une paire de langues précise, ce qui ne permet pas la traduction dans d'autres langues ;
- les règles doivent être écrites par des experts, ce qui prend beaucoup de temps ;
- si une phrase ou section de phrase n'est pas dans la table des règles, alors elle ne peut pas être traduite.

Un premier exemple de l'utilisation de cette méthode a été le système METEO. Ce système, développé par Environnement Canada à partir d'un prototype nommé TAUM-METEO originalement créé par l'Université de Montréal, a été utilisé dans les années 1980 et 1990 pour traduire des bulletins météorologiques produits par ordinateur de l'anglais vers le français et vice-versa. Dans ce cas, la traduction automatique à base de règles était bien adaptée parce que les bulletins, étant produits par un ordinateur, étaient toujours dans le même format et contenaient toujours les mêmes mots.

Un autre exemple de l'utilisation de cette méthode est le système SYSTRAN, qui est d'ailleurs encore utilisé de nos jours. Ce système écrit dans les années 1960 ne traduisait qu'entre le russe et l'anglais au départ. Par la suite, plusieurs autres paires de langues ont été ajoutées pour en avoir une vingtaine en ce moment, et le système a évolué en un système de traduction à base de règles utilisant aussi un modèle probabiliste.

Une autre approche à la traduction automatique a été l'approche par analogies, nommée *Example-Based Machine Translation* (EBMT). Cette méthode, proposée par Nagao [11], utilise le principe d'analogie pour trouver des segments de phrases similaires et en déduire une traduction.

2.2 Modèles probabilistes

En 1993, IBM reprend une idée de Warren Weaver, la traduction automatique statistique. La base de cette méthode est l'utilisation de la probabilité conditionnelle d'une phrase traduite (e) considérant la phrase source (f), $p(e|f)$. Le problème est donc de trouver la phrase traduite parmi toutes les traductions possibles qui maximise cette probabilité :

$$\tilde{e} = \arg \max_{e \in e^*} p(e|f) \quad (1)$$

Nous pouvons ensuite utiliser la loi de Bayes ($p(e|f) = \frac{p(f|e)p(e)}{p(f)}$) pour calculer les probabilités. Étant donné que f ne change pas, et donc $p(f)$ non plus, on peut l'exclure de la maximisation à calculer, ce qui nous laisse à calculer :

$$\tilde{e} = \arg \max_{e \in e^*} p(f|e)p(e) \quad (2)$$

La première probabilité peut être calculée avec un modèle de traduction et la deuxième avec un modèle de langue. Le modèle de traduction construit une table de traduction à partir d'un corpus bilingue entre f et e . Chaque entrée est un segment de phrase de f , un segment de phrase de e et la probabilité de traduire f par e , calculée par exemple par le nombre de fois où cette traduction a été vue dans le corpus d'entraînement. Le modèle de langue construit aussi une table, mais cette fois-ci à partir d'un corpus monolingue. Chaque entrée est donc un segment de phrase de e et la probabilité de retrouver ce segment, calculée encore ici par la fréquence de ce segment dans le corpus d'entraînement. D'un point de vue technique, les tables utilisées pour ces deux modèles sont enregistrées en format texte avec une syntaxe spécifique au modèle et sont souvent compressées pour en diminuer la taille. Les figures 1 et 2 présentent des exemples des tables produites par les modèles de traductions et les modèles de langue respectivement.

```

ce vide est ||| that vacuum can ||| 0.120238 0.000492755 0.0200396 8.5176e-05 2.718
||| 0-0 1-1 2-2 ||| 1 6 1
ce vide est ||| that vacuum is ||| 0.120238 0.00905215 0.0200396 0.0117862 2.718
||| 0-0 1-1 2-2 ||| 1 6 1
ce vide est ||| this gap will ||| 0.120238 0.000141526 0.0200396 0.000174447 2.718
||| 0-0 1-1 2-2 ||| 1 6 1
ce vide ||| a gap ||| 0.000796277 0.000142174 0.00286281 0.000582373 2.718
||| 0-0 1-1 ||| 151 42 1
ce vide ||| the gap ||| 0.00143955 0.000439225 0.0245067 0.00813749 2.718
||| 0-0 1-1 ||| 715 42 2
ce vide ||| the vacuum ||| 0.00601189 0.00659194 0.00286281 0.0220326 2.718
||| 0-0 1-1 ||| 20 42 1
ce vide ||| this gap ||| 0.0743802 0.0131093 0.214286 0.0356488 2.718
||| 0-0 1-1 ||| 121 42 9
ce vide ||| this hole ||| 0.0300595 0.0105263 0.00286281 0.00302676 2.718
||| 0-0 1-1 ||| 4 42 1
ce vide ||| this vacuum ? ||| 0.0601189 0.196745 0.00286281 1.33005e-05 2.718
||| 0-0 1-1 ||| 2 42 1
ce vide ||| this vacuum ||| 0.641454 0.196745 0.168 0.0965207 2.718
||| 0-0 1-1 ||| 11 42 8
ce vide ||| with the void ||| 0.120238 0.00248216 0.00286281 2.63272e-05 2.718
||| 0-1 1-2 ||| 1 42 1

```

FIGURE 1: Exemple de table utilisée dans un modèle de traduction

Les barres verticales séparent les différentes colonnes de la table : segment source, segment cible, scores à utiliser dans la fonction objectif (ici 5), alignement mot-à-mot et comptes d'occurrences.

| | |
|------------|-----------------------------------|
| -0.6930043 | in a report which I |
| -1.280893 | in a report which the |
| -0.9489306 | is a report which has |
| -1.324133 | is a report which will |
| -1.006696 | of a report which I |
| -0.9817029 | of a report which is |
| -1.184751 | on a report which will |
| -0.9044265 | presented a report which we |
| -0.5367818 | produced a report which is |
| -0.8576636 | support a report which proposes |
| -0.4531554 | that a report which is |
| -0.9118987 | to a report which , |
| -1.032808 | with a report which I |
| -0.5731179 | with a report which is |
| -0.7085655 | 's excellent report which is |
| -0.9135105 | an excellent report which is |
| -0.8666081 | his excellent report which , |
| -1.224352 | the excellent report which she |

FIGURE 2: Exemple de table utilisée dans un modèle de langue

Chaque ligne donne le score en premier et le segment cible ensuite.

Avec ces deux modèles nous pouvons donc commencer à traduire. La méthode naïve pour créer une traduction avec cette méthode est de traduire chacun des symboles¹ un par un, avec un modèle dit unigramme (1-gramme). De cette manière nous obtenons maintenant une nouvelle équation pour la probabilité d'une traduction :

$$p(f|e) \approx \prod_{i=0}^n p(f_i|e_i) \quad (3)$$

Nous pouvons maintenant générer des traductions en utilisant la table de traduction pour trouver les symboles de la langue cible qui maximisent cette probabilité. Malheureusement, cette méthode de création de traductions ne permet pas de résoudre certains problèmes, notamment celui des termes polysémiques. Prenons par exemple la phrase «Cette glace est délicieuse.». Si nous avons les traductions pour glace «mirror» et «ice cream» dans notre table de traductions, avec des probabilités de 0.5 et 0.3 respectivement, cette méthode retournera la traduction «This mirror is delicious.», ce qui n'est pas la bonne traduction ici.

1. Un symbole définit tout ce qui peut être considéré comme un mot, un regroupement de mots comme une locution figée (par exemple «pomme de terre») ou un symbole de ponctuation.

Pour remédier à ce problème, nous utilisons un modèle utilisant plus de symboles, et en approximant la probabilité pour une phrase complète à l'aide des chaînes de Markov, par exemple ici en utilisant un bigramme :

$$p(f|e) \approx \prod_{i=0}^n p(f_i, f_{i-1} | e_i, e_{i-1}) \quad (4)$$

À partir de ceci, nous pouvons maintenant avoir un peu plus d'information. En se servant de la même idée que la génération d'une traduction avec un modèle unigramme, on peut créer une traduction mot par mot, mais en ayant des probabilités plus précises et prenant en compte les symboles précédents, ce qui donne une meilleure idée du contexte de la phrase, permettant ainsi d'améliorer la sélection d'une traduction pour un mot. De nos jours, les modèles de traduction statistiques utilisent entre 5 et 7 symboles afin d'avoir le plus d'information possible par rapport aux autres segments dans la phrase, sans trop utiliser de ressources pour effectuer la traduction.

Cette idée en amène une autre, celle de créer la traduction à partir de segments (suite de symboles) au lieu de la créer symbole par symbole. C'est la méthode qui est maintenant la plus utilisée dans les systèmes de traduction. Ce type de traduction statistique, nommé *Phrase-Based Machine Translation*, a été présenté dans Koehn, Och, and Marcu [5].

2.3 Fonction objectif complexe

Cette nouvelle méthode de traduction à base de segments est celle qui est utilisée dans les logiciels Pharaoh [4] et son successeur Moses [6]. L'idée de base est la même que pour la génération mot à mot, mais on travaille maintenant avec des segments de plusieurs symboles, généralement de 3 à 7 symboles par segments. Cette méthode permet de traduire des locutions figées beaucoup plus facilement, ainsi de que capturer une meilleure information sur le contexte.

En plus d'utiliser un modèle de traduction plus complexe que précédemment, Pharaoh remplace la fonction à maximiser pour trouver la traduction. On ajoute à la fonction originale (Équation 2) une série de plusieurs autres fonctions pour capturer un peu plus d'information. Cette nouvelle fonction à optimiser, qui n'est plus probabiliste, contient plusieurs éléments, typiquement le modèle de traduction ($p_{tm}(f|e)$), le modèle de langue ($p_{lm}(e)$), un modèle de réordonnancement ($p_d(f, e)$) pour faire quelques inversions

qui surviennent souvent entre certaines paires de langues, une pénalité lexicale (p_w) et plusieurs autres. La fonction objectif prend donc la forme d'un produit pondéré des différentes fonctions présentes, mais est généralement calculé comme la somme pondérée des logarithmes des différentes fonctions présentes :

$$p(f|e) = \prod_{i \in \text{fonctions}} p_i(f|e)^{\lambda_i} \quad (5)$$

$$\log(p(f|e)) = \sum_{i \in \text{fonctions}} \lambda_i \log(p_i(f|e)) \quad (6)$$

C'est ici qu'on voit survenir le premier problème : comment choisir les poids pour chaque fonction pour obtenir la meilleure traduction ? Plusieurs recherches ont été effectuées dans ce domaine, et plusieurs outils existent pour tenter de trouver les meilleurs poids possibles (par exemple : Minimum Error Rate Training [12] (MERT)). Une mauvaise sélection des poids entraîne généralement des mauvaises traductions et c'est donc une étape importante dans la chaîne de traitement pour la traduction automatique statistique.

L'avantage majeur de l'utilisation d'une telle fonction provient du fait que chacun des modèles capture une partie d'information différentes des autres. Ainsi, la combinaison de ces différents modèles nous permet d'obtenir le plus d'information possible afin de produire une meilleure traduction.

2.4 Création des hypothèses de traduction

Anciennement, la création des hypothèses de traduction (la liste de e^* dans l'équation 1) se faisait d'un bout de la traduction à un autre, et on créait généralement très peu d'hypothèses à cause de la complexité algorithmique que la production engendre. Maintenant, la création se fait du début de la phrase à la fin, mais nous conservons plusieurs hypothèses possibles en parallèle. Cette technique s'appelle un *beam-search* ou recherche à faisceau. On garde plusieurs hypothèses possibles en utilisant une série de nœuds dans un arbre. La racine de l'arbre est une hypothèse vide où le score pour la fonction objectif est essentiellement nul. On lie cette racine à plusieurs nœuds enfants en trouvant les traductions possibles pour le premier segment et en ne conservant que les n meilleures hypothèses. À chacun de ces nœuds on lie les traductions possibles pour le deuxième segment en tenant

compte de ce qui a déjà été traduit, et on conserve les n meilleures hypothèses parmi toutes celles que nous venons de créer. L'algorithme continue de cette façon jusqu'à la traduction complète de la phrase source, à quel point on retourne la meilleure hypothèse qui a été générée.

Cette méthode a été conçue pour accélérer le processus de création des hypothèses de traduction en supprimant les hypothèses moins prometteuses plus tôt dans la chaîne. Par contre, la suppression de ces hypothèses est un problème car nous pourrions supprimer une hypothèse moins prometteuse au départ alors qu'elle pourrait s'avérer être une meilleure hypothèse à la fin. Étant donné la nature même de l'algorithme, le choix de supprimer ou non une hypothèse est basé uniquement sur la portion traduite de la phrase. Ceci s'avère problématique dans le cas où la portion non traduite contient des informations nécessaires à l'évaluation complète de la traduction. La connaissance de ces informations pourrait impliquer un choix de nœuds différent de celui basé uniquement sur la portion traduite de la phrase.

2.5 Vérification de la performance des systèmes de traductions

Le dernier problème concernant la traduction automatique statistique est de choisir quel système de traduction est le meilleur. Ceci peut être fait en vérifiant chacune des traductions produites manuellement, mais le coup d'une telle opération est très élevé, tant en temps qu'en argent. C'est pourquoi il existe plusieurs méthodes pour calculer la performance d'un système de traduction. La méthode la plus simple est de trouver la distance entre la traduction produite par un système et celle produite par un traducteur professionnel pour une certaine phrase. Deux métriques utilisées sont le taux d'erreurs de mots ou *Word Error Rate* (WER) et le taux d'erreur de traduction ou *Translation Error Rate* (TER). Ces deux métriques sont basées sur l'algorithme de distance d'édition de Levenshtein qui sert à calculer le nombre d'opérations (substitution, suppression ou insertion) pour passer d'une phrase à une autre. Les taux d'erreurs sont calculés par la distance Levenshtein entre la traduction et la référence divisée par le nombre de mots ou segments dans la référence.

Une autre métrique souvent utilisée en traduction est le *Bilingual Evaluation Understudy* (BLEU). Cette méthode, maintenant fortement répan-

due, a été introduite dans Papineni, Roukos, Ward, and Zhu [13]. Elle se base sur l'idée de précision (rapport entre le nombre de bons résultats obtenus et le nombre total de résultats obtenus). On peut utiliser cette métrique avec plusieurs traductions ou une seule. L'utilisation de plusieurs traductions est plus coûteuse, car il faut produire ces traductions, mais le résultat est généralement plus précis. Le score BLEU a l'avantage de souvent ressembler plus à ce qu'un traducteur donnerait comme score à un système de traduction et c'est une des raisons majeures pour laquelle celui-ci est devenu la norme dans l'évaluation de systèmes de traduction.

3 Métaheuristiques

Les métaheuristiques sont une branche de l'optimisation se concentrant à trouver les meilleures heuristiques pour un problème ainsi que l'algorithme le plus adapté au problème. Une heuristique peut être vue comme une fonction à optimiser, mais qui pourrait ne retourner aucune solution ou une solution moins bonne que la solution optimale. Elles sont plus souvent utilisées pour des problèmes NP-difficiles, où un algorithme exact n'existe pas en temps raisonnable pour trouver une solution. Elles sont aussi très bien adaptées pour des problèmes où il n'y a pas vraiment de meilleure solution, comme la traduction automatique.

Une notion importante en métaheuristique est l'espace de recherche. Cet espace définit l'ensemble des solutions possibles, ainsi que leur score. Avec un espace de recherche à deux dimensions, nous pouvons afficher le score pour chaque solution comme étant un troisième axe. Le pic d'une montagne dans cette représentation indique un maximum, global si c'est le plus haut de l'espace de recherche ou local s'il en existe un autre plus haut, et les vallées indiquent des minimums, globaux ou locaux encore une fois. Selon le problème, les métaheuristiques tentent de trouver le maximum ou le minimum global pour un espace de recherche, ce qui est parfois très difficile parce que les algorithmes restent dans des maximums et minimums locaux très souvent.

3.1 Algorithmes

Les métaheuristiques sont généralement divisées en trois branches.

construction Des algorithmes partant d'une solution vide et créant une solution au fur et à mesure pour tenter de créer directement la meilleure solution possible (on pourrait penser à la recherche par faisceau comme étant une métaheuristique de construction de solution) ;

recherche locale Des algorithmes prenant une solution déjà construite et la modifiant itérativement pour trouver une meilleure solution ;

évolution Des algorithmes prenant plusieurs solutions déjà construites et les combinant pour générer des meilleures solutions.

Pour les algorithmes de construction, on pense souvent aux colonies de fourmis. Les algorithmes de colonies de fourmis construisent la solution en

utilisant l’analogie des fourmis qui empruntent de plus en plus souvent le chemin que les autres fourmis ont pris en suivant les phéromones laissées par ces dernières. On commence donc avec une solution vide dans laquelle on envoie des fourmis et après plusieurs itérations, les fourmis ont trouvé un chemin à prendre jusqu’à la solution qu’on veut.

Pour les algorithmes de recherche locale, on pense souvent au recuit simulé, à la recherche taboue, à la recherche locale itérée et plusieurs autres variantes de ces algorithmes. Ces algorithmes modifient itérativement une solution en utilisant des opérateurs de voisinage. Les opérateurs de voisinage prennent la solution courante, la modifient légèrement et retournent le résultat de cette modification. L’algorithme le plus simple de cette branche, nommé “Hill Climbing” en anglais, consiste à prendre une solution dont le score est meilleur que la solution courante, jusqu’à ce qu’il n’y ait plus de solutions meilleures. Le grand défaut de cet algorithme est le fait qu’il peut facilement tomber dans un maximum local et donc ne pas trouver la solution optimale. Le recuit simulé est un algorithme qui réplique les cycles de refroidissement et de réchauffement des métaux. Il utilise un paramètre nommé température qui, lorsque plus élevé, permet d’accepter des solutions moins bonnes, ce qui permet de faire des sauts dans l’espace de recherche, et lorsque plus faible (approchant asymptotiquement 0), permet de se restreindre seulement aux solutions améliorant la fonction objectif, pour trouver la meilleure solution possible. Les algorithmes tabous, de leur côté, tentent d’agrandir l’espace de recherche en utilisant une liste d’opérations ou de solutions qui ne peuvent pas être répétées avant un certain nombre d’itérations. Ceci force l’algorithme à aller chercher des solutions différentes, même si la valeur de la fonction objectif est moins bonne, pour tenter de trouver un autre optimum local. La recherche locale itérée, ainsi que la recherche locale à voisinage variable, sont d’autres approches pour agrandir l’espace de recherche. L’idée de la recherche locale itérée est de faire une recherche locale à partir d’une solution, de perturber la nouvelle solution pour se déplacer dans l’espace de recherche et de refaire une recherche locale à partir de cette nouvelle solution. Ceci est répété plusieurs fois en conservant la meilleure solution trouvée jusqu’à date. La recherche locale à voisinage variable, de son côté, comprend plusieurs opérateurs de voisinage lui permettant de se déplacer plus ou moins dans l’espace de recherche. Elle commence d’abord avec des opérateurs lui permettant de grands déplacements pour ensuite réduire les déplacements de plus en plus. Ceci permet une plus grande vision de l’espace de recherche.

Pour les algorithmes évolutifs, on mentionne souvent les algorithmes génétiques et les algorithmes mémétiques. Les algorithmes génétiques utilisent des opérateurs de croisement, en prenant plusieurs solutions existantes et en les combinant pour générer des nouvelles solutions, et des opérateurs de mutation, prenant une solution et la modifiant un peu. Les algorithmes mémétiques utilisent la même stratégie que les algorithmes génétiques, mais ajoutent une étape de recherche locale à chaque nouvelle solution créée pour tenter d’avoir les meilleures solutions possibles.

3.2 Applications en traduction automatique

Les algorithmes de construction sont les plus souvent utilisés en traduction automatique, car on crée une solution à partir de rien en tentant d’optimiser une fonction objectif. La recherche à faisceau utilisée dans les applications de traduction automatique statistique utilise une forme de construction gloutonne, mais en conservant plusieurs hypothèses en parallèle pour agrandir l’espace de recherche le plus possible. C’est le cas pour les systèmes développés par Philipp Koehn : Pharaoh [4] et Moses [6].

Certaines recherches effectuées plus récemment (depuis Germann [1] et Langlais et al. [7]), utilisent des algorithmes de recherche locale pour tenter d’améliorer la solution une fois qu’elle est créée. Les algorithmes les plus souvent utilisés pour ceci sont des algorithmes gloutons. Ces algorithmes vont toujours prendre la meilleure solution parmi les solutions générées par les opérateurs de voisinage. Ils ont l’avantage d’être très rapides, mais ont le désavantage de rarement trouver la meilleure solution. On peut aussi définir une version aléatoire de l’algorithme glouton qui, à chaque itération, choisit aléatoirement une solution parmi celles qui améliorent la fonction objectif par rapport à la solution courante au lieu de prendre la solution améliorant le plus cette fonction. De cette manière, on visite un peu plus l’espace de recherche sans trop augmenter le temps d’exécution l’algorithme. Les autres algorithmes de recherche locale, tel que la recherche taboue, le recuit simulé et la recherche à voisinage variable, n’ont pas été étudiées autant en profondeur dû à leurs exigences en terme de temps d’exécution et à la complexité à bien définir leurs paramètres.

Les algorithmes évolutifs, quoique parmi les plus adaptés pour certains problèmes, sont rarement utilisés en traduction automatique. Un premier problème est leur temps d’exécution beaucoup plus grand que les autres

algorithmes. Un second problème est la complexité d'implémentation de tels algorithmes. La définition des opérateurs de croisement et de mutation est difficile à effectuer pour un problème comme la traduction automatique.

4 Réalisations

4.1 Chaîne de traitement

Pour bien comprendre ce qui se passe lors de la traduction automatique, il est important de connaître les différentes étapes présentes lors de la traduction. Moses est muni d'un outil pour faciliter cette tâche, le *Experiment Management System (EMS)*, et c'est sur ce système que notre chaîne de traitement a été basée. La figure 3 montre les différentes étapes utilisées dans la génération d'une traduction par Moses ainsi que par notre décodeur. Cette chaîne de traitement est divisée en 4 étapes majeures :

L'entraînement lit des fichiers de corpus (ici Europarl-v7²) pour créer les modèles nécessaires ;

La préparation des fichiers prépare les fichiers contenant les phrases à traduire qui seront passées à Moses pour s'assurer que la traduction obtenue sera la meilleure traduction possible ;

Le décodage par Moses traduit les phrases une première fois ;

La recherche locale tente d'améliorer la traduction que l'étape précédente a produite.

La préparation des fichiers et le décodage par Moses sont des étapes déjà implémentées entièrement par les outils entourant Moses. L'entraînement est majoritairement fait avec les outils de Moses, mais nous avons ajouté notre propre programme pour générer la table de collocations à ceux-ci. Nos modèles ont été ajoutés en utilisant le modèle de Moses et peuvent donc être utilisés avec Moses directement. La recherche locale utilise la gestion des modèles et de la fonction objectif de Moses pour calculer la valeur de chaque hypothèse, mais l'engin de recherche a été réécrit pour implémenter une recherche locale à partir d'hypothèses complètes au lieu d'une recherche constructive par faisceau à partir d'hypothèses nulles.

Entraînement

L'entraînement se fait à partir d'un corpus bitexte (anglais-français dans notre cas). Cette étape est découpée en plusieurs étapes mineures. On commence par s'assurer que le corpus est bien décomposé en unités lexicales.

2. Disponible à l'adresse <http://statmt.org/europarl>.

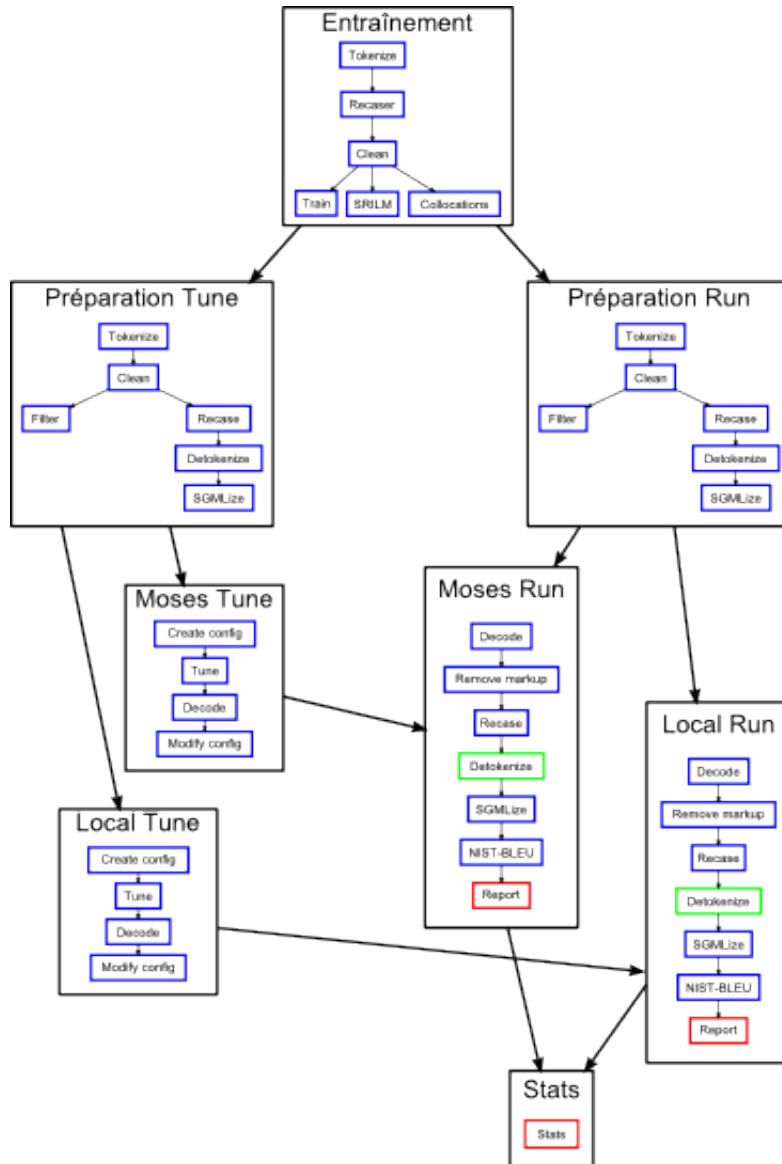


FIGURE 3: Chaîne de traitement du décodeur par recherche locale a posteriori

C'est aussi à cette étape que l'on remplace certains caractères pour s'assurer qu'il n'y aura pas de problème avec Moses (par exemple remplacer les apostrophes par `'` et les guillemets par `"`). Ensuite, il crée un modèle similaire à un modèle de langue pour pouvoir remettre la casse à un fichier. Une fois le modèle créé, il fait un ménage dans les phrases et met tous le texte en minuscules. Puisque tout le traitement se fait en minuscules, c'est après cette étape que l'entraînement crée les autres modèles nécessaires à la traduction. Il crée donc un modèle de langue n -gramme (en utilisant SRILM³ dans notre cas), un modèle de collocations (qui sera expliqué plus tard) et les modèles de traductions et de réordonnement (en utilisant le script `train-moses.pl`⁴).

Préparation

Une fois tous les modèles créés, la chaîne de traitement prépare les fichiers utilisés par Moses de façon similaire au corpus durant l'entraînement pour assurer la plus grande similarité possible. Tout le traitement est ici fait sur deux fichiers en parallèle, soit un fichier à traduire et un fichier contenant une traduction humaine à utiliser comme référence pour l'analyse finale. Elle commence donc par décomposer les fichiers en unités lexicales et par mettre le texte en minuscules. C'est ce fichier qui sera utilisé pour la traduction. Par la suite, on crée un filtre du modèle de traduction pour que la lecture du modèle soit plus rapide à l'exécution du décodeur en créant un modèle ne contenant que les segments jugés pertinents au fichier à traiter. À partir du fichier créé pour la traduction, on remet la casse au fichier avec le modèle entraîné précédemment, on défait les unités lexicales et on transforme le tout en un fichier SGML qui sera utilisé pour l'analyse. À la fin, il reste donc 6 fichiers importants :

- Le fichier à traduire lisible par Moses ;
- Le fichier à traduire lisible par l'analyse (en SGML) ;
- Le fichier contenant la référence de la traduction lisible par Moses ;
- Le fichier contenant la référence de la traduction lisible par l'analyse ;
- Un modèle de traduction simplifié et

3. Plus de détail sur ce modèle de langues sont disponible à l'adresse <http://www.speech.sri.com/projects/srilm/>.

4. Expliqué en détail à l'adresse <http://www.statmt.org/moses/?n=FactoredTraining.HomePage>.

- Un modèle de réordonnement lexical simplifié pour permettre une lecture plus rapide.

Cette étape est fait pour chaque paire de fichiers à traduire, ainsi que la paire de fichiers servant pour le réglage des paramètres de Moses.

Décodage initial par Moses

Le décodage complet est divisé en deux grandes sous-sections. La première est le réglage des poids pour la fonction objectif de Moses. Cette sous-section commence par créer une configuration ayant des poids égaux. Elle utilise ensuite le *Minimum Error-Rate Training* pour tenter d’optimiser les poids de la fonction à optimiser. Ensuite, on génère une traduction du même fichier pour pouvoir l’utiliser dans la recherche locale et on modifie la configuration avec les nouveaux poids trouvés pour pouvoir l’utiliser avec d’autres fichiers à traduire. La deuxième sous-section du décodage est un décodage réel. On utilise la configuration des poids trouvée lors de l’ajustement pour traduire un fichier test. Une fois la traduction terminée, on remet la casse et défait les unités lexicales pour produire le fichier de traduction finale. À partir de cette traduction finale et du fichier contenant la référence de la traduction, on analyse le résultat en utilisant le score BLEU pour pouvoir comparer nos résultats de recherche locale avec ceux du décodage initial. Cette étape créé donc 3 fichiers importants :

- La traduction initiale de Moses, qui sera passée en paramètre à la recherche locale ;
- La traduction finale de Moses, pour comparer les résultats et
- Un rapport contenant les scores BLEU pour cette traduction.

Second décodage avec la recherche locale

Le second décodage contient essentiellement les mêmes opérations que le décodage initial par Moses à la différence que le décodeur est maintenant celui créé pour cette recherche, et que ce décodeur utilise comme entrée un fichier à traduire ainsi que la traduction produite par Moses pour ce fichier à traduire. La recherche locale crée donc 2 fichiers importants :

- La traduction finale, afin de pouvoir comparer les résultats avec ceux de Moses et
- Un rapport contenant les scores BLEU pour cette traduction, qui sera comparé avec celui produit par Moses.

4.2 Algorithmes

Pour la recherche locale, deux algorithmes ont été implémentés. Le premier est l'algorithme glouton décrit dans Langlais et al. [7]. Cet algorithme prend la meilleure transformation à chaque itération jusqu'à ce qu'il n'y ait aucune transformation qui améliore la fonction objectif.

Algorithm 1 Pseudo-code de l'algorithme glouton

```
courant ← traduction produite par Moses
changement ← true
while changement do
  transformations ← courant.GetTransformations()
  meilleur ← transformation avec le meilleur score
  if score(meilleur) > score(courant) then
    changement ← true
    courant ← meilleur
  else
    changement ← false
  end if
end while
return courant
```

Le deuxième algorithme est une version aléatoire de cet algorithme. Au lieu de prendre la meilleure transformation à chaque itération, il prend aléatoirement une transformation parmi la liste des transformations améliorant la fonction à optimiser. Ceci permet une recherche un peu plus approfondie de l'espace de recherche sans trop ralentir le processus de traduction.

Algorithm 2 Pseudo-code de l'algorithme glouton randomisé

```
courant ← traduction produite par Moses
changement ← true
while changement do
  transformations ← courant.GetTransformations()
  possible ← transformations avec un score supérieur à current
  if size(possible) == 0 then
    changement ← false
  else
    changement ← true
    courant ← transformation aléatoire dans possible
  end if
end while
return courant
```

D'autres algorithmes auraient pu être implémentés, tel le recuit simulé ou la recherche avec tabous, mais ces algorithmes requérant plusieurs méta-paramètres à régler (formule de refroidissement pour le recuit simulé et longueur de la liste de tabous pour la recherche taboue), ils n'ont pas été implémentés. La complexité additionnelle pour de tels algorithmes, ainsi que le temps d'exécution beaucoup plus grand, ont fait en sorte qu'ils n'ont pas été choisis pour l'implémentation dans ce travail.

4.3 Transformations

Quatre transformations ont été implémentées. Ces quatre transformations sont les opérations élémentaires pour trouver de nouvelles traductions sans changer la phrase source à traduire. Chacune des transformations applique un changement sur une portion réduite de la phrase cible. La première est le déplacement d'un segment de la traduction vers un autre endroit dans la phrase et est surnommé *move*. Elle agit aussi comme *swap* qui inverse deux segments dans la traduction. La deuxième transformation est le remplacement d'un segment par une autre traduction tirée de la table de traductions pour le même segment de la phrase source et est surnommé *replace*. La troisième, *merge*, prend deux segments côte-à-côte et les joint ensemble en trouvant une nouvelle traduction pour les deux segments maintenant combinés. La dernière transformation, *split*, est l'inverse de *merge*.

Elle prend un segment et le divise en deux segments consécutifs, trouvant une nouvelle traduction pour chacun des nouveaux segments créés.

| | |
|----------------------|---|
| Avant <i>move</i> | {Le chat gris} {boit} {du lait}. |
| Après <i>move</i> | {Le chat gris} {du lait} {boit}. |
| Avant <i>replace</i> | {Le chat gris} {boit} {du lait}. |
| Après <i>replace</i> | {Le chat grisâtre} {boit} {du lait}. |
| Avant <i>merge</i> | {Le chat gris} {boit} {de} {le lait}. |
| Après <i>merge</i> | {Le chat gris} {boit} {du lait}. |
| Avant <i>split</i> | {Le chat gris} {boit} {du lait}. |
| Après <i>split</i> | {Le chat gris} {est en train de} {boire} {du lait}. |

FIGURE 4: Exemples des transformations possibles

Considérant la phrase source «The gray cat is drinking milk.». Les accolades représentent les segments générés durant la traduction.

Dénotons N le nombre de segments dans la phrase traduite et T le nombre de traductions possibles pour un segment dans la table de traduction. *move* crée donc $O(N^2)$ nouvelles traductions et *replace* crée $O(N * T)$ nouvelles traductions. *merge* sera aussi de l'ordre de $O(N * T)$ traductions et *split* sera de l'ordre de $O(N * T^2)$.

4.4 Modèles

L'ajout de différents modèles tel le modèle de langue n -gramme inverse et le modèle de collocations modifient la fonction objectif. Nous ajoutons un ou plusieurs nouveaux éléments dans la sommation de l'équation 5 définie dans la section 2.3. Ceci a pour conséquence que nous devons recalculer les poids pour cette nouvelle fonction objectif pour chaque configuration des différents modèles. Ceci est aussi vrai lorsque nous modifions les paramètres pour un modèle en particulier.

Modèle de langue inverse

Le modèle de langue n -gramme inverse est basé sur le modèle de langue n -gramme utilisé par Moses. La probabilité est calculée en utilisant une chaîne de Markov de droite à gauche au lieu de l'habituel gauche à droite

des modèles de langue. On cherche donc la probabilité d'un mot étant donné les mots suivants. Ce modèle est simple à implémenter et ajoute un peu d'information sur la traduction. Malheureusement, étant donné le mode de traduction des systèmes de traduction comme Moses qui traduisent de gauche à droite, ce genre de modèle de langue n'a pas son utilité lors de la traduction parce qu'il n'y a aucune information sur les mots suivants le mot courant, sauf une fois la traduction terminée. Puisque la recherche locale travaille sur une traduction complétée, le modèle de langue n -gramme inverse ajoute une nouvelle information qui n'est pas disponible lors de la traduction par Moses. Les espérances sont que cette nouvelle information permette un meilleur choix de traduction pour un segment ou un meilleur ordonnancement des segments pour une traduction que ce qui peut être produit par Moses.

Modèle de collocations

Le modèle de collocations qui a été ajouté ici est basé sur un principe similaire à celui d'un modèle de langue n -gramme. Un modèle de langue n -gramme trouvera la probabilité d'un mot étant donné les $n - 1$ mots précédents. Notre modèle de collocations n -gramme trouve la probabilité d'un mot étant donné un mot à une distance de n mots auparavant. L'intuition est que ce modèle ajoutera une information sur les mots qui sont souvent présents dans une même phrase. L'idée de ce modèle est qu'il donne une information similaire à celle d'un modèle de langue n -gramme, mais dans une version beaucoup plus simplifiée, ce qui permet de pouvoir utiliser une distance plus grande que le n -gramme des modèles de langues. La figure 5 montre cette différence. Le mot en italique est le mot pour lequel nous voulons calculer la valeur donnée par le modèle, les parties soulignées sont les mots qui sont pris en compte. Pour un modèle de langue n -gramme on ne regarde que les quelques mots précédant le mot à calculer, ce qui peut causer problème lorsque nous avons une phrase à traduire comme dans l'exemple. Il devient très difficile de correctement accorder «gris» avec «chat», et aurait plus tendance à accorder «gris» avec «voisine», ce qui n'est pas souhaitable. Par contre pour un modèle de collocations, nous regardons tous les mots de la phrase, mais de façon individuelle. Il devient donc plus facile d'accorder «gris» avec «chat» dans ce cas, parce que les mots «chat» et «gris» ont probablement été vus plus souvent dans la même phrase que «voisine» et «grise».

Modèle de langue n -gramme :

Le chat, qui appartient à la voisine, est *gris*.

Modèle de collocations :

Le chat, qui appartient à la voisine, est *gris*.

FIGURE 5: Différence entre un modèle de langue n -gramme ordinaire et notre modèle de collocations

L'entraînement de ce modèle est fait en lisant la partie cible du corpus d'entraînement, le même que pour le modèle de langue n -gramme. On crée ensuite un tableau de toutes les cooccurrences entre deux termes pour chacune des fenêtres possibles, ainsi que le nombre de fois où chacune de ces cooccurrences a été vue. Les fenêtres possibles indiquent le nombre de mots entre le premier mot et le second. Une fenêtre de 0 veut donc dire que les deux mots ont été vus l'un directement à côté de l'autre, alors qu'une fenêtre de 3 indique qu'il y avait 3 mots, ici ignorés, entre les mots sauvegardés. Le modèle est ensuite enregistré dans un fichier avec le premier mot, le second mot et le nombre d'occurrences pour chacune des distances (voir Figure 6 pour un exemple de fichier produit). Chaque ligne débutant par “//” indique le nombre de mots entre le premier et le deuxième pour les lignes suivantes et les autres lignes définissent le premier mot, le deuxième mot et le nombre d'occurrences respectivement. On peut donc noter qu'une distance de zéro revient exactement à un modèle de langue bigramme. Une distance de trois est équivalente à un modèle de langue 5-gramme, comme celui qui est utilisé dans nos tests. Une distance de huit serait équivalente à un modèle de langue 10-gramme, ce qui est rarement utilisé parce que les modèles deviennent beaucoup trop gros, ce qui n'est pas le cas avec le modèle de collocations car il n'y a que deux mots à se souvenir peu importe la distance.

```

// 0
brit not 1
specialise and 2
or italian 20
...
traditional air 2
// 1
specialise and 1
retain genuine 1
defined framework 4
...

```

FIGURE 6: Exemple de fichier du modèle de collocations

Seulement le nombre d'occurrences est sauvegardé ici pour permettre de ne lire qu'une section du fichier et calculer les probabilités sur cette section uniquement. Toutes les distances possibles sont sauvegardées lors de l'entraînement et c'est à la lecture du modèle qu'une coupure est faite sur la distance. Sauvegarder seulement le nombre d'occurrences permet aussi d'utiliser des fonctions de score différentes pour le modèle. Trois fonctions de score ont été implémentées à partir de ce modèle.

La première est une fonction de compte binaire. On passe à travers la phrase traduite mot à mot et on compte le nombre de collocations vues. C'est donc une somme d'une fonction définie comme 1 si la collocation est présente dans le modèle et 0 sinon. La somme est ensuite divisée par le nombre de mots dans la traduction pour normaliser le score. Cette fonction sera référée par la suite comme *collc*.

La deuxième fonction est une variation de la première. Dans cette version, la fonction utilisée dans la somme retourne le nombre d'occurrences du modèle si la collocation est présente et 0 sinon. La même normalisation est appliquée que pour la première fonction. Cette fonction sera référée par la suite comme *colls*.

La troisième fonction utilise des probabilités. On fait une somme des probabilités d'un mot étant donné un autre mot à une certaine distance de celui-ci. La formule 7 montre le calcul des probabilités. On calcule donc le nombre de fois où la collocation a été vue divisé par le nombre de collocations à la même distance où le premier mot est le même. Ces probabilités

sont calculées lors de la lecture du modèle au début du décodage. Encore ici, la normalisation est utilisée pour ne pas favoriser les traductions plus longues. Cette fonction sera référée plus tard par *collp*.

$$p(w_j|w_i, d) = \frac{nb(w_i, w_j, d)}{\sum_{k \in corpus} nb(w_i, w_k, d)} \quad (7)$$

Ces trois fonctions pour l'évaluation du modèle de collocations peuvent aussi être modifiées par deux paramètres booléens :

distanced On tient compte ou non de la distance lors du calcul. Une valeur positive indique que la distance entre les mots est utilisée dans le calcul de la fonction de score. Une valeur négative indique qu'elle n'est pas prise en compte, soit $F(w_i, w_j) = \sum_{d=0}^{\infty} nb(w_i, w_j, d)$.

directed La direction de la collocation est prise en compte ou non. Une valeur positive pour ce paramètre est équivalente à ce qui a été décrit précédemment pour les fonctions de scores. Si une valeur négative est utilisée, les calculs se font avec une fonction qui ne tient pas compte de l'ordre des mots, soit $F(w_i, w_j, d) = nb(w_i, w_j, d) + nb(w_j, w_i, d)$.

Le modèle de collocation est donc défini dans Moses par une fonction prenant comme paramètres la distance maximale à utiliser et les paramètres *distanced* et *directed*. Intuitivement, on tend à croire qu'une distance trop courte, soit une distance plus petite ou égale à celle du modèle de langue *n*-gramme, ne donnera pas plus d'information que le modèle de langue *n*-gramme. De même, on peut penser qu'une distance trop grande amènera du bruit dans la fonction de score, ce qui pourrait être néfaste pour les résultats finaux. Le fait de ne pas prendre en compte la distance pourrait donner plus d'information, parce que notre corpus n'est probablement pas assez massif pour contenir toutes les collocations possibles à toutes les distances possibles et le fait de ne pas prendre en compte la direction pourrait aussi donner un peu plus d'information, mais pourrait aussi ajouter du bruit dans certains cas ou un mot devrait se trouver avant et non après un autre.

5 Résultats

5.1 Données

L’entraînement des modèles a été effectué sur le corpus europarl⁵ qui contient environ 2 millions de phrases du Parlement Européen qui ont été traduites par des humains dans la plupart des langues parlées en Europe (1 961 944 phrases dans notre cas pour le corpus Français-Anglais). Toutes les expériences ont été effectuées en réglant les poids pour la fonction de score avec le corpus de test «newstest2008» et les résultats sont présentés pour les corpus «newstest2009», «newstest2010» et «newstest2011» disponibles avec le corpus europarl. Ces corpus proviennent d’articles de nouvelles au courant de l’année mentionnée. La table 1 présente plusieurs statistiques sur les différents corpus de tests. La colonne “Connu” présente le nombre de phrases dont tous les mots sont connus (c’est-à-dire que les mots ont été vus dans le corpus d’entraînement). La colonne “Inconnu” présente le nombre de phrases dont au moins un mot est inconnu.

| Utilisation | Corpus | Total | Connu | Inconnu |
|--------------|--------------|-----------|-------------------|---------------|
| Entraînement | Europarl | 1 961 944 | 1 961 944 (100 %) | 0 (0 %) |
| Dev | newstest2008 | 2051 | 1337 (65.2 %) | 714 (34.8 %) |
| Test | newstest2009 | 2525 | 1606 (63.6 %) | 919 (36.4 %) |
| Test | newstest2010 | 2489 | 1531 (61.5 %) | 958 (38.5 %) |
| Test | newstest2011 | 3003 | 1832 (61.0 %) | 1171 (39.0 %) |

TABLE 1: Statistiques pour les corpus utilisés pour les tests

5.2 Présentation des résultats

Les sections suivantes présenteront plusieurs tables pour montrer les effets des algorithmes et de leurs paramètres. Les tables seront divisées selon les groupes de colonnes suivants :

- Les premières colonnes présenteront les différents algorithmes ou paramètres examinés ;

5. Disponible à l’adresse <http://statmt.org/europarl>

- Les colonnes “op” et “mod” présenteront respectivement le nombre de lignes ayant subi au moins une opération et le nombre de lignes ayant été modifiées textuellement ;
- Les colonnes #mov, #rep, #mer et #spl présenteront respectivement le nombre d’opérations *move*, *replace*, *merge* et *split* ;
- Les colonnes “B”, “BK” et “BU” présenteront le score BLEU pour tout le fichier de test (“B”), pour les lignes dont tous les mots sont connus (“BK”) et pour les lignes dont au moins un mot est inconnu (“BU”) ;
- La colonne “Temps” présentera le temps de calcul utilisé en secondes sur notre machine. Il est à noter que ce temps doit être ajouté à celui du décodeur Moses car notre décodeur, utilisant la sortie de Moses, doit être lancé après celui-ci.

Le nombre de lignes ayant subi au moins une opération sera toujours supérieur ou égal au nombre de lignes ayant été modifiées, car on ne peut pas avoir une ligne modifiée sans qu’il n’y ait eu d’opération sur celle-ci. La différence entre ces deux nombres représente donc le nombre de lignes pour lesquelles les modifications apportées n’ont eu aucun effet sur la phrase finale. Les opérations *move* et *replace* modifient obligatoirement le résultat final à moins qu’une opération inverse ne soit appliquée. Les opérations *merge* et *split* ne modifient pas nécessairement le résultat final car on peut par exemple prendre un segment de 5 unités et le segmenter avec l’opération *split* en un segment de 3 unités et un segment de 2 unités identique à l’origine. La même logique peut être appliquée à l’inverse pour l’opération *merge*. Étant donné que plusieurs éléments de la fonction de coût calculent leur score selon les segments de la phrase (par exemple le modèle de traduction et le modèle de langue), il arrive que des opérations *merge* et *split* ne modifiant pas la phrase soient appliquées car la fonction de coût juge la segmentation meilleure.

L’implémentation de BLEU que nous utilisons pour faire l’évaluation automatique des traductions est celle disponible avec Moses et nous utilisons une seule phrase de référence pour faire la comparaison.

Comme décrit à la section 4.2, deux algorithmes ont été implémentés : un algorithme glouton et une version randomisée de cet algorithme glouton. Nous référerons à l’algorithme glouton non-randomisé dans les tableaux par «Glouton» et à la version randomisée par «GloutonR». Nous comparons nos résultats avec ceux produits par Moses.

Pour nos tests, tous les poids pour la fonction de coût sont trouvés avec l’algorithme de “Minimum Error Rate Training” (“MERT”) [12] en

utilisant le programme fourni avec le code source de Moses. Ce dernier utilise Moses itérativement avec le même corpus en changeant les poids pour chaque modèle à chaque itération. Il analyse ensuite les résultats obtenus avec BLEU et tente de trouver les poids qui maximisent le score obtenu avec BLEU. Puisque l'analyse est aussi effectuée en utilisant BLEU, nous nous attendons à ce que les poids soient correctement choisis pour notre tâche. L'ajout de modèles dans la fonction objectif nécessite un réglage des poids pour cette nouvelle fonction objectif. Ceci a été effectué pour nos tests en utilisant toujours le même algorithme, MERT.

5.3 Algorithmes et réglage

Le premier point important à considérer est le réglage des poids de la fonction de score. Pour vérifier l'efficacité de nos algorithmes, nous utilisons donc des poids réglés pour l'utilisation avec Moses et des poids réglés avec l'utilisation pour nos algorithmes. Les tables 2, 3 et 4 donnent les résultats BLEU pour chaque expérience, ainsi que le temps de calcul en secondes. Cette première section démontre seulement l'effet du réglage des poids sans ajout de modèles dans la fonction objectif.

La ligne Moses sans réglage indique notre point de départ. Les lignes Glouton et GloutonR avec le réglage Moses sont le résultat de chaque algorithme avec les mêmes poids que pour Moses. Les lignes Glouton et GloutonR avec le réglage *retune* sont le résultat de chaque algorithme, mais cette fois-ci en refaisant le réglage des poids avec l'algorithme utilisé par la recherche locale. Les dernières lignes avec Moses et les réglages Glouton et GloutonR sont les résultats de Moses seul avec les paramètres réglés pour Glouton ou GloutonR. Ces dernières expériences sont pour vérifier l'apport du réglage des poids de la fonction de score dans le résultat final. Les scores BLEU donnent une indication sur la qualité de la traduction (plus élevé est mieux) et le temps de calcul donné est à titre d'indication seulement. Les temps de calcul donnés sont ceux du décodage uniquement. Pour Glouton et GloutonR, seul le temps de ce décodage est indiqué, il faut donc y ajouter le temps de calcul pour Moses (première ligne).

| Algorithme | Réglage | B | BK | BU | Temps |
|------------|----------|-------|-------|-------|-------|
| Moses | | 23.83 | 23.60 | 24.13 | 977 |
| Glouton | Moses | 23.83 | 23.60 | 24.13 | 478 |
| | Retune | 23.75 | 23.58 | 23.97 | 508 |
| GloutonR | Moses | 23.83 | 23.60 | 24.13 | 445 |
| | Retune | 23.80 | 23.58 | 24.08 | 542 |
| Moses | Glouton | 23.60 | 23.39 | 23.86 | 1009 |
| | GloutonR | 23.76 | 23.52 | 24.07 | 984 |

TABLE 2: Comparaison des algorithmes et des réglages (newstest2009)

| Algorithme | Réglage | B | BK | BU | Temps |
|------------|----------|-------|-------|-------|-------|
| Moses | | 24.36 | 25.18 | 23.40 | 973 |
| Glouton | Moses | 24.36 | 25.19 | 23.39 | 437 |
| | Retune | 24.34 | 25.14 | 23.39 | 449 |
| GloutonR | Moses | 24.36 | 25.19 | 23.39 | 441 |
| | Retune | 24.31 | 25.14 | 23.34 | 478 |
| Moses | Glouton | 24.26 | 25.11 | 23.27 | 987 |
| | GloutonR | 24.29 | 25.25 | 23.16 | 966 |

TABLE 3: Comparaison des algorithmes et des réglages (newstest2010)

| Algorithme | Réglage | B | BK | BU | Temps |
|------------|----------|-------|-------|-------|-------|
| Moses | | 25.22 | 25.95 | 24.34 | 1133 |
| Glouton | Moses | 25.22 | 25.95 | 24.33 | 514 |
| | Retune | 25.21 | 26.04 | 24.21 | 518 |
| GloutonR | Moses | 25.22 | 25.95 | 24.33 | 525 |
| | Retune | 25.19 | 25.98 | 24.24 | 524 |
| Moses | Glouton | 25.24 | 25.95 | 24.39 | 1119 |
| | GloutonR | 25.24 | 26.01 | 24.33 | 1127 |

TABLE 4: Comparaison des algorithmes et des réglages (newstest2011)

La première chose que l'on peut observer est le fait que les scores pour Glouton et GloutonR avec la configuration de Moses sont identiques à ceux de Moses. En poussant un peu l'analyse, on se rend compte que moins d'un

pourcent des phrases ont subi une transformation. Quant aux algorithmes utilisant des poids reparamétrisés, les scores sont un peu inférieurs à ceux de Moses, mais on voit un beaucoup plus grand nombre de phrases transformées, de l'ordre de 30 à 40 pourcents. Ceci indique donc une très grande importance dans la reconfiguration des poids de la fonction de score pour les algorithmes de recherche locale a posteriori. Un autre point que l'on peut remarquer est la différence de score en utilisant les poids de Glouton et GloutonR avec l'algorithme de Moses. La différence de score étant minime, et même négative dans la plupart des cas, il est difficile de comprendre ce qui se passe sans regarder un peu plus en profondeur les résultats et les statistiques.

En regardant des statistiques sur les opérations qui ont été effectuées pendant la recherche locale, nous pouvons comprendre un peu plus ce qui se passe. Prenons par exemple le corpus newstest2009, un fichier de test de 2495 lignes, et ses statistiques présentées dans la table 5 :

| Algorithme | Réglage | op | mod | #mov | #rep | #mer | #spl |
|------------|----------|-----|-----|------|------|------|------|
| Glouton | Moses | 3 | 3 | 3 | 0 | 0 | 1 |
| | Glouton | 815 | 687 | 37 | 570 | 243 | 206 |
| GloutonR | Moses | 3 | 3 | 3 | 0 | 0 | 1 |
| | GloutonR | 608 | 491 | 81 | 342 | 68 | 285 |

TABLE 5: Statistiques pour le corpus newstest2009 avec les algorithmes de base

On voit rapidement qu'il y a une grande différence entre les poids paramétrés par Moses et les poids paramétrés avec l'algorithme de recherche locale. La première chose que l'on remarque est la très grande différence entre le nombre de lignes modifiées en utilisant les poids de Moses (3 dans les deux cas) comparativement à 687 et 491 pour les poids trouvés avec l'algorithme glouton ou glouton-randomisé respectivement. Le fichier de test ne comprenant que 2495 lignes, on comprend donc rapidement pourquoi la différence de score est si minime dans la plupart des cas : seulement 20% à 30% des lignes sont modifiées. Ceci semble indiquer, d'une part, que Moses adapte très bien ses poids pour son algorithme, et d'autre part, que Moses adapte très mal ses poids pour un autre algorithme. Malgré le fait que les tableaux précédents démontrent une diminution dans le score, on peut pen-

ser que la méthode de paramétrage de poids, dans notre cas MERT, n'est peut-être pas optimale ou bien appliquée à la recherche locale a posteriori.

Un autre point que l'on peut remarquer dans ces statistiques sont le nombre d'opérations. Dans les deux cas où l'algorithme utilisait ses poids, on note une forte proportion de *replace*, ceci pourrait indiquer qu'il y a justement une grande partie pouvant être améliorée à partir des *replace*. En regardant les statistiques pour les autres fichiers de tests, on observe les mêmes tendances, soit une forte proportion de *replace* et beaucoup plus de lignes modifiées en utilisant les poids reparamétrés avec l'algorithme versus ceux de Moses.

Avec ces statistiques, nous pouvons plus facilement comprendre pourquoi les scores sont très similaires. Malgré les résultats négatifs, ces résultats nous indiquent clairement que nous ne pouvons pas utiliser les poids de Moses avec la recherche locale. Pour le reste de la recherche les poids seront toujours recalculés, d'une part en vue de ces résultats et d'autre part parce que l'on ajoute des modèles qui ne sont pas présents dans la fonction objectif de Moses.

Pour voir un peu plus en détail pourquoi les résultats en BLEU sont aussi bas comparativement à Moses, nous avons analysé manuellement plusieurs phrases prises aléatoirement parmi celles qui ont été modifiées par la recherche locale. La table 6 présente les résultats de cette analyse. La colonne "Moses" présente le nombre de phrases pour lesquelles la traduction de Moses était plus juste selon nous, la colonne "Locale" présente le nombre de phrases pour lesquelles la traduction de la recherche locale était plus juste et la colonne "Équivalent" présente le nombre de phrases pour lesquelles la traduction était équivalente entre Moses et la recherche locale. Selon ces résultats, notre recherche locale semble produire de meilleures traductions que Moses, contrairement à ce que BLEU nous indique. Ceci peut être dû au fait que BLEU n'est pas bien adapté pour bien évaluer des traductions. Ce résultat est par contre très encourageant pour la recherche locale a posteriori. L'analyse complète des résultats est disponible en annexe A.1.

| Test | Moses | Locale | Équivalent | Total |
|--------------|-------|--------|------------|-------|
| newstest2009 | 14 | 22 | 14 | 50 |
| newstest2010 | 19 | 21 | 10 | 50 |
| newstest2011 | 20 | 18 | 12 | 50 |
| Total | 53 | 61 | 36 | 150 |

TABLE 6: Comparaison humaine entre Moses et la recherche locale sans modèle ajouté sur un échantillon aléatoire

5.4 Modèle de langue inverse

La deuxième série d’expériences est pour vérifier l’apport du modèle de langue inverse dans la fonction de score. Les tables 7, 8 et 9 présentent les résultats de l’utilisation du modèle de langue inverse (MLI) ou non avec chacun des algorithmes. Les deux premières lignes présentent la performance sans l’utilisation du modèle de langue inverse, ces lignes sont les mêmes que les lignes Glouton et GloutonR avec *retune* des tables de la sous-section précédente. Les deux lignes suivantes présentent la performance avec l’utilisation du modèle de langue inverse.

| Algorithme | MLI | B | BK | BU | Temps |
|------------|-----|-------|-------|-------|-------|
| Moses | Non | 23.83 | 23.60 | 24.13 | 977 |
| Glouton | Non | 23.80 | 23.58 | 24.08 | 508 |
| | Oui | 23.86 | 23.68 | 24.09 | 674 |
| GloutonR | Non | 23.80 | 23.58 | 24.08 | 542 |
| | Oui | 23.80 | 23.61 | 24.04 | 694 |

TABLE 7: Utilisation du modèle de langue inverse (newstest2009)

| Algorithme | MLI | B | BK | BU | Temps |
|------------|-----|-------|-------|-------|-------|
| Moses | Non | 24.36 | 25.18 | 23.40 | 973 |
| Glouton | Non | 24.34 | 25.14 | 23.39 | 449 |
| | Oui | 24.25 | 25.08 | 23.27 | 655 |
| GloutonR | Non | 24.31 | 25.14 | 23.34 | 478 |
| | Oui | 24.26 | 25.10 | 23.28 | 679 |

TABLE 8: Utilisation du modèle de langue inverse (newstest2010)

| Algorithme | MLI | B | BK | BU | Temps |
|------------|-----|-------|-------|-------|-------|
| Moses | Non | 24.36 | 25.18 | 23.40 | 973 |
| Glouton | Non | 24.34 | 25.14 | 23.39 | 449 |
| | Oui | 25.17 | 25.94 | 24.25 | 742 |
| GloutonR | Non | 24.31 | 25.14 | 23.34 | 478 |
| | Oui | 25.14 | 25.88 | 24.24 | 770 |

TABLE 9: Utilisation du modèle de langue inverse (newstest2011)

Selon le travail de Langlais et al. [7], l'ajout d'un modèle de langue inverse devrait améliorer les résultats. Dans notre cas, ceci n'est vrai que pour un des trois corpus (newstest2011). Ceci démontre malgré tout qu'il y a là une information qui est pertinente à aller chercher. L'information apportée par le modèle peut prioriser une traduction qui est plus souvent rencontrée dans la langue cible. Prenons par exemple une phrase du corpus newstest2010 (Figure 7). L'algorithme Glouton sans le modèle de langue inverse n'a pas pu améliorer cette traduction, mais avec le modèle de langue inverse, la traduction est maintenant améliorée. Le remplacement de «the workforce» par «labour» reflète l'utilisation générale en anglais : «workforce» réfère aux employés en tant que tels alors que «labour» réfère au travail effectué.

Phrase Source :

«L'utilisateur peut inclure des variables comme le prix de la main d'œuvre, des majorations ou des réductions, puis il peut imprimer le devis pour le remettre au client.»

Traduction de Moses :

«The user can include variables as the price of the workforce, increases or reductions, then it can set the estimate for the customer.»

Traduction par la recherche locale avec le modèle de langue inverse :

«The user can include variables as the price of labour, increases or reductions, then it can set the estimate for the customer.»

FIGURE 7: Exemple d'amélioration en utilisant le modèle de langue inverse

Un autre exemple de l'amélioration du modèle de langue inverse est dans l'ordre des mots. Il y a souvent des inversions de mots à faire entre l'anglais et le français. L'expression «les grandes compagnies mondiales d'énergie» a été traduite par Moses par «the big global companies energy» alors que la recherche locale avec le modèle de langue inverse a amélioré la traduction pour produire «the big global energy companies» qui est beaucoup plus approprié. La même phrase n'a pas été modifiée par l'algorithme glouton sans le modèle de langue.

Malheureusement nous pouvons aussi trouver des exemples pour lesquels le modèle de langue inverse détériore la traduction. Un merveilleux exemple se trouve dans la figure 8. Comme on peut le voir, la traduction par Moses n'est pas celle souhaitée, et celle avec la recherche locale sans le modèle de langue inverse non plus, quoiqu'elle est équivalente à celle de Moses. Par contre, la recherche locale gloutonne avec le modèle de langue inverse a modifié la traduction pour obtenir une phrase qui n'est d'une part pas syntaxiquement correcte en anglais, et d'autre part ne transmet pas bien le message.

Phrase Source :

«Pour soutenir le secteur, il faudrait entre 16 et 18 millions d’euros par an.»

Traduction souhaitée :

«To support this sector, we need between 16 and 18 million euros per year.»

Traduction de Moses :

«To support this sector, it should be between 16 and 18 million euros per year.»

Traduction par la recherche locale sans le modèle de langue inverse :

«To support this sector, there should be between 16 and 18 million euros per year.»

Traduction par la recherche locale avec le modèle de langue inverse :

«To support this sector, we should between 16 and 18 million euros per year.»

FIGURE 8: Exemple de détérioration en utilisant le modèle de langue inverse

Une première conclusion serait donc que l’utilisation d’un modèle de langue n’est pas toujours une bonne idée. Il est difficile, voire impossible, de prédire si le modèle améliorera ou détériorera la traduction. Ceci réduit grandement son potentiel d’utilisation dans un contexte où les traductions ne seront pas revues manuellement. Ce que l’on remarque aussi est le fait que l’ajout d’un seul modèle dans la fonction de coût peut complètement changer les résultats de la recherche locale a posteriori.

| Corpus | MLI | op | mod | #mov | #rep | #mer | #spl | BLEU |
|--------|-----|-----|-----|------|------|------|------|-------|
| 2009 | Non | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | Oui | 830 | 547 | 80 | 384 | 106 | 550 | 23.86 |
| 2010 | Non | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | Oui | 837 | 576 | 77 | 370 | 113 | 538 | 24.25 |
| 2011 | Non | 866 | 730 | 45 | 570 | 266 | 246 | 24.72 |
| | Oui | 963 | 647 | 97 | 437 | 129 | 615 | 25.17 |

TABLE 10: Statistiques pour les différents corpus avec l’utilisation du modèle de langue inverse ou non

En regardant les statistiques produites par la recherche locale, présentées dans la table 10, nous voyons que deux des trois corpus ont vu une augmentation du score BLEU avec le modèle de langue inverse, alors que le troisième a vu une réduction. Une deuxième observation est que l’opération la plus utilisée est maintenant la *split*, comparativement au *replace* sans le modèle de langue inverse. Il y a aussi un plus grand nombre de phrase modifiées par des opérations, mais un plus petit nombre de phrases physiquement modifiées.

Ces statistiques démontrent bien une amélioration globale de la traduction due au modèle de langue inverse, mais démontrent aussi que le modèle de langue inverse peut aussi détériorer la traduction.

Nous avons ici aussi analysé plusieurs phrases modifiées choisies aléatoirement. La table 11 présente les résultats de l’analyse manuelle sur celles-ci. Malgré les résultats ambigus selon BLEU, nous voyons encore une fois que la recherche locale semble améliorer les résultats. Les résultats sont toutefois similaires à BLEU :

- Pour le corpus newstest2009, BLEU indiquait une légère amélioration du score et notre analyse indique une grande amélioration de la traduction ;
- Pour le corpus newstest2010, BLEU indiquait une légère diminution du score et notre analyse indique une amélioration de la traduction ;
- Pour le corpus newstest2011, BLEU indiquait une plus grande amélioration du score et notre analyse indique aussi une plus grande amélioration de la traduction.

Ainsi, BLEU n’est probablement pas la métrique la mieux adaptée pour analyser nos traductions. Il analyse correctement les modifications globalement, mais ses critères d’évaluation sont trop sévères et évalue donc souvent

à la baisse les traductions. On peut aussi constater que la différence entre le nombre de phrases où Moses a produit une meilleure traduction et le nombre de phrases où la recherche locale a produit une meilleure traduction est plus grand que dans les tests sans modèle ajouté. Ceci indique donc que le modèle de langue inverse a une grande influence sur la recherche locale a posteriori. L'analyse complète est disponible dans l'annexe A.1.

| Test | Moses | Locale | Équivalent | Total |
|--------------|-------|--------|------------|-------|
| newstest2009 | 10 | 25 | 15 | 50 |
| newstest2010 | 14 | 27 | 9 | 50 |
| newstest2011 | 9 | 32 | 9 | 50 |
| Total | 33 | 84 | 33 | 150 |

TABLE 11: Comparaison humaine entre Moses et la recherche locale avec le modèle de langue inverse sur un échantillon aléatoire

5.5 Modèle de collocations

La troisième série d'expériences utilise le modèle de collocations pour tenter d'améliorer les traductions. Cette fois-ci, afin d'analyser plus en détail le modèle sans notion d'aléatoire, seul l'algorithme glouton non-randomisé a été utilisé. Les paramètres pour le modèle de collocations sont :

évaluation méthode d'évaluation telle que décrite à la section 4.4 (compte, somme, prob) ;

taille taille de la fenêtre (entier ; le nombre de mots entre les deux mots, soit deux de moins que l'équivalent n-gramme) ;

ordre utilisation de l'ordre des mots ou non (booléen ; vrai indique que les mots doivent se retrouver dans le même ordre, faux utilise aussi l'inversion des deux mots dans les calculs de score) ;

distance utilisation de la distance entre les mots ou non (booléen ; vrai indique que les mots sont comptabilisés seulement s'ils se retrouvent à la même distance ; faux indique que les mots sont comptabilisés s'ils ont été vus à une distance inférieure ou égale à la taille de la fenêtre).

Les résultats seront regroupés par méthode d'évaluation, en analysant d'abord l'effet de la taille de la fenêtre puis l'effet de l'ordre et de la distance sur les résultats. Un retour sera ensuite fait sur l'ensemble des résultats.

La table 12 présente des statistiques sur le modèle de collocations entraîné sur le corpus europarl qui contient 1 961 944 phrases. À titre de comparaison, notre modèle de traduction entraîné sur le même corpus contient 65 972 502 différentes paires de segments et notre modèle de langue 5-gramme contient 14 086 848 segments différents de longueur 5 ou moins. Pour une fenêtre équivalente au modèle de langue 5-gramme, nous avons recensé plus de 20 millions de collocations. On voit qu’une grande partie des collocations, soit 72.023%, n’ont été vues qu’une seule fois et près de 90% ont été vues moins de trois fois.

| | | |
|--|-------------|--------|
| Nombre total de collocations | 156 672 582 | 100% |
| Nombre de collocations vues plus d’une fois | 43 831 979 | 27.98% |
| Nombre de collocations vues plus de trois fois | 16 377 252 | 10.45% |
| Nombre de collocations vues plus de dix fois | 3 679 454 | 2.35% |
| Nombre de collocations vues plus de cent fois | 367 353 | 0.23% |
| Nombre de collocations vues plus de mille fois | 26 795 | 0.017% |
| Nombre de collocations vues plus de dix mille fois | 1 585 | 0.001% |
| Nombre pour une fenêtre de 0 | 2 856 775 | 1.82% |
| Nombre pour une fenêtre de 3 | 6 324 235 | 4.04% |
| Nombre pour une fenêtre de 5 | 6 368 481 | 4.07% |
| Nombre pour une fenêtre de 8 | 5 886 334 | 3.78% |
| Nombre pour une fenêtre de 10 | 5 493 269 | 3.51% |
| Nombre pour une fenêtre de 0 ou moins | 2 856 775 | 1.82% |
| Nombre pour une fenêtre de 3 ou moins | 20 267 752 | 12.94% |
| Nombre pour une fenêtre de 5 ou moins | 33 019 822 | 21.08% |
| Nombre pour une fenêtre de 8 ou moins | 51 207 043 | 32.69% |
| Nombre pour une fenêtre de 10 ou moins | 62 394 732 | 39.83% |
| Plus grande fenêtre vue : 78 | | |
| Collocation la plus vue : “the” “,” | 99 606 | |
| Fenêtre la plus peuplée : 4 | 6 383 589 | 4.08% |
| Première fenêtre ayant moins de 1% : 34 | 1 545 680 | 0.99% |
| Première fenêtre ayant moins de 2% : 23 | 3 012 902 | 1.92% |
| Première fenêtre ayant plus de 90% cumulatif : 35 | 141 021 964 | 90.01% |
| Première fenêtre ayant plus de 50% cumulatif : 14 | 82 341 965 | 52.56% |

TABLE 12: Statistiques pour les collocations vues dans le corpus europarl

Nous avons décidé de faire nos tests avec une fenêtre d’une taille allant jusqu’à 10 pour conserver des temps de calculs raisonnables comparativement à notre algorithme sans le modèle de collocations. Le nombre

de collocations pour une fenêtre spécifique reflète le nombre de collocations utilisées lorsque notre paramètre “distance” est à Oui alors que le nombre de collocations pour une fenêtre ou moins reflète la valeur du paramètre “distance” Non. Le paramètre “ordre” à Oui revient essentiellement à doubler ces chiffres car on permet l’inversion.

Méthode d’évaluation par compte binaire

Regardons d’abord le modèle de collocations avec la méthode de compte binaire. Cette méthode est de loin la plus simple des trois, mais pourrait nous donner un bon aperçu du modèle.

| Corpus | Taille | op | mod | #mov | #rep | #mer | #spl | B |
|--------|---------|------|-----|------|------|------|------|-------|
| 2009 | Moses | | | | | | | 23.83 |
| | Glouton | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | 0 | 881 | 547 | 99 | 374 | 137 | 577 | 23.89 |
| | 3 | 968 | 777 | 77 | 631 | 352 | 253 | 23.78 |
| | 5 | 588 | 478 | 67 | 338 | 168 | 117 | 23.81 |
| | 8 | 545 | 509 | 49 | 353 | 40 | 226 | 23.89 |
| | 10 | 617 | 503 | 28 | 408 | 201 | 109 | 23.75 |
| 2010 | Moses | | | | | | | 24.36 |
| | Glouton | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | 0 | 912 | 598 | 98 | 376 | 125 | 638 | 24.30 |
| | 3 | 921 | 730 | 66 | 553 | 348 | 278 | 24.29 |
| | 5 | 576 | 459 | 60 | 317 | 186 | 119 | 24.35 |
| | 8 | 545 | 523 | 39 | 332 | 44 | 230 | 24.28 |
| | 10 | 621 | 506 | 39 | 376 | 202 | 119 | 24.34 |
| 2011 | Moses | | | | | | | 25.22 |
| | Glouton | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | 0 | 1061 | 722 | 128 | 465 | 141 | 670 | 25.18 |
| | 3 | 1063 | 807 | 77 | 638 | 391 | 318 | 25.24 |
| | 5 | 620 | 494 | 85 | 361 | 176 | 127 | 25.22 |
| | 8 | 616 | 571 | 57 | 372 | 47 | 281 | 25.14 |
| | 10 | 689 | 566 | 50 | 443 | 215 | 110 | 25.22 |

TABLE 13: Effet de la taille de la fenêtre pour la méthode d’évaluation par compte du modèle de collocations (colle)

Selon les résultats et les statistiques présentés dans la table 13, la taille de la fenêtre ne semble pas avoir d'impact significatif sur ceux-ci. Nous nous attendions à retrouver des scores qui suivent une distribution normale, où les tailles trop petites et trop grandes ne donneraient pas d'aussi bons résultats. Ceci ne semble pas être le cas. Le modèle étant simpliste, nous ne nous attendions pas à obtenir de très bons résultats, et c'est effectivement le cas. La plupart des scores sont plus bas que ceux de la recherche locale sans le modèle de collocations, sauf pour le corpus newstest2009 où tous les scores sont supérieurs à ceux de la recherche locale seule. Un point important à noter par contre est le temps d'exécution qui augmente grandement avec la taille de la fenêtre. Rappelons que l'algorithme glouton a requis 600 à 700 secondes pour tenter d'améliorer la traduction de Moses. En ajoutant le modèle de collocations, le temps de calcul augmente rapidement, passant d'une centaine de secondes de plus pour la fenêtre de taille 0, augmentant à 1300, 1800, 2900 et 3500 secondes pour les fenêtres de tailles 3, 5, 8 et 10 respectivement. Ceci voudrait donc dire que la recherche a posteriori pourrait demander jusqu'à six fois plus de temps de calcul pour des résultats non significatifs.

| Corpus | Dist | Ordre | op | mod | #mov | #rep | #mer | #spl | B |
|--------|------|---------|-----|-----|------|------|------|------|-------|
| 2009 | | Moses | | | | | | | 23.83 |
| | | Glouton | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | Oui | Oui | 545 | 509 | 49 | 353 | 40 | 226 | 23.89 |
| | Oui | Non | 524 | 439 | 62 | 298 | 159 | 112 | 23.85 |
| | Non | Oui | 491 | 450 | 36 | 326 | 60 | 162 | 23.90 |
| | Non | Non | 599 | 441 | 80 | 293 | 87 | 287 | 23.81 |
| 2010 | | Moses | | | | | | | 24.36 |
| | | Glouton | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | Oui | Oui | 545 | 523 | 39 | 332 | 44 | 230 | 24.28 |
| | Oui | Non | 480 | 373 | 54 | 260 | 153 | 96 | 24.34 |
| | Non | Oui | 525 | 477 | 39 | 311 | 86 | 177 | 24.29 |
| | Non | Non | 649 | 498 | 81 | 304 | 85 | 308 | 24.30 |
| 2011 | | Moses | | | | | | | 25.22 |
| | | Glouton | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | Oui | Oui | 616 | 571 | 57 | 372 | 47 | 281 | 25.14 |
| | Oui | Non | 643 | 510 | 78 | 358 | 196 | 125 | 25.17 |
| | Non | Oui | 551 | 514 | 53 | 335 | 70 | 197 | 25.17 |
| | Non | Non | 721 | 552 | 95 | 351 | 104 | 329 | 25.15 |

TABLE 14: Effet de la distance et de l'ordre pour la méthode d'évaluation par compte du modèle de collocations (colloc)

Les résultats de la table 14 ne sont pas très concluants. Il est difficile de détecter une tendance dans les résultats. Ceci est particulièrement entre les corpus de test newstest2009 et newstest2010 pour lesquels les scores BLEUs semblent s'inverser dépendamment des paramètres utilisés. Dans l'ensemble, on ne semble pas améliorer les résultats par rapport à la traduction initiale ou à la traduction produite par l'algorithme glouton de base, sauf pour newstest2009. Certaines tendances semblent ressortir du nombre d'opérations par contre. Par exemple, utiliser la distance et ne pas utiliser l'ordre semble produire beaucoup moins de *split* et beaucoup plus de *merge*. Aussi, le fait que ne pas utiliser l'ordre augmente beaucoup le nombre de *move*. Ce dernier résultat est probablement dû à un grand nombre d'inversions entre deux segments successifs qui pourraient se produire lors de la recherche a posteriori. Une dernière tendance est le grand nombre de *split* lorsque les deux paramètres sont identiques, plus particulièrement lorsque

les deux sont à Non.

Méthode d'évaluation par somme des occurrences

Regardons ensuite le modèle de collocations avec la méthode d'évaluation par somme. Cette version de l'algorithme est un peu plus avancée que la méthode par compte, mais reste très simple tout de même. On devrait donc s'attendre à une très légère amélioration par rapport au modèle par compte.

| Corpus | Taille | op | mod | #mov | #rep | #mer | #spl | B |
|--------|---------|------|------|------|------|------|------|-------|
| 2009 | Moses | | | | | | | 23.83 |
| | Glouton | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | 0 | 2483 | 2482 | 1318 | 5099 | 1940 | 6024 | 18.80 |
| | 3 | 1975 | 1805 | 686 | 1694 | 730 | 2096 | 23.49 |
| | 5 | 2200 | 2018 | 717 | 2379 | 1289 | 3067 | 23.05 |
| | 8 | 2210 | 2097 | 637 | 2870 | 1200 | 2640 | 21.05 |
| | 10 | 2250 | 2173 | 656 | 3789 | 1475 | 3374 | 22.33 |
| 2010 | Moses | | | | | | | 24.36 |
| | Glouton | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | 0 | 2455 | 2454 | 1328 | 5012 | 1805 | 6011 | 18.61 |
| | 3 | 1937 | 1771 | 674 | 1589 | 756 | 2008 | 23.60 |
| | 5 | 2201 | 1993 | 747 | 2303 | 1266 | 2843 | 23.03 |
| | 8 | 2172 | 2050 | 636 | 2703 | 1117 | 2410 | 22.11 |
| | 10 | 2245 | 2167 | 651 | 3531 | 1467 | 3220 | 22.45 |
| 2011 | Moses | | | | | | | 25.22 |
| | Glouton | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | 0 | 2955 | 2952 | 1581 | 6094 | 2234 | 7253 | 19.67 |
| | 3 | 2309 | 2085 | 755 | 1954 | 899 | 2311 | 24.97 |
| | 5 | 2620 | 2404 | 833 | 2788 | 1542 | 3400 | 24.34 |
| | 8 | 2600 | 2456 | 726 | 3305 | 1356 | 2872 | 22.91 |
| | 10 | 2693 | 2591 | 693 | 4203 | 1700 | 3811 | 24.01 |

TABLE 15: Effet de la taille de la fenêtre pour la méthode d'évaluation par somme du modèle de collocations (colls)

| Corpus | Dist | Ordre | op | mod | #mov | #rep | #mer | #spl | B |
|--------|---------|-------|------|------|------|------|------|------|-------|
| 2009 | Moses | | | | | | | | 23.83 |
| | Glouton | | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | Oui | Oui | 2210 | 2097 | 637 | 2870 | 1200 | 2640 | 21.05 |
| | Oui | Non | 1061 | 958 | 66 | 700 | 190 | 531 | 23.83 |
| | Non | Oui | 2073 | 1963 | 299 | 2564 | 854 | 2039 | 23.22 |
| | Non | Non | 1249 | 1068 | 183 | 817 | 268 | 688 | 23.81 |
| 2010 | Moses | | | | | | | | 24.36 |
| | Glouton | | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | Oui | Oui | 2172 | 2050 | 636 | 2703 | 1117 | 2410 | 22.10 |
| | Oui | Non | 1023 | 922 | 72 | 666 | 195 | 533 | 24.27 |
| | Non | Oui | 2052 | 1946 | 348 | 2415 | 869 | 2038 | 23.29 |
| | Non | Non | 1250 | 1058 | 209 | 777 | 281 | 700 | 24.25 |
| 2011 | Moses | | | | | | | | 25.22 |
| | Glouton | | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | Oui | Oui | 2600 | 2456 | 726 | 3305 | 1356 | 2872 | 22.91 |
| | Oui | Non | 1193 | 1091 | 88 | 768 | 210 | 610 | 25.28 |
| | Non | Oui | 2447 | 2329 | 375 | 2888 | 1075 | 2357 | 24.67 |
| | Non | Non | 1486 | 1256 | 234 | 927 | 358 | 773 | 25.25 |

TABLE 16: Effet de la distance et de l'ordre pour la méthode d'évaluation par somme du modèle de collocations (colls)

Contrairement à notre hypothèse originale, les résultats pour le modèle de collocations par somme sont inférieurs à ceux pour le modèle de collocations par compte. Ceci est démontré autant en modifiant les paramètres de taille que de distance et d'ordre. Il semblerait donc que cette méthode d'évaluation ne soit pas adéquate pour le modèle.

Méthode d'évaluation par probabilité

Regardons finalement la méthode d'évaluation par probabilité. Celle-ci est légèrement plus complexe que les deux autres, et c'est pourquoi nous croyons qu'elle devrait donner de meilleurs résultats que les deux autres.

| Corpus | Taille | op | mod | #mov | #rep | #mer | #spl | B |
|--------|---------|-----|-----|------|------|------|------|-------|
| 2009 | Moses | | | | | | | 23.83 |
| | Glouton | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | 0 | 377 | 324 | 83 | 184 | 53 | 106 | 23.90 |
| | 3 | 777 | 509 | 106 | 306 | 87 | 513 | 23.84 |
| | 5 | 505 | 444 | 71 | 299 | 57 | 186 | 23.84 |
| | 8 | 668 | 453 | 136 | 265 | 84 | 362 | 23.85 |
| | 10 | 586 | 524 | 102 | 376 | 76 | 163 | 23.76 |
| 2010 | Moses | | | | | | | 24.36 |
| | Glouton | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | 0 | 330 | 275 | 76 | 150 | 52 | 98 | 24.32 |
| | 3 | 803 | 532 | 119 | 263 | 104 | 557 | 24.35 |
| | 5 | 524 | 459 | 62 | 283 | 66 | 189 | 24.27 |
| | 8 | 669 | 473 | 141 | 247 | 85 | 389 | 24.23 |
| | 10 | 579 | 518 | 97 | 344 | 85 | 174 | 24.32 |
| 2011 | Moses | | | | | | | 25.22 |
| | Glouton | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | 0 | 424 | 369 | 98 | 210 | 56 | 123 | 25.17 |
| | 3 | 930 | 622 | 136 | 345 | 102 | 621 | 25.19 |
| | 5 | 620 | 529 | 72 | 354 | 76 | 241 | 25.17 |
| | 8 | 850 | 593 | 175 | 310 | 103 | 461 | 25.08 |
| | 10 | 673 | 587 | 129 | 400 | 91 | 213 | 25.20 |

TABLE 17: Effet de la taille de la fenêtre pour la méthode d'évaluation par probabilité du modèle de collocations (collp)

Les résultats pour la fenêtre sont encore une fois ici peu concluants. La tendance générale est que le modèle de collocations semble diminuer le score par rapport à Moses et à la recherche locale sans ce modèle, sauf dans certains corpus de test spécifiques. La méthode d'évaluation par probabilité performe donc moins bien qu'attendu, n'ayant presque pas de différence avec les deux autres méthodes d'évaluation. Le temps d'exécution est aussi proportionnel avec la taille de la fenêtre et ressemble beaucoup plus à la méthode d'évaluation par compte que celle par somme.

| Corpus | Dist | Ordre | op | mod | #mov | #rep | #mer | #spl | B |
|--------|------|---------|-----|-----|------|------|------|------|-------|
| 2009 | | Moses | | | | | | | 23.83 |
| | | Glouton | 815 | 687 | 37 | 570 | 243 | 206 | 23.75 |
| | Oui | Oui | 668 | 453 | 136 | 265 | 84 | 362 | 23.85 |
| | Oui | Non | 648 | 478 | 71 | 280 | 67 | 387 | 23.87 |
| | Non | Oui | 571 | 485 | 76 | 315 | 83 | 212 | 23.85 |
| | Non | Non | 785 | 657 | 63 | 522 | 211 | 217 | 23.78 |
| 2010 | | Moses | | | | | | | 24.36 |
| | | Glouton | 796 | 659 | 42 | 511 | 252 | 223 | 24.34 |
| | Oui | Oui | 669 | 473 | 141 | 247 | 85 | 389 | 24.23 |
| | Oui | Non | 653 | 492 | 79 | 256 | 69 | 389 | 24.32 |
| | Non | Oui | 591 | 503 | 86 | 293 | 107 | 229 | 24.29 |
| | Non | Non | 782 | 633 | 62 | 468 | 242 | 220 | 24.34 |
| 2011 | | Moses | | | | | | | 25.22 |
| | | Glouton | 866 | 730 | 45 | 570 | 266 | 246 | 25.21 |
| | Oui | Oui | 850 | 593 | 175 | 310 | 103 | 461 | 25.08 |
| | Oui | Non | 744 | 536 | 90 | 299 | 71 | 441 | 25.23 |
| | Non | Oui | 659 | 557 | 109 | 344 | 98 | 261 | 25.14 |
| | Non | Non | 853 | 685 | 65 | 543 | 242 | 240 | 25.24 |

TABLE 18: Effet de la distance et de l'ordre pour la méthode d'évaluation par probabilité du modèle de collocations (collp)

Les résultats pour les paramètres de distance et d'ordre sont similaires à ceux des autres résultats. Mettre les deux paramètres à Faux semble produire les meilleurs résultats et le paramètre d'ordre semble avoir le plus d'impact sur les résultats. On voit une relation entre le nombre d'opérations *move* et le score final produit. Lorsque le nombre de *move* est plus grand, comme dans le cas de Oui-Oui particulièrement, le score produit est plus bas. Ceci pourrait être expliqué par un plus grand nombre d'inversions entre deux segments adjacents qui étaient correctement placés par Moses.

Pour compléter notre comparaison avec Moses, nous avons ici aussi analysé manuellement un jeu de phrases sélectionnées aléatoirement. Nous avons utilisé les traductions produites avec le modèle de collocations en utilisant la méthode d'évaluation par probabilité avec une fenêtre de taille 8, en considérant les collocations à une distance inférieure à 8 aussi, mais en conservant l'ordre des mots. Les résultats présents dans la table 19

indiquent, tout comme pour les deux analyses précédentes, que notre recherche locale semble produire de meilleurs résultats que Moses, malgré l’analyse de BLEU. On remarque ici que la différence est plus grande que pour l’analyse sans modèle ajouté, mais plus faible que pour celle avec le modèle de langue inverse. Ceci indiquerait donc que le modèle de collocations n’est pas aussi efficace que le modèle de langue inverse pour la recherche locale. L’analyse complète est disponible dans l’annexe A.1.

| Test | Moses | Locale | Équivalent | Total |
|--------------|-------|--------|------------|-------|
| newstest2009 | 16 | 27 | 7 | 50 |
| newstest2010 | 18 | 23 | 9 | 50 |
| newstest2011 | 16 | 22 | 12 | 50 |
| Total | 50 | 72 | 28 | 150 |

TABLE 19: Comparaison humaine entre Moses et la recherche locale avec le modèle de collocations (collp) sur un échantillon aléatoire

Retour sur le modèle de collocations

Les points importants à noter par rapport au modèle de collocations sont d’une part le fait qu’il peut améliorer les traductions produites par rapport à la recherche locale, quoique dans certains cas seulement et d’autre part, que le bon choix des paramètres est très important. Malgré le fait que la méthode d’évaluation par probabilité ne semble pas donner d’aussi bons résultats qu’escomptés, l’utilisation de ce modèle de collocations pour la recherche locale semble améliorer les résultats. Un autre grand point est le même qui a été énoncé pour le modèle de langue inverse, soit le fait que ce genre de modèles semblent très variable par rapport à la traduction à produire. Nous avons souvent vu un corpus obtenir une augmentation dans le score BLEU alors que les deux autres voyaient leur score diminuer. Pour ce qui est des paramètres, voici nos conclusions :

taille La taille de la fenêtre est un paramètre qui peut facilement être modifié dépendamment des besoins, tout comme la taille du n-gramme d’un modèle de langue. Plus la taille est élevée, plus le temps de calcul est long. Ceci peut avoir un impact sur la taille à choisir dépendamment des besoins de l’engin de traduction. Malheureusement, nous n’avons pas assez de données concluantes pour définir une longueur optimale à utiliser.

ordre L'ordre semble être une option qu'il faut mettre à Faux dans notre modèle. Ceci produit donc un modèle qui regarde les collocations entre deux mots peu importe l'ordre d'apparition de ceux-ci dans la phrase. On priorise donc les mots qui sont souvent présents ensemble dans des phrases. Ceci peut avoir un impact sur le segment à choisir, et pourrait aider à uniformiser un peu les traductions. Mettre ce paramètre à Vrai coupe essentiellement de moitié cet effet pour augmenter l'effet de l'ordre des mots, ce qui est déjà implémenté dans les modèles de langues, et donc pas très intéressant comparativement à l'autre option.

distance Le paramètre de distance, qui sert à soit évaluer les collocations à une distance fixe donnée (Vrai) ou toutes les collocations jusqu'à cette même distance (Faux), semble produire de meilleurs résultats lorsqu'il est à Faux. Ceci est en lien avec le paramètre d'ordre et l'idée d'utiliser le modèle de collocations pour prioriser des mots qui se retrouvent souvent dans les mêmes phrases et donc augmenter l'uniformisation des traductions dans un texte complet.

5.6 Retour sur les résultats

D'une part, la recherche locale ne semble pas améliorer les traductions générées par Moses, comme il a été décrit dans Marie [9], et contrairement à ce qui a été trouvé par Langlais et al. [7]. La meilleure explication à ceci est probablement le fait que Moses (anciennement Pharaoh) a beaucoup évolué entre les travaux de Langlais en 2007 et les travaux de Marie en 2012. L'engin est maintenant capable de produire de meilleures traductions de base, ce qui complique le travail pour une recherche locale a posteriori.

L'utilisation de BLEU pour analyser les résultats automatiquement semble aussi être un problème. Nous avons montré que pour une sélection aléatoire de phrases, la recherche locale semble produire de meilleures traductions que Moses, mais ceci n'est pas représenté dans le score BLEU. Nous n'avons utilisé qu'une seule phrase de référence pour l'analyse de BLEU, ceci a pour conséquence que des traductions équivalentes selon un humain peuvent avoir des scores très différents selon BLEU. Utiliser plusieurs phrases de références pourrait réduire cette variance.

Le réglage des poids à lui seul apporte une grande modification aux traductions comme montré dans la section 5.3. Il est clair que la recherche locale ne peut pas être utilisée avec les mêmes paramètres que Moses car

on n'obtient aucune modification en se fiant à cette fonction de coût. Il faut donc recalibrer les poids et idéalement ajouter des modèles de plus à la fonction objectif pour améliorer la traduction.

D'autre part, les modèles que nous avons implémentés, soit le modèle de langue inverse et le modèle de collocations, peuvent améliorer les résultats par rapport à la recherche locale sans ces modèles, mais très rarement améliorer les résultats par rapport à la traduction originale de Moses. Malheureusement, les résultats semblent dépendre beaucoup du corpus à traduire et des paramètres, ce qui complexifie l'utilisation de ces modèles pour un engin de traduction automatique.

Le modèle de langue inverse semble être une addition utile lorsque l'on utilise une recherche locale a posteriori et le modèle de collocations semble un peu trop chaotique pour l'instant pour pouvoir l'implémenter de façon permanente. Il serait par contre intéressant de voir les résultats que Moses pourrait produire avec un modèle de collocations plus avancé. Nous croyons que le modèle de collocations doit être utilisé comme un outil d'uniformisation plutôt qu'un outil de recherche de traductions comme les modèles de langues. Ceci peut être accompli en augmentant la taille de la fenêtre, en utilisant toutes les collocations jusqu'à cette taille et en observant autant avant et après le mot courant. Une optimisation de ce modèle pourrait aussi aider à réduire le temps de chargement du modèle dans Moses et le temps de traduction qui est grandement augmenté par rapport à la recherche locale simple.

Somme toute, les résultats sont prometteurs pour la recherche locale. Malgré les scores BLEU qui semblent indiquer une dégradation de la traduction, les analyses manuelles semblent indiquer pour leur part une amélioration de la traduction, et ceci pour toutes les configurations analysées.

6 Perspectives

Comme les résultats l'ont démontré, nous sommes encore très loin d'obtenir une traduction équivalente à celle qu'un humain pourrait produire. C'est pourquoi nous proposons plusieurs approches qui pourraient être envisagées dans la recherche future pour ce domaine, autant pour la recherche locale a posteriori que la création d'une traduction par faisceau comme le fait l'engin de traduction Moses.

6.1 Métaheuristiques

Le premier point abordé par notre recherche était les algorithmes de recherche locale. Les deux algorithmes étudiés ici sont l'algorithme glouton et une variante de celui-ci, le glouton randomisé. Nous proposerons donc plusieurs autres algorithmes qu'il pourrait être intéressant ou non d'utiliser dans le contexte de recherche locale a posteriori pour la traduction automatique statistique.

L'algorithme glouton randomisé qui a été utilisé pour cette recherche ne fait qu'accepter des solutions améliorant la fonction de score. Une première amélioration possible pourrait être d'accepter certaines solutions dégradant légèrement cette fonction pour agrandir d'avantage l'espace de recherche. Le glouton randomisé explorant une plus grande région de l'espace de recherche, il a le potentiel de produire de meilleurs résultats sans trop augmenter le temps d'exécution. Malheureusement, ce potentiel est dû à la partie aléatoire de celui-ci et ceci peut donc lui faire produire de moins bonnes traductions que son équivalent non-aléatoire, quoique ce soit peu probable. Plusieurs autres algorithmes de recherche locale pourraient accroître davantage l'exploration dans l'espace de recherche.

Le recuit simulé est un algorithme souvent utilisé en recherche locale. Celui-ci consiste à accepter des solutions si la différence entre l'évaluation de l'état précédent et l'évaluation de l'état présent est soit positif dans la direction de l'optimisation, soit à l'intérieur d'un seuil calculé à partir d'une fonction mathématique. Cette fonction mathématique accepte ou non une nouvelle solution selon la différence de score et un paramètre température. Une température plus élevée acceptera des solutions dégradant beaucoup la fonction de score, alors qu'une température de zéro produit un algorithme glouton randomisé, prenant seulement des solutions améliorant la fonction

de score. De plus, il utilise aussi un cycle de refroidissement pour faire varier la température. Le cycle le plus simple est de faire diminuer la température, ceci réduit la taille des sauts dans l'espace de recherche et tente de forcer l'algorithme dans un optimum local. Certains cycles peuvent faire augmenter à nouveau la température pour tenter de faire sortir l'algorithme d'un optimum local. Cet algorithme explore donc une plus grande région de l'espace de recherche. Il produit généralement des meilleures solutions, mais requiert plus de temps que l'algorithme glouton. Le principal inconvénient dans l'implémentation de cet algorithme est le paramétrage des hyperparamètres de température et de cycle de refroidissement. Ces hyperparamètres peuvent être très compliqués à bien paramétrer, ce qui fait parfois apparaître le recuit simulé comme un mauvais algorithme. Cet algorithme pourrait bien s'adapter au problème de recherche locale en traduction automatique.

Un autre algorithme de recherche locale pouvant bien se prêter au problème courant est la recherche taboue. Cet algorithme se base sur un algorithme plus simple, par exemple l'algorithme de glouton randomisé, et y ajoute une contrainte qu'un état ou une transformation ne peut se reproduire avant un certain nombre d'itérations. L'utilisation d'une liste taboue permet de retenir des états déjà parcourus où l'on ne veut pas retourner ou des transformations que l'on ne veut pas effectuer à l'inverse (détruisant ainsi l'effet de la transformation). La recherche taboue utilisera toujours une solution du voisinage satisfaisant les contraintes de la liste taboue, qu'elle soit meilleure ou non que la solution courante. Ceci force la recherche à parfois effectuer de plus grand sauts dans l'espace de recherche. Le principal paramètre de cette fonction est la longueur de la liste taboue, qui influence sur la taille de la fenêtre de recherche dans l'espace de solutions. Les deux approches, par état et par transformation, peuvent s'appliquer au problème courant.

Les algorithmes de recherche locale itérée et de recherche locale à voisinage variable ne semblent pas être adaptées à ce problème. La recherche locale itérée tente de trouver une solution avec un algorithme de recherche locale simple, comme ceux décrits plus hauts, avant d'effectuer un grand saut dans l'espace de recherche et refaire une recherche locale dans cette nouvelle région. On fait plusieurs itérations de cette façon et on conserve la meilleure solution trouvée dans toutes les itérations effectuées. La recherche locale à voisinage variable de son côté implémente une recherche locale simple, mais ajoute des opérations sur le voisinage qui permettent de plus grand sauts dans l'espace de recherche. Dans les deux cas, ces

algorithmes nécessitent une opération de saut dans l'espace de recherche qui est difficilement implémentable avec la traduction automatique. Ils ont généralement des temps d'exécution beaucoup plus grands que les autres algorithmes parce qu'on doit faire plusieurs recherches locales. Considérant le fait que notre recherche locale doublait effectivement le temps de calcul pour la recherche d'une traduction, l'utilisation d'un tel algorithme ferait grandement augmenter les temps de calculs sans améliorer la qualité du résultat obtenu proportionnellement. C'est une autre pour laquelle ce genre d'algorithme ne serait pas très efficace pour la recherche locale a posteriori.

Une autre classe de métaheuristiques étudie les algorithmes à mémoire adaptative et les algorithmes à essais de particules. Un excellent exemple des algorithmes à mémoire adaptative est l'algorithme de colonies de fourmis, qui est d'ailleurs utilisé en désambiguisation lexicale. Les algorithmes à mémoire adaptative utilisent une mémoire centrale pour retenir de l'information sur les itérations précédentes et donc guider l'itération suivante. Ce type d'algorithme pourrait être utilisé pour l'amélioration a posteriori de la recherche locale, mais il requiert souvent beaucoup plus de temps, du fait qu'il faut effectuer plusieurs fois une recherche locale, et ne produirait probablement pas de meilleurs résultats. Les algorithmes à essais de particules de leur côté vont observer plusieurs solutions (particules) en parallèle à l'intérieur d'un groupe et à chaque itération déplacent les particules en fonction non seulement de la meilleure solution pour cette particule, mais aussi des meilleures solutions trouvées par les autres particules. Encore une fois, ceci ne semble pas adapté pour la traduction automatique à cause du temps d'exécution et des solutions qui ne seraient probablement que légèrement supérieures aux meilleures solutions trouvées par les algorithmes simples de recherche locale.

Une dernière possibilité est l'utilisation d'algorithmes génétique et mémétique. La base de ces algorithmes réside dans le fait qu'on utilise des solutions parentes pour générer des solutions enfants à l'aide d'opérations de croisement et d'opérations de mutations. À chaque itération, l'algorithme prend plusieurs parents, souvent seulement deux, et les croisent pour générer une ou plusieurs nouvelles solutions. Ces nouvelles solutions sont ensuite mutées, permettant ainsi d'explorer un peu plus l'espace de recherche. L'algorithme mémétique ajoute une étape de recherche locale à partir de la solution mutée avant de passer à la prochaine itération. Ces algorithmes auraient pu être intéressants à explorer dans le domaine de la traduction automatique, si ce n'était de leur temps de calcul qui lui aussi est sou-

vent un multiple d'une recherche locale simple. Ce genre d'algorithme peut gagner beaucoup de temps par l'utilisation du parallélisme pour générer les solutions enfants. Malheureusement, dans l'état actuel de la traduction automatique statistique, la recherche est souvent déjà parallélisée au maximum en utilisant un fil d'exécution par «phrase» à traduire. Leur principal inconvénient est l'élaboration des opérations de croisement et de mutation qui peuvent facilement devenir très complexes.

6.2 Modèle de langue inverse

Les résultats prometteurs du modèle de langue inverse couplé à la recherche locale nous amènent à penser qu'on devrait poursuivre dans cette direction. Malheureusement, ce modèle étant entièrement basé sur le modèle de langue traditionnel, il dépend aussi grandement de ce dernier. Ce modèle n'étant pas adapté à la traduction par faisceau, tel Moses, nous déconseillons cette ouverture pour un tel engin de traduction, mais proposons de l'utiliser avec d'autres engins de traduction comme la recherche locale a posteriori.

6.3 Modèle de collocations

L'autre point majeur de cette recherche était le modèle de collocations. Ce modèle essentiellement basé sur un modèle de langue à «trous» amène une information différente par rapport à ce dernier. Ce modèle n'amène pas une information grammaticale comme le modèle de langue, mais plutôt une information lexicale. Ceci permet parfois de trouver une meilleure traduction pour un certain segment de la phrase dépendamment du reste de la phrase. Trois méthodes d'évaluation ont été implémentées, dont deux simples (Compte et Somme) et une un peu plus complexe (Prob). Parmi les trois, la méthode probabiliste offre le meilleur potentiel de trouver de meilleurs résultats en général. Il serait intéressant d'avancer la recherche dans cette direction en amenant de nouvelles méthodes d'évaluation.

Un des points majeurs dans notre modèle de collocations est l'absence de lissage ou d'interpolation. Un lissage pourrait améliorer un peu les résultats, surtout avec l'évaluation probabiliste ou une autre évaluation de ce type. Le modèle ressemblant grandement à un modèle de langue côté im-

plémentation, il serait relativement facile d'ajouter un lissage similaire à ce qui est utilisé dans les modèles de langue. Une interpolation pourrait aussi être utilisée. Dans les tests, nous avons utilisé une interpolation «extrême» en mettant le paramètre de distance à faux. Ceci à l'effet de faire une interpolation uniforme sur toutes les occurrences plus rapprochées que celle cherchée. Ceci pourrait être amélioré en ajoutant une fonction donnant un poids plus élevé à des distances avoisinant celle recherchée et moindre pour les distances plus éloignées.

Les statistiques sur le modèle de collocations que nous avons montrées dans la table 12 indiquent un très grand nombre de collocations qui n'ont été vues qu'une seule fois. On peut se demander à quel point ces collocations aident notre modèle ou au contraire lui nuisent. Il serait aussi intéressant de refaire nos tests en supprimant ces collocations pour voir l'impact autant sur le score BLEU, que sur la qualité des traductions et le temps de calcul ajouté. Il pourrait aussi être intéressant de construire un filtre pour retirer les ponctuations et autres éléments de la langue qui n'ont pas vraiment leur place dans un tel modèle.

Un point à noter pour ce modèle est le fait qu'il pourrait être utilisé directement avec l'engin de traduction Moses. Il n'y aurait pas de problème à l'utiliser avec les traductions partielles en modifiant l'implémentation pour respecter les modèles de Moses. Ceci pourrait aider à trouver de meilleures traductions pour des segments plus loin dans la phrase.

6.4 Autres modèles possibles

Outre les modèles de langue inverse et de collocation étudiés, il pourrait être intéressant de trouver des modèles qui seraient propres à la recherche locale a posteriori. Ces modèles utiliseraient le fait que la traduction est déjà complétée pour aller chercher une information supplémentaire. Deux modèles sont envisageables ici : un modèle par recherche d'information à partir du corpus initial et un modèle de traduction basé sur l'ensemble des phrases qui viennent d'être traduites.

Le modèle par recherche d'information serait basé sur les engins de recherche d'information. L'idée serait d'importer le corpus de langue source dans un engin de recherche d'information. Ensuite pour chaque traduction que l'engin de recherche locale reçoit, sous le format d'une phrase source et de sa traduction produite par Moses, il utilise l'engin de recherche d'infor-

mation pour trouver les phrases similaires à la phrase source dans le corpus, ainsi que leur traduction avec le corpus parallèle. Il utilise ensuite l'ensemble de ces traductions pour orienter les modifications à faire pour obtenir une similarité semblable du côté de la langue cible. L'idée de similarité entre deux phrases est un point qu'il faudrait étudier en profondeur dans ce cas.

Le modèle de traduction basé sur l'ensemble des phrases qui viennent d'être traduites, de son côté, utiliserait le fait que nous recevons généralement des phrases provenant d'une même source. L'idée serait de créer un modèle de traduction à partir des traductions trouvées par Moses et d'utiliser celui-ci dans la recherche locale pour uniformiser les traductions dans le texte en favorisant la même traduction pour le même segment qui serait répété dans une autre phrase. Ce genre de modèle ne serait utile que si nous avons un assez long texte à traduire, comme quelques milliers de lignes provenant toutes du même endroit, comme une série d'articles de journaux sur le même sujet ou un livre. S'il n'y a pas assez de phrases dans le texte à traduire, les probabilités générées dans la table de traductions seront souvent disproportionnées entre-elles, ce qui n'aiderait pas la traduction. L'idée de ce modèle pourrait aussi être utilisé pour *bootstrapper* Moses directement. On traduit une fois le texte avec Moses pour ensuite créer notre modèle de traduction adapté et retraduire le texte avec Moses. Il pourrait aussi être utilisé phrase par phrase, en modifiant le modèle de traduction adapté après chaque traduction pour mieux orienter la traduction des phrases suivantes dans le texte. Ceci augmenterait énormément le temps de calcul par contre car il ne serait plus possible de paralléliser la traduction et augmenterait énormément la complexité parce qu'au départ les probabilités des traductions seraient disproportionnées entre elles et donc elle n'aideraient à la traduction qu'après un assez grand nombre de phrases traduites. Dans ce cas il faudrait donc augmenter graduellement le poids du modèle de traduction adapté à chaque phrase. L'idée de l'utiliser avec la recherche locale plutôt qu'en *bootstrapping* Moses serait de réduire l'effet des probabilités disproportionnées.

6.5 Autres métriques pour l'analyse

Comme nous l'avons vu dans les résultats, la métrique BLEU ne semble pas être bien adaptée pour évaluer automatiquement les traductions dans notre cas. Particulièrement lorsque nous utilisons une seule phrase de ré-

férence. Il serait donc intéressant de voir si plusieurs phrases de référence donneraient des résultats plus près de la réalité. L'utilisation d'autres métriques pourrait aussi nous donner une meilleure évaluation des nos modèles de traduction. Nous pouvons penser notamment au WER (Word Error Rate) et ROUGE (Recall-Oriented Understudy for Gisting Evaluation) ou à certaines variantes de BLEU tel que NIST-BLEU (National Institute of Standards and Technology) et METEOR (Metric for Evaluation of Translation with Explicit ORdering).

7 Conclusion

Lors de cette recherche, nous avons exploré la possibilité d'utiliser la recherche locale pour améliorer une hypothèse de traduction déjà produite. Ceci a été fait à l'aide de l'algorithme glouton et une variante de ce dernier, l'algorithme glouton aléatoire. De plus, nous avons aussi tenté d'ajouter d'autres modèles pour améliorer la fonction de score du décodeur : un modèle de langue inverse et un modèle de collocations.

Malgré le fait que les résultats obtenus ne semblent pas améliorer la traduction de base, ils invitent à explorer davantage des solutions similaires, surtout concernant le modèle de collocations, et pousser plus loin dans ce champ pour tenter de produire de meilleurs résultats. Les analyses manuelles indiquent par ailleurs une utilité aux modèles ajoutés lors de la traduction automatique que les scores BLEU ne captent pas correctement. Plusieurs métaheuristiques sont aussi à considérer pour bien étudier le problème de l'amélioration de la traduction automatique statistique a posteriori.

A Annexe

A.1 Analyses manuelles

Les analyses manuelles présentées ici montrent pour chaque phrase à traduire les informations suivantes :

SOURCE La phrase source telle qu’entrée dans Moses et dans notre recherche locale ;

MOSES La traduction produite par Moses ;

LOCAL La traduction produite par la recherche locale à partir de la traduction de Moses ;

OPS Le point de départ de la traduction (“KNOWN” si tous les mots sont connus et “UNKNOWN” si au moins un mot est inconnu) et les opérations effectuées par la recherche locale pour produire une nouvelle traduction ainsi que le résultat de la fonction de score pour cette opération entre parenthèses ;

STATS Des statistiques sur la recherche locale :

src le nombre de mots dans la phrase source ;

moses le nombre de mots dans la traduction de Moses ;

local le nombre de mots dans la traduction de la recherche locale ;

views le nombre d’hypothèses analysées lors de la recherche locale ;

worst le pire score de la fonction objectif observé lors de la recherche locale ;

best le meilleur score de la fonction objectif observé lors de la recherche locale (pour une recherche gloutonne non-aléatoire, la dernière opération aura le même score) ;

string changed indication que la phrase a été modifiée (l’analyse ne comporte que des phrases modifiées) ;

MOSES/LOCAL/EQUIV le résultat de l’analyse manuelle, soit la meilleure des deux traductions ou si elle sont équivalentes.

Les analyses étant très volumineuses, seulement un sous-ensemble est présenté ici. Les analyses complètes sont disponibles ici : <http://www-etud.iro.umontreal.ca/~lavoiecs/index.php?page=memoire>.

Recherche locale gloutonne sans modèle ajouté

Corpus de test newstest2009

SOURCE les législateurs espèrent qu'après il sera adopté durant les jours à venir .
MOSES Legislators are hoping that it will be adopted in the coming days.
LOCAL The legislators are hoping that it will be adopted in the coming days.
OPS KNOWN(-8.825) REPLACE(-8.802)
STATS src(13), moses(13), local(14), views(6414), worst(-15.642000), best(-8.802000), string changed, EQUIV

SOURCE les prix ne sont pas non plus des plus bas .
MOSES Prices are not of the lowest.
LOCAL Prices are not the lowest.
OPS KNOWN(-6.956) SPLIT(-6.862)
STATS src(11), moses(7), local(6), views(4725), worst(-13.419000), best(-6.862000), string changed, LOCAL

SOURCE pour un prix de trois quatre mille couronnes , on peut avoir un walkman déjà très convenable .
MOSES For a price of three four thousand kroner, there can be one walkman already very appropriate.
LOCAL For a price of three four thousand kroner, we can have a walkman already very appropriate.
OPS UNKNOWN(-116.011) SPLIT(-115.940)
STATS src(18), moses(18), local(18), views(5983), worst(-120.427000), best(-115.940000), string changed, LOCAL

SOURCE olmert rendrait en échange de la paix presque tous les territoires occupés
MOSES Olmert would make in exchange for peace almost all the occupied territories
LOCAL Olmert would almost all the occupied territories in exchange for peace
OPS KNOWN(-10.872) MOVE(-10.869) REPLACE(-10.610)
STATS src(12), moses(12), local(11), views(7317), worst(-17.904000), best(-10.610000), string changed, MOSES

SOURCE l'index de bourse des technologies nasdaq était à 1548 points avec une baisse de plus de sept points .
MOSES The index of stock exchange technologies NASDAQ was in 1548 points with a drop of more than seven points.
LOCAL The index of stock exchange of technologies NASDAQ was in 1548 points with a drop of more than seven points.
OPS KNOWN(-18.223) REPLACE(-18.182)
STATS src(20), moses(20), local(21), views(7074), worst(-23.918000), best(-18.182000), string changed, LOCAL

SOURCE il est évident que le repos lui a permis de resté brillant , mais pas rouillé .
MOSES It is obvious that the rest has enabled it to remained brilliant, but not rouillé.
LOCAL It is obvious that the rest has enabled him to remained brilliant, but not rouillé.
OPS UNKNOWN(-112.834) REPLACE(-112.628)
STATS src(17), moses(17), local(17), views(7570), worst(-118.378000), best(-112.628000), string changed, LOCAL

SOURCE essayer de ne pas exciter l'enfant avant d'aller dormir .
MOSES Try not to whip up the child before we go to sleep.
LOCAL Try not to annoy the child before we go to sleep.
OPS KNOWN(-10.452) REPLACE(-10.449)
STATS src(12), moses(13), local(12), views(3558), worst(-15.360000), best(-10.449000), string changed, LOCAL

SOURCE le milieu de terrain antonio ribeiro a marqué le but décisif à la 90e minute de jeu .
MOSES The middle ground Antonio Ribeiro marked the winning goal to the 90th minute's game.
LOCAL The middle ground Antonio Ribeiro marked the winning goal the 90th minute's game.
OPS KNOWN(-17.360) REPLACE(-17.349)
STATS src(18), moses(16), local(15), views(6848), worst(-22.272000), best(-17.349000), string changed, MOSES

SOURCE de plus , la facture mentionne aussi des repas , et notre restaurant est fermé depuis bien longtemps .
MOSES Furthermore, the bill also mentions meals, and our restaurant is closed for far longer. '
LOCAL Furthermore, the bill also mentions meals, and our restaurant is closed for much longer. '
OPS KNOWN(-16.495) MERGE(-16.486)
STATS src(20), moses(18), local(18), views(8824), worst(-22.062000), best(-16.486000), string changed, EQUIV

Corpus de test newstest2010

SOURCE heureusement ai @-@ je trouvé l' endroit adéquat en allant prendre le tram .
MOSES Fortunately have I found the right place by going take the tram.
LOCAL Fortunately have I found the right place going take the tram.
OPS KNOWN(-13.722) REPLACE(-13.704)
STATS src(14), moses(13), local(12), views(3746), worst(-18.583000), best(-13.704000), string changed, MOSES

SOURCE je lui ai dit : " si je dois vous incarner , je devrais pouvoir vous voir personnellement ..
MOSES I said to her: "If I have to embody, I should be able to see you personally.
LOCAL I said to her: "If I must embody, I should be able to see you personally.
OPS KNOWN(-15.480) REPLACE(-15.358) MERGE(-15.348)
STATS src(19), moses(21), local(20), views(8776), worst(-22.177000), best(-15.348000), string changed, EQUIV

SOURCE c' est l' affaire du chef du gouvernement fischer et de tout le gouvernement .
MOSES It is a matter of the head of government Fischer and all the government.
LOCAL It is a matter of head of government Fischer and all the government.
OPS KNOWN(-10.340) REPLACE(-10.306)
STATS src(15), moses(15), local(14), views(7234), worst(-15.958000), best(-10.306000), string changed, EQUIV

SOURCE pourtant les parents payaient 1600 couronnes voire plus pour une dose .
MOSES Yet the parents paid 1600 kroner even more for a dose.
LOCAL Yet the parents paid 1,600 kroner even more for a dose.
OPS KNOWN(-14.130) REPLACE(-14.116)
STATS src(12), moses(12), local(12), views(2816), worst(-18.657000), best(-14.116000), string changed, LOCAL

SOURCE affaire liliane bettencourt : françois @-@ marie banier sera jugé pour abus de faiblesse
MOSES Case liliane bettencourt: François-Marie banier will be judged for abuse of weakness
LOCAL Case liliane bettencourt François-Marie banier will be judged for abuse of weakness:
OPS UNKNOWN(-313.796) MOVE(-313.715)
STATS src(14), moses(15), local(15), views(2852), worst(-319.405000), best(-313.715000), string changed, MOSES

SOURCE le rêve est un état tout à fait différent , écrit @-@ il en décrivant ce qu' il a vécu .
MOSES The dream is a state quite different, he writes in describing what he has experienced.
LOCAL The dream is a state quite different, he writes in describing what it has experienced.
OPS KNOWN(-14.552) MERGE(-14.499) REPLACE(-14.459)
STATS src(21), moses(17), local(17), views(13651), worst(-19.712000), best(-14.459000), string changed, MOSES

SOURCE la ligue des champions est devenue une source de revenus pour les clubs , depuis sa naissance en 1992 .
MOSES The Champions League has become a source of income for clubs, since its inception in 1992.
LOCAL The Champions League has become a source of income for clubs, since its birth in 1992.
OPS KNOWN(-11.662) SPLIT(-11.658)
STATS src(20), moses(18), local(18), views(7357), worst(-18.506000), best(-11.658000), string changed, MOSES

SOURCE les projets qui ont été révélés paraissent prometteurs .
MOSES Projects which were revealed appear promising.
LOCAL The projects which were revealed appear promising.
OPS KNOWN(-8.180) REPLACE(-8.177)
STATS src(9), moses(7), local(8), views(3638), worst(-13.146000), best(-8.177000), string changed, LOCAL

SOURCE quelque 50.000 personnes , selon les organisateurs , étaient descendues dans les rues en australie .
MOSES Some 50 000 people, according to the organisers, were taken to the streets in Australia.
LOCAL Some 50 000 people, according to the organisers, were in the streets in Australia.
OPS KNOWN(-13.645) MERGE(-13.484)
STATS src(16), moses(18), local(17), views(3836), worst(-19.541000), best(-13.484000), string changed, LOCAL

Corpus de test newstest2011

SOURCE c' est le serveur de la bbc qui a transmis cette information .
MOSES It is the BBC server which passed this information.
LOCAL It is the BBC server which has forwarded this information.
OPS KNOWN(-9.825) SPLIT(-9.780)
STATS src(13), moses(10), local(11), views(5799), worst(-16.816000), best(-9.780000), string changed, LOCAL

SOURCE c' est ainsi qu' est la maison casa battlo à barcelone de l' architecte antoni gaudi .
MOSES That is how is the House Casa battlo in Barcelona the architect Antoni sit.
LOCAL That is how is the home Casa battlo in Barcelona the architect Antoni sit.
OPS UNKNOWN(-117.074) REPLACE(-117.053)
STATS src(17), moses(15), local(15), views(5302), worst(-122.115000), best(-117.053000), string changed, EQUIV

SOURCE la victoire lui a apporté plus de popularité et d' être connu à libérec et ses environs .
MOSES The victory has brought more popularity and be seen to libérec and its surrounding area.
LOCAL The victory has brought more popularity and be seen to libérec and its environs.
OPS UNKNOWN(-114.808) MERGE(-114.759)
STATS src(18), moses(16), local(15), views(5561), worst(-119.768000), best(-114.759000), string changed, MOSES

SOURCE en comparaison avec le monde entier la situation en république tchèque ne s' améliore pas du tout .
MOSES In comparison with the world the situation in the Czech Republic does not improve at all.
LOCAL In comparison with the world the situation in the Czech Republic has not improved at all.
OPS KNOWN(-10.869) REPLACE(-10.780)
STATS src(18), moses(17), local(17), views(8206), worst(-16.415000), best(-10.780000), string changed, MOSES

SOURCE " nous travaillons dur , mais c' est gratifiant " , affirme le recycleur de déchets .
MOSES 'We are working hard, but it is gratifying', says the recycleur of waste.
LOCAL 'We are working hard, but it is gratifying', says the recycleur waste.
OPS UNKNOWN(-111.526) REPLACE(-111.464)
STATS src(17), moses(18), local(17), views(6582), worst(-116.943000), best(-111.464000), string changed, MOSES

SOURCE medvedev a évoqué plusieurs fois la lutte contre la corruption mais il a admis le peu de progrès .
MOSES Medvedev has repeatedly addressed the fight against corruption but it has admitted the little progress.
LOCAL Medvedev mentioned several times the fight against corruption but it has admitted the little progress.
OPS KNOWN(-12.892) SPLIT(-12.863)
STATS src(19), moses(16), local(16), views(7505), worst(-18.685000), best(-12.863000), string changed, LOCAL

SOURCE le dimanche passé , le ppc a obtenu 384.091 votes .
MOSES Sunday past, the PPC obtained 384.091 votes.
LOCAL Sunday past the PPC obtained 384.091 votes.
OPS UNKNOWN(-208.898) REPLACE(-208.857)
STATS src(11), moses(9), local(8), views(4202), worst(-213.538000), best(-208.857000), string changed, MOSES

Recherche locale gloutonne avec le modèle de langue inverse

Corpus de test newstest2009

SOURCE l' aac (format utilisé par la firme apple et ses ipods) , quant à lui , est moins courant .
MOSES The PCA (format used by the firm apple and its iPods taken from them), for its part, is less course.
LOCAL The PCA (format used by the firm apple and its iPods taken from them), for its part, is less aware.
OPS KNOWN(-24.155) REPLACE(-24.110)
STATS src(22), moses(25), local(25), views(7242), worst(-29.819000), best(-24.110000), string changed, LOCAL

SOURCE les passagers peuvent recevoir leur café et leur journal à l'&apost; embarquement .
 MOSES Passengers can receive their coffee and their newspaper at boarding.
 LOCAL Passengers can receive their coffee and their newspaper to boarding.
 OPS KNOWN(-12.317) SPLIT(-12.312)
 STATS src(13), moses(11), local(11), views(3403), worst(-18.563000), best(-12.312000), string changed, MOSES

SOURCE dont les ordres semblent avoir été semblables : et que ça saute avec ce deuxième but !
 MOSES Whose orders seem to have been similar: and that it is with this second aim!
 LOCAL Whose orders seem to have been similar: and that it springs with this second aim!
 OPS KNOWN(-20.272) REPLACE(-20.188)
 STATS src(17), moses(17), local(17), views(4756), worst(-25.621000), best(-20.188000), string changed, LOCAL

SOURCE eric smith fit une autre passe et l'&apost; a interceptée .
 MOSES Eric Smith had another happening and has stopped.
 LOCAL Eric Smith did another happening and has stopped.
 OPS KNOWN(-14.514) REPLACE(-14.500)
 STATS src(11), moses(9), local(9), views(2734), worst(-19.839000), best(-14.500000), string changed, LOCAL

SOURCE microsoft essaiera sûrement de lancer au même moment windows 7 et windows mobile 7 .
 MOSES Microsoft will surely to launch at the same time windows windows 7 and mobile 7.
 LOCAL Microsoft will surely to launch at the same time windows 7 windows and mobile 7.
 OPS KNOWN(-20.264) MOVE(-20.176)
 STATS src(15), moses(16), local(16), views(3726), worst(-25.757000), best(-20.176000), string changed, LOCAL

SOURCE libre opinion - le discours orwellien de barrick gold
 MOSES Free opinion - the speech of barrick gold Orwellian
 LOCAL Free opinion - the speech of gold barrick Orwellian
 OPS UNKNOWN(-112.058) MOVE(-112.045)
 STATS src(9), moses(9), local(9), views(1838), worst(-117.123000), best(-112.045000), string changed, MOSES

SOURCE de plus , la facture mentionne aussi des repas , et notre restaurant est fermé depuis bien longtemps . "
 MOSES Furthermore, the bill also mentions meals, and our restaurant is closed for far longer. '
 LOCAL Furthermore, the bill also mentions meals, and our restaurant is closed for much longer. '
 OPS KNOWN(-18.475) MERGE(-18.398) SPLIT(-18.371)
 STATS src(20), moses(18), local(18), views(13054), worst(-24.436000), best(-18.371000), string changed, EQUIV

SOURCE est @-@ ce que la vue d'&apost; une vierge marie de meilleure qualité provoquerait de meilleurs résultats ?
 MOSES Does the view of a Virgin Mary better would create better results?
 LOCAL Does the view of a Virgin Mary better would better results?
 OPS KNOWN(-16.847) REPLACE(-16.770)
 STATS src(18), moses(13), local(12), views(8104), worst(-23.432000), best(-16.770000), string changed, MOSES

SOURCE l'&apost; automobile intelligente imite le vol du bourdon
 MOSES The intelligent car theft of Bourdon imitate
 LOCAL The intelligent car theft of Bourdon emulating
 OPS KNOWN(-12.307) REPLACE(-12.263)
 STATS src(8), moses(7), local(7), views(1082), worst(-18.448000), best(-12.263000), string changed, MOSES

SOURCE la dépense de la deuxième moitié des 700 milliards dépend d'&apost; un nouveau décret du congrès .
 MOSES The spending of the second half of the 700 billion depends on a new decree of Congress.
 LOCAL The spending of the second half of 700 billion depends on a new decree of Congress.
 OPS KNOWN(-14.566) SPLIT(-14.464)
 STATS src(17), moses(18), local(17), views(7213), worst(-21.739000), best(-14.464000), string changed, MOSES

Corpus de test newstest2010

SOURCE sauf que je n'&apost; ai aperçu aucune benne de couleur rue kounicova .
MOSES Except that I have no overview collection lorry colour street kounicova.
LOCAL Except that I did no overview collection lorry colour street kounicova.
OPS UNKNOWN(-115.405) REPLACE(-115.281)
STATS src(13), moses(12), local(12), views(3602), worst(-120.698000), best(-115.281000), string changed, MOSES

SOURCE les graffiti anti @-@ gouvernementaux et les discours sont considérés comme des crimes graves .
MOSES The anti-government graffiti and speeches are regarded as serious crimes.
LOCAL The graffiti anti-government and the speeches are regarded as serious crimes.
OPS KNOWN(-14.162) MOVE(-14.139) SPLIT(-14.023)
STATS src(15), moses(13), local(14), views(9220), worst(-21.014000), best(-14.023000), string changed, MOSES

SOURCE l'&apost; abkhazie séparatiste a élu son président , la géorgie dénonce une " farce ";
MOSES Abkhazia separatist election of its President, Georgia denounces a 'farce'
LOCAL Abkhazia separatist has elected its President, Georgia denounces a 'farce'
OPS KNOWN(-16.336) SPLIT(-16.305)
STATS src(15), moses(13), local(13), views(4219), worst(-23.153000), best(-16.305000), string changed, LOCAL

SOURCE quant aux pistes cyclables , les services du fes ne peuvent généralement pas s'&apost; en occuper .
MOSES As for the cycle paths, the services of fes cannot generally not deal with it.
LOCAL Regarding the cycle paths, the services of fes cannot generally not deal with it.
OPS UNKNOWN(-116.447) REPLACE(-116.435)
STATS src(17), moses(17), local(16), views(6574), worst(-121.513000), best(-116.435000), string changed, MOSES

SOURCE à partir de 2020 , la question change , et il faudra augmenter la puissance .
MOSES From 2020, the issue is changing, and it will increase the power.
LOCAL From 2020, the question is changing, and it will increase the power.
OPS KNOWN(-12.213) REPLACE(-12.213)
STATS src(16), moses(15), local(15), views(6290), worst(-18.166000), best(-12.213000), string changed, EQUIV

SOURCE le real madrid et le f.c. barcelona ont été bien servis lors de cette répartition .
MOSES Real Madrid and Barcelona F.C. have been well served at this distribution.
LOCAL The Real Madrid and Barcelona F.C. were well served during this distribution.
OPS UNKNOWN(-215.167) SPLIT(-215.133) REPLACE(-215.108) REPLACE(-215.091)
STATS src(16), moses(13), local(13), views(10460), worst(-221.517000), best(-215.091000), string changed, EQUIV

Corpus de test newstest2011

SOURCE vous avez dans votre lit une planche rigide au lieu d'&apost; un sommier ?
MOSES You have done in your read a rigid instead of a sommier?
LOCAL You have done in your bed a rigid instead of a sommier?
OPS UNKNOWN(-113.506) REPLACE(-113.460)
STATS src(14), moses(13), local(13), views(4010), worst(-119.905000), best(-113.460000), string changed, LOCAL

SOURCE le tempur se rammollit à la chaleur et se durcit , au contraire , au froid .
MOSES The tempur is rammollit heat and is strengthening, on the contrary, the cold.
LOCAL The tempur is rammollit heat and hardening, on the contrary, the cold.
OPS UNKNOWN(-215.367) SPLIT(-215.262)
STATS src(17), moses(16), local(15), views(4580), worst(-220.832000), best(-215.262000), string changed, LOCAL

SOURCE la tchéquie comme la plupart de l'&apost; europe est ce jeudi matin recouverte par la neige .
MOSES The Czech Republic as most of Europe is this Thursday morning covered in snow.
LOCAL The Czech Republic as most of Europe is this Thursday morning covered by snow.

OPS KNOWN(-13.664) REPLACE(-13.657)
 STATS src(17), moses(15), local(15), views(5138), worst(-20.380000), best(-13.657000), string changed, EQUIV

SOURCE même au large de la côte atlantique , de nouveaux forages demeurent pour l’instant interdits .
 MOSES Even off the Atlantic coast, new drilling remain for the time being banned.
 LOCAL Even off the Atlantic coast, new drilling remain at the moment banned.
 OPS KNOWN(-15.309) REPLACE(-15.289)
 STATS src(17), moses(15), local(14), views(6964), worst(-22.087000), best(-15.289000), string changed, MOSES

SOURCE par endroits , il y est tombé 30 centimètres de neige fraîche que les vents forts et frais ont dispersée .
 MOSES In places, there fell 30 centimetres of snow fresh that the strong winds and costs are fragmented.
 LOCAL In places, there fell 30 centimetres of snow fresh that the strong winds and costs are dispersed.
 OPS KNOWN(-22.095) SPLIT(-22.022)
 STATS src(21), moses(19), local(19), views(6903), worst(-28.245000), best(-22.022000), string changed, LOCAL

SOURCE il faut que le kmt fasse une introspection de soi @-@ même .
 MOSES It is important that the kmt make a introspection of itself.
 LOCAL We need the kmt make a introspection of itself.
 OPS UNKNOWN(-111.927) REPLACE(-111.902)
 STATS src(13), moses(12), local(10), views(5778), worst(-119.498000), best(-111.902000), string changed, MOSES

SOURCE le ministère de la justice s’est abstenu de commentaire à ce sujet .
 MOSES The Ministry of Justice abstained comment about this.
 LOCAL The Ministry of Justice abstained comment on this matter.
 OPS KNOWN(-9.568) REPLACE(-9.557)
 STATS src(14), moses(9), local(10), views(6398), worst(-16.420000), best(-9.557000), string changed, LOCAL

SOURCE ils totalisent 449 783 344 \$, y compris le contrat à énergie gérance de 295 131 564 \$.
 MOSES They account for 449 783 344 \$, including the contract to energy stewardship ranks of 295 131 564 \$.
 LOCAL They total 449 783 344 \$, including the contract to energy stewardship ranks of 295 131 564 \$.
 OPS UNKNOWN(-226.266) REPLACE(-226.252)
 STATS src(20), moses(21), local(20), views(3936), worst(-232.208000), best(-226.252000), string changed, LOCAL

SOURCE les dernières prévisions de bruxelles ont été une douche froide pour le gouvernement .
 MOSES The latest forecasts Brussels were a cold shower for the government.
 LOCAL The latest forecasts of Brussels were a cold shower for the government.
 OPS KNOWN(-10.794) SPLIT(-10.762)
 STATS src(14), moses(12), local(13), views(4484), worst(-17.388000), best(-10.762000), string changed, LOCAL

Recherche locale gloutonne avec le modèle de collocations

Corpus de test newstest2009

SOURCE olmert rendrait en échange de la paix presque tous les territoires occupés
 MOSES Olmert would make in exchange for peace almost all the occupied territories
 LOCAL Olmert would almost all the occupied territories in exchange for peace
 OPS KNOWN(-11.390) MOVE(-11.160) REPLACE(-10.919)
 STATS src(12), moses(12), local(11), views(7317), worst(-19.056000), best(-10.919000), string changed, MOSES

SOURCE sylvana et les secrets de la boîte à bento
 MOSES Sylvana and the secrets of the boîte to bento
 LOCAL And the secrets of the boîte to bento sylvana
 OPS UNKNOWN(-308.116) MOVE(-308.069)

STATS src(9), moses(9), local(9), views(718), worst(-312.451000), best(-308.069000), string changed, MOSES
 SOURCE les ravisseurs soudanais avaient exigé une rançon de plusieurs millions d' euros .
 MOSES The kidnappers of Sudan had demanded a ransom of several million euros.
 LOCAL The kidnappers Sudanese had demanded a ransom of several million euros.
 OPS KNOWN(-11.600) REPLACE(-11.592)
 STATS src(13), moses(13), local(12), views(4480), worst(-19.310000), best(-11.592000), string changed, LOCAL

 SOURCE sa belle @-@ fille partit à plymouth .
 MOSES His daughter beautiful they sent out Plymouth.
 LOCAL Its daughter beautiful they sent out Plymouth.
 OPS KNOWN(-10.819) REPLACE(-10.807)
 STATS src(8), moses(8), local(8), views(1006), worst(-111.662000), best(-10.807000), string changed, EQUIV

 SOURCE l' épopée de meat loaf devra être honorée .
 MOSES The journey of meat loaf must be honoured.
 LOCAL The saga of meat loaf must be honoured.
 OPS UNKNOWN(-111.838) REPLACE(-111.798)
 STATS src(9), moses(9), local(9), views(1100), worst(-116.967000), best(-111.798000), string changed, EQUIV

 SOURCE " des fois dieu crée des personnes parfaites
 MOSES 'time God creates people perfect
 LOCAL 'time God creates perfect people
 OPS KNOWN(-10.580) MOVE(-10.528)
 STATS src(8), moses(6), local(6), views(1888), worst(-15.176000), best(-10.528000), string changed, LOCAL

 SOURCE cette semaine on a parlé en plus d' un changement de stratégie chez microsoft .
 MOSES This week we have spoken in addition to a change of strategy among Microsoft.
 LOCAL This week we spoke in addition to a change of strategy among Microsoft.
 OPS KNOWN(-11.671) SPLIT(-11.669)
 STATS src(15), moses(15), local(14), views(5947), worst(-19.163000), best(-11.669000), string changed, EQUIV

 SOURCE le silence de la nuit retombe sur la place d' octobre .
 MOSES The silence of the night falls on the place in October.
 LOCAL The silence of the night falls on the place of October.
 OPS KNOWN(-10.260) SPLIT(-10.246)
 STATS src(12), moses(12), local(12), views(5167), worst(-17.543000), best(-10.246000), string changed, LOCAL

 SOURCE on a même donné un nom au bébé hors de prix
 MOSES It has even given a name the baby overpriced
 LOCAL It has even given a name at the baby overpriced
 OPS KNOWN(-11.520) MERGE(-11.517)
 STATS src(11), moses(9), local(10), views(4097), worst(-18.011000), best(-11.517000), string changed, LOCAL

 SOURCE comme il l' a mentionné , l' orientation schüssel @-@ molterer a été , en tout cas , rejetée .
 MOSES As he said, the orientation Schüssel-Molter was, in any case, rejected.
 LOCAL As he said, the direction Schüssel-Molter was, in any case, rejected.
 OPS KNOWN(-15.299) REPLACE(-15.293)
 STATS src(20), moses(17), local(17), views(8528), worst(-21.249000), best(-15.293000), string changed, MOSES

 SOURCE depuis 1946 , le spd a atteint ses plus mauvais résultats avec 18,6 pour cent en bavière .
 MOSES Since 1946, the SPD has reached its worst results with 18,6 percent in Bavaria.
 LOCAL Since 1946, the SPD has reached its worst results with 18,6 per cent in Bavaria.
 OPS UNKNOWN(-114.313) REPLACE(-114.268)
 STATS src(18), moses(16), local(17), views(5220), worst(-120.719000), best(-114.268000), string changed, EQUIV

Corpus de test newstest2010

SOURCE il suffit d'& un bouton et vous passez de windows mobile à android en quelques secondes .
MOSES It is enough of a button and mentions mobile windows android in a few seconds.
LOCAL It is enough of a button and mobile windows mentions android in a few seconds.
OPS UNKNOWN(-115.951) MOVE(-115.902)
STATS src(17), moses(16), local(16), views(4330), worst(-122.545000), best(-115.902000), string changed, MOSES

SOURCE c'& est un petit geste que peuvent faire les civils pour soutenir nos troupes , a t-elle dit .
MOSES It is a small gesture that can do civilians to support our troops, as she said.
LOCAL It is a small gesture that can make civilians to support our troops, as she said.
OPS KNOWN(-15.200) REPLACE(-15.191)
STATS src(20), moses(18), local(18), views(6870), worst(-22.352000), best(-15.191000), string changed, EQUIV

SOURCE affaire liliane bettencourt : françois t marie banier sera jugé pour abus de faiblesse
MOSES Case liliane bettencourt: François-Marie banier will be judged for abuse of weakness
LOCAL Case liliane bettencourt François-Marie banier will be judged for abuse of weakness:
OPS UNKNOWN(-314.560) MOVE(-314.216)
STATS src(14), moses(15), local(15), views(2852), worst(-320.678000), best(-314.216000), string changed, MOSES

SOURCE rer a : perturbations ce week t end , syndicats mécontents des propositions de la ratp
MOSES RER has: disruption this weekend, unions dissatisfied with the proposals of the RATP
LOCAL RER disruption has: this weekend, trade unions dissatisfied with the proposals of the RATP
OPS UNKNOWN(-117.276) MOVE(-117.235) SPLIT(-117.215) MERGE(-117.136)
STATS src(16), moses(15), local(16), views(8691), worst(-125.399000), best(-117.136000), string changed, EQUIV

SOURCE les trois espèces ne peuvent descendre d'& un ancêtre commun qui aurait migré , avancent les chercheurs
MOSES The three species cannot take a common forbear which would have migrated, give researchers.
LOCAL The three species cannot take a common forbear which would have migrated, are progressing researchers.
OPS KNOWN(-19.598) SPLIT(-19.576)
STATS src(18), moses(16), local(17), views(5285), worst(-26.015000), best(-19.576000), string changed, EQUIV

SOURCE championnat du monde poids lourds
MOSES World Cup HGV
LOCAL World Cup heavy goods vehicles
OPS KNOWN(-5.722) REPLACE(-5.717)
STATS src(5), moses(3), local(5), views(1814), worst(-12.117000), best(-5.717000), string changed, EQUIV

SOURCE jusqu'& à présent , il n 'a pas obtenu de réponse
MOSES So far, it has not been answered
LOCAL Until now, it has not been answered
OPS KNOWN(-7.996) SPLIT(-7.959)
STATS src(12), moses(8), local(8), views(6991), worst(-15.053000), best(-7.959000), string changed, MOSES

SOURCE gm admet être en contact avec de possibles nouveaux acheteurs .
MOSES GM admits be in contact with possible new buyers.
LOCAL GM admits come into contact with possible new buyers.
OPS KNOWN(-11.004) MERGE(-10.944)
STATS src(11), moses(10), local(10), views(2885), worst(-16.463000), best(-10.944000), string changed, EQUIV

Corpus de test newstest2011

SOURCE celle qui devient 'sorcière ' est isolée pour ne pas menacer le reste du village .
MOSES That which becomes' apprentice 'is isolated so as not to threaten the rest of the village.
LOCAL That becomes' apprentice 'is isolated so as not to threaten the rest of the village.
OPS KNOWN(-16.700) REPLACE(-16.679)

STATS src(17), moses(19), local(18), views(6142), worst(-22.648000), best(-16.679000), string changed, MOSES
 SOURCE la tendance est également au matelas bio .
 MOSES The trend is also the mattress bio.
 LOCAL The trend is also the mattress organic.
 OPS KNOWN(-8.417) REPLACE(-8.370)
 STATS src(8), moses(8), local(8), views(2568), worst(-13.768000), best(-8.370000), string changed, LOCAL

 SOURCE l&apostrophe; euphorie de la victoire a été rapidement remplacée par les effets de la crise .
 MOSES The euphoria of victory was rapidly replaced by the effects of the crisis.
 LOCAL The euphoria of the victory was rapidly replaced by the effects of the crisis.
 OPS UNKNOWN(-109.166) SPLIT(-109.158)
 STATS src(16), moses(14), local(15), views(6666), worst(-117.453000), best(-109.158000), string changed, LOCAL

 SOURCE est déterminante non seulement la bonne saveur , mais également la souplesse soyeuse et la bonne consistance
 MOSES Is crucial not only the right flavour, but also the flexibility soyeuse and good substance.
 LOCAL Is crucial not only the right, but also the flexibility soyeuse flavour and good substance.
 OPS UNKNOWN(-115.552) MOVE(-115.211)
 STATS src(18), moses(17), local(17), views(7602), worst(-122.091000), best(-115.211000), string changed, MOSES

 SOURCE président de 2000 à 2008 , il était un provocateur en faveur d&apostrophe; une indépendance explicite .
 MOSES President 2000-2008, it was a provocateur for an explicit independence.
 LOCAL President 2000-2008, it was a provocateur in favour of an explicit independence.
 OPS KNOWN(-13.804) REPLACE(-13.726)
 STATS src(17), moses(14), local(16), views(6360), worst(-19.017000), best(-13.726000), string changed, LOCAL

 SOURCE " en conséquence , les organismes ont du mal à distinguer ces éléments séparément . "
 MOSES 'accordingly, the bodies have difficulty distinguish these elements separately.'
 LOCAL 'accordingly, the bodies have difficulty to distinguish these elements separately.'
 OPS KNOWN(-14.768) SPLIT(-14.720) MERGE(-14.663) SPLIT(-14.662)
 STATS src(16), moses(13), local(14), views(12976), worst(-20.643000), best(-14.662000), string changed, LOCAL

 SOURCE il peut s&apostrophe; agir de patins à roulettes ou d&apostrophe; un animal de compagnie jouet puffle .
 MOSES This may be roller castors or a pet animal puffle toy.
 LOCAL It can be roller castors or a pet animal puffle toy.
 OPS UNKNOWN(-115.976) SPLIT(-115.970)
 STATS src(17), moses(12), local(12), views(4775), worst(-216.658000), best(-115.970000), string changed, EQUIV

 SOURCE ils envoyaient toutes les minutes des rapports de témoignages horribles via twitter .
 MOSES They sent every minute reports of testimonies horrific via Twitter.
 LOCAL They sent every minute of reports of testimonies horrific via Twitter.
 OPS KNOWN(-15.121) SPLIT(-15.109)
 STATS src(13), moses(11), local(12), views(2946), worst(-20.250000), best(-15.109000), string changed, MOSES

 SOURCE la cellule catalane a facilité des milliers de passeports à al qaida
 MOSES The Catalan cell facilitated thousands of passports to Al Qaida
 LOCAL The Catalan cell has facilitated thousands of passports to al-Qa'ida
 OPS KNOWN(-12.470) MERGE(-12.411) REPLACE(-12.378)
 STATS src(12), moses(10), local(13), views(4734), worst(-18.724000), best(-12.378000), string changed, LOCAL

 SOURCE le rapport détaille les gains que ferait chaque pays grâce à ce programme international .
 MOSES The report spells out the gains that would each country through this programme international.
 LOCAL The report spells out the gains that would each country thanks to this programme international.
 OPS KNOWN(-13.598) REPLACE(-13.580)
 STATS src(15), moses(15), local(16), views(5022), worst(-19.609000), best(-13.580000), string changed, LOCAL

 SOURCE les sociétés auront un an pour se préparer à l&apostrophe; application de la loi , a @-@ t-il précisé .

MOSES Companies will have a year to prepare for the application of the law, he stated.
 LOCAL Companies will have a year to prepare for the application of the law, he said.
 OPS KNOWN(-13.847) SPLIT(-13.843)
 STATS src(20), moses(17), local(17), views(9079), worst(-20.444000), best(-13.843000), string changed, EQUIV

SOURCE " on les couvre de boue , on essaie de les compromettre " , a lancé m. poutine .
 MOSES 'They covers mud, it attempts to the compromise', launched Mr Putin.
 LOCAL 'It covers mud, it attempts to the compromise', launched Mr Putin.
 OPS KNOWN(-18.053) REPLACE(-18.037)
 STATS src(19), moses(16), local(16), views(6682), worst(-24.731000), best(-18.037000), string changed, LOCAL

SOURCE autrefois , la presse essayait de comprendre ce qui se tramait dans le secret des ambassades .
 MOSES In the past, the press was trying to understand what went on in secret embassies.
 LOCAL In the past, the press was trying to understand what went on in secrecy embassies.
 OPS KNOWN(-14.788) REPLACE(-14.772)
 STATS src(17), moses(17), local(17), views(5184), worst(-115.749000), best(-14.772000), string changed, EQUIV

SOURCE certains , comme la france , considèrent qu ' 1 € prêté équivaut à 1 € donné .
 MOSES Some, like France, consider that 1 princely lent amounts to €1.
 LOCAL Some, like France, consider that 1 princely lent is tantamount to €1.
 OPS KNOWN(-20.462) REPLACE(-20.423)
 STATS src(18), moses(15), local(16), views(6160), worst(-26.724000), best(-20.423000), string changed, LOCAL

Bibliographie

- [1] Ulrich Germann. Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8. Association for Computational Linguistics, 2003.
- [2] Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 228–235. Association for Computational Linguistics, 2001.
- [3] Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. Docent : A document-level decoder for phrase-based statistical machine translation. In *ACL 2013 (51st Annual Meeting of the Association for Computational Linguistics); 4-9 August 2013; Sofia, Bulgaria*, pages 193–198. Association for Computational Linguistics, 2013.
- [4] Philipp Koehn. Pharaoh : a beam search decoder for phrase-based statistical machine translation models. In *Machine translation : From real users to research*, pages 115–124. Springer, 2004.
- [5] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.
- [6] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses : Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.
- [7] Philippe Langlais, Alexandre Patry, and Fabrizio Gotti. A greedy decoder for phrase-based statistical machine translation. *Proc. of TMI*, 2007.

- [8] Lewis, M. Paul, Gary F. Simons, and Charles D. (eds.) Fennig. Ethnologue : Languages of the world, seventeenth edition. <http://www.ethnologue.com>, 2013.
- [9] Benjamin Marie. Amélioration a posteriori d'une traduction automatique par recherche locale. <https://perso.limsi.fr/marie/pdf/memoire.pdf>, 2012.
- [10] Pierre Paul Monty. Traduction statistique par recherche locale. 2010.
- [11] Makoto Nagao. A framework of a mechanical translation between Japanese and English by analogy principle, 1984.
- [12] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.
- [13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.