Mémoire de recherche
Master Ingéniere linguistique

Amélioration *a posteriori* d'une traduction automatique par recherche locale

Benjamin Marie

15 novembre 2012

Directeur de mémoire :
Cyril Grouin
Encadrant de stage :
Aurélien Max

LIMSI - CNRS
Équipe Traitement du Langage Parlé

Résumé

Les traductions automatiques produites par des systèmes de traduction statistiques état de l'art sont encore aujourd'hui très imparfaites. La recherche locale est une technique qui permet l'amélioration, *a posteriori*, d'une traduction déjà produite par un système par maximisation de la fonction de score du système lors d'une recherche par transformations locales. Cependant, l'amélioration des systèmes de traduction a rendu la recherche locale moins efficace au point que celle-ci dégrade plus souvent qu'elle n'améliore les traductions initiales. Notre travail met en évidence qu'en dépit du fait qu'elle produise majoritairement des dégradations, la recherche locale permet toujours un fort potentiel d'amélioration des traductions. Nous avons pour cela étudié différentes pistes. De nouveaux modèles ont été introduits dans la fonction objectif utilisée par la recherche locale afin de mieux évaluer les hypothèses de traduction en tant qu'énoncés. Nous avons également étudié la mise en place d'une mesure de confiance, notamment au travers d'expériences oracle, qui permet à la recherche locale de ne pas modifier les zones de la traduction qui sont correctes d'après cette mesure. De plus, afin d'élargir l'espace de recherche, nous avons introduit un nouvel opérateur, `paraphrase`, qui permet d'atteindre de nouvelles améliorations en modifiant la phrase à traduire. Nos travaux mettent en avant des perspectives prometteuses, tel que le développement d'une mesure de confiance automatique capable de mieux guider la recherche locale ou encore, donner à la recherche locale la possibilité de parcourir plus en profondeur son espace de recherche pour trouver de nouvelles améliorations de traduction.

Mots-clés

traduction automatique statistique, modèle à base de segments, recherche locale, recherche locale oracle, mesure de confiance, paraphrase, grammaticalité des traductions automatiques

Remerciements

Je tiens tout d'abord à remercier mon directeur de mémoire, Cyril Grouin, pour ses suggestions dans la rédaction du mémoire ainsi que Aurélien Max qui m'a encadré tout au long du stage en me donnant de nombreux et précieux conseils pour réaliser les recherches présentées dans ce mémoire.

Je voudrais aussi remercier le LIMSI-CNRS qui m'a accueilli au sein de l'équipe TLP et m'a donné l'environnement idéal pour mener à bien mes recherches. Je souhaiterais plus particulièrement remercier Hai Son Le pour ses scripts et ses conseils, Nadi Tomeh pour son aide dans la compréhension du fonctionnement de `moses` et Li Gong qui m'a fourni tout au long du stage les données et corpus constituant la base de mes recherches.

Je remercie également l'INaLCO pour ses enseignements et pour m'avoir fait rencontrer de nombreuses personnes travaillant dans le domaine du Traitement Automatique des Langues.

Table des matières

Table des figures

Liste des tableaux

1 Introduction

Ce mémoire a été rédigé dans le cadre du Master d'ingénierie linguistique préparé à l'INaLCO. Il présente les recherches réalisées au cours d'un stage effectué au sein du laboratoire LIMSI-CNRS durant la période allant de début mai à fin octobre 2012.
Ces recherches s'inscrivent dans le domaine de la traduction automatique statistique, domaine qui a connu d'importantes avancées au cours des deux dernières décennies permettant aux systèmes de traduction automatique de produire des traductions de plus en plus acceptables. Cependant de nombreux progrès restent à faire, notamment sur le plan de la grammaticalité des traductions produites par ce type de système.

Les recherches décrits dans ce mémoire proposent d'enrichir une approche existante permettant l'analyse et la correction *a posteriori* d'erreurs produites par les systèmes de traduction automatique statistique : la recherche locale. La recherche locale rend possible une amélioration des traductions automatiques en s'appuyant sur les informations qu'une hypothèse de traduction complète, déjà produite par un autre système, peut fournir.

Le présent mémoire décrit tout d'abord un état de l'art général sur la traduction automatique statistique (section 2), de ses débuts jusqu'aux derniers progrès effectués. Nous détaillons en particulier le fonctionnement de la traduction automatique statistique dite *à base de segments* et présentons également les principales métriques permettant l'évaluation automatique des traductions produites par de tels systèmes. Nous mettons ensuite en évidence les limites actuelles des systèmes de traduction automatique ainsi que la mesure dans laquelle il est possible d'améliorer les traductions produites (section 3). Puis nous introduisons la recherche locale comme technique pour rechercher et appliquer des améliorations *a posteriori* sur la meilleure hypothèse d'un système de traduction automatique. Nous nous sommes attaché à reproduire les résultats de l'état de l'art en recherche locale pour souligner ses limites et mettre en évidence son potentiel grâce à des expériences oracle (section 4). En outre nous présentons différentes solutions pour contourner ces limites au moyen d'un meilleur guidage de la recherche locale afin de diminuer le risque que celle-ci fasse des erreurs (section 5). Nous y développons 3 solutions : l'ajout de nouveaux modèles à la fonction objectif calculée par les sytèmes de traduction automatique statistique ; un meilleur guidage de la recherche locale pour corriger en priorité les zones de la traduction dites de *faible confiance* ; un élargissement de l'espace de recherche de la recherche locale *via* l'introduction d'un nouvel opérateur transformant la phrase traduite. Dans une dernière partie (section 6) nous discutons des perspectives et améliorations envisgeables à notre travail.

2 La traduction automatique

2.1 Les débuts de la traduction automatique

La recherche en Traduction Automatique (TA) remonte au début des années 50[1]. Ses débuts sont généralement attribués au mathématicien Warren Weaver qui décrivit une langue étrangère comme un code secret à décoder. Toutefois, si ce domaine existe depuis plusieurs décennies, les progrès des systèmes de TA ont été très lents.
Les premiers systèmes de traduction furent des systèmes à base de règles (RBMT, pour *Rule-Based Machine Translation*). Un système RBMT utilise un ensemble de règles pour analyser morphologiquement, syntaxiquement et sémantiquement une phrase à traduire en langue *source* afin de produire la phrase traduite en langue *cible*. Les ensembles de règles utilisés par un système RBMT ont plusieurs défauts : ils sont difficiles et coûteux à développer et à faire évoluer, car ils sont élaborés par des humains, et ils sont généralement très liés à une paire de langues particulière, puisqu'ils sont fondés sur des considérations linguistiques.
Prenons l'exemple du système METEO (voir une description dans [Hutchins and Somers, 1992]) développé dans les années 70. Ce système avait été conçu pour traduire automatiquement des bulletins météorologiques en anglais vers le français. Il ne disposait que d'un vocabulaire très restreint et ne pouvait par conséquent traduire que des textes de ce domaine particulier, pour cette paire de langue particulière.
C'est au début des années 90 qu'un nouveau type de système commence à voir le jour. Face au développement rapide des corpus électroniques ainsi que d'approches statistiques du Traitement Automatique des Langues (TAL), il devint possible d'utiliser des méthodes probabilistes en Traduction Automatique. Des chercheurs de la société IBM ont posé les fondements de ces nouveaux systèmes de TA dits statistiques (SMT, *Statistical Machine Translation*). Cette approche devaient notamment offrir une plus grande robustesse face à la diversité des textes à traduire, tout en tirant parti des corpus bilingues déjà produits en grandes quantités pour plusieurs paires de langues.

2.2 Les fondements de la Traduction Automatique Statistique

Formellement, la Traduction Automatique Statistique consiste à résoudre le problème probabiliste suivant :

$$\mathbf{e}_{best} = \arg\max_{e} p(e, f) \tag{1}$$

On cherche la probabilité maximale telle que pour une phrase à traduire f on a la phrase e[2] dans la langue vers laquelle on traduit. Ce problème peut être décomposé en appliquant la règle de Bayes :

$$\mathbf{e}_{best} = \arg\max_{e} p(f|e)p(e) \tag{2}$$

Cette formulation fait apparaître $p(f|e)$, soit la probabilité d'apparier la phrase f avec une traduction générée automatiquement e. Cette probabilité peut être fournie par un *modèle de traduction* appris sur des *corpus bilingues parallèles*. Dans de tels corpus, les textes sont alignés phrase à phrase avec une traduction produite par un humain. Ceci est illustré dans la Table 1 pour la paire anglais-français[3].

L'équation 2 fait également apparaître $p(e)$, la probabilité que la phrase e produite existe dans la langue vers laquelle on traduit. Si la phrase e est mal formée, localement grammaticalement incorrecte, ou plus généralement peu probable étant donné un grand corpus représentatif de la langue en question, alors la valeur de $p(e)$ sera faible. Cette probabilité est donnée par un *modèle de langue* (voir section 2.4). Le fonctionnement général d'un système de Traduction Automatique Statistique est résumé dans la Figure 1.

[1] Voir par exemple [Hutchins and Somers, 1992, Koehn, 2010, Allauzen and Yvon, 2011] pour des descriptions générales.
[2] La notation f est utilisée pour signifier *foreign* (i.e. la langue depuis laquelle on traduit), et la notation e *English* (i.e. la langue vers laquelle on traduit).
[3] Il faut noter que, dans la majorité des cas, la langue d'origine ayant servi à obtenir le texte source par traduction n'est pas connue, alors qu'il a été montré que la direction d'une traduction a une grande influence sur la performance des systèmes de TA statistique [Kurokawa et al., 2009]. Il est même possible qu'aucune de ces deux langues ne soit la vraie langue d'origine.

anglais	**français**
Mr Frazetta Snr , aged 81 , is famed for his depiction of characters such as Conan the Barbarian and Tarzan .	M. Frazetta Senior , âgé de 81 ans , est célèbre pour ses dessins de personnages tels que Conan le Barbare et Tarzan .
the artist was in Florida at the time of the incident , the Associated Press news agency reported .	au moment de l' incident , l' artiste était en Floride , a rapporté l' agence de presse Associated Press (AP) .
AP quoted an unnamed police official as saying the younger Mr Frazetta may have been motivated by a family feud .	AP a cité un responsable de la police selon lequel le jeune M. Frazetta peut avoir été motivé par une querelle familiale .

TAB. 1 – Exemple d'alignement phrase à phrase

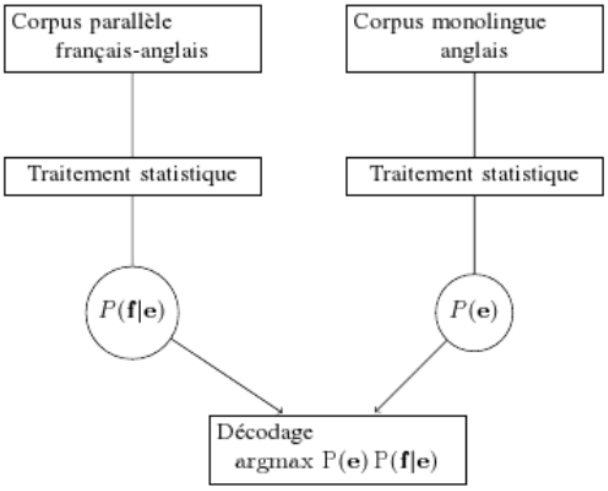

FIG. 1 – Fonctionnement d'un système de traduction probabiliste (tiré de Allauzen and Yvon [2011]).

Le problème de la construction d'une traduction ne peut cependant pas se résumer à chercher la traduction d'une phrase présente dans un corpus bilingue. Cela reviendrait en effet à considérer que toutes les phrases qu'il est possible de former ont déjà été traduites par des humains et que nous avons accès à cette traduction. Pour tenter de résoudre ce problème, les premiers systèmes de TA statistiques découpaient la phrase en mots : on les décrit donc comme systèmes *à base de mots* (*word-based*).
Avant de traduire à proprement parler, un tel système doit donc décomposer la phrase en mots, ou plus généralement en *tokens*, par un traitement de *tokénisation* (ou *segmentation en tokens*)[4]. Ce découpage permet de reformuler la probabilité $p(f|e)$ de l'équation 2 de la façon suivante :

$$p(f|e) = \prod_{i=1}^{I} p(f_i|e_i) \tag{3}$$

Le modèle de traduction estime ici la probabilité d'apparier chaque mot de la phrase source avec chacune de ses traductions en langue cible. Cette modélisation se confronte notamment au problème de la polysémie lexicale et de l'homographie. Dans une traduction du français vers l'anglais, si on a par exemple le mot "avocat", le système de traduction devra décider s'il le traduit par le sens "lawyer" ou "avocado". En se basant sur les corpus bilingues alignés le modèle de traduction pourrait donner les probabilités suivantes : $p(\text{lawyer}|\text{avocat}) = 0.8$, et $p(\text{avocado}|\text{avocat}) = 0.2$. Concrètement, la traduction

[4]Ce traitement est relativement peu ambigü pour de nombreuses langues qui séparent explicitement les mots, telle que l'anglais ou le français. Il n'en va pas de même pour des langues très agglutinantes ou qui n'ont pas de tels séparateurs, comme l'ont montré les travaux réalisés au LIMSI sur la tokénisation [Déchelotte et al., 2008].

"avocado" ne pourrait être effectivement choisie que si, par exemple, le modèle de langue $p(e)$ avait une forte préférence pour cette traduction (voir la Table 2).

f	*cet avocat est mon défenseur*
e1	*this lawyer is my defender*
p(f|e1)	p(cet|this)p(avocat|lawyer)p(est|is)p(mon|my)p(défenseur|defender)
e2	*this avocado is my defender*
p(f|e2)	p(cet|this)p(avocat|avocado)p(est|is)p(mon|my)p(défenseur|defender)

TAB. 2 – Problème du choix d'une traduction pour un mot polysémique ayant deux traductions distinctes.

De plus, il existe des différences dans l'ordre des mots entre langues qui doivent être prises en compte. Prenons l'exemple du japonais où le verbe se situe le plus souvent en fin de phrase. Une traduction mot à mot vers l'anglais ne tenant pas compte de cela commettrait l'erreur de mettre le verbe en fin de phrase. Des différences plus locales existent également, par exemple concernant la position des adjectifs relativement au nom modifié en anglais et en français. La solution proposée dans l'approche statistique standard est d'ajouter à l'équation 2 un ou plusieurs modèles dits de *réordonnancement* (voir la section 2.4), qui estiment diverses probabilités de déplacer les mots relativement à une traduction *monotone* de la phrase source.

Un autre problème important concerne la non-compositionnalité des traductions : il est souvent impossible de traduire un mot par un mot exactement. Par exemple, le mot composé "pomme de terre" se traduit en anglais par "potatoe", et plus généralement il est fréquemment nécessaire de traduire un groupe de tokens par un groupe de tokens. La traduction statistique dite *à base de segments* (*phrase-based*[5]) [Koehn et al., 2003], qui tente d'apporter des solutions à ces problèmes, s'est progressivement développée depuis le début des années 2000.

2.3 La Traduction Automatique Statistique à base de segments

Contrairement à la traduction à base de mots, la traduction à base de segments [Koehn et al., 2003] utilise comme unités de traduction des groupes de tokens, dont la taille maximale est bornée (une valeur typique est de 6 tokens). Un segment peut donc capturer des éléments de contexte local, permettant de diminuer les risques d'ambiguïtés lexicales.

À partir d'un corpus parallèle aligné en phrases et au niveau des tokens[6], il est possible d'extraire des appariemments entre *segments source* (segments de la phrase à traduire) et un ou plusieurs *segments cible* correspondant (segments de la traduction). Pour chaque appariemment (on parle de *bi-segment* (*biphrase*)), un ensemble de scores peuvent être calculés, par exemple :

- la pondération lexicale directe : $lex(e|f)$
- la probabilité de traduction directe : $p(e|f)$
- la probabilité de traduction inverse : $p(f|e)$
- la pondération lexicale inverse : $lex(f|e)$

La probabilité de traduction évalue la probabilité d'avoir le bi-segment (e, f) dans un corpus parallèle. Dans sa forme la plus simple elle estime donc sur le corpus parallèle la fréquence F relative de cette association rapportée à toutes les occurrences de e ou de f :

$$p(e|f) = \frac{F(e,f)}{\Sigma_{e^k} F(e^k,f)'} \tag{4}$$

Cette estimation est toutefois très optimiste pour déterminer à elle seule la probabilité d'avoir un bi-segment (e, f), notamment en ce qui concerne les segments rares ou lorsqu'on utilise des corpus parallèles

[5] Quelques précisions terminologiques s'imposent : *phrase-based* doit ici se traduire par "segment", et non par "syntagme", car les unités considérées ne sont pas supposées répondre à la définition d'un constituent linguistique.

[6] Voir les références [Koehn, 2010, Allauzen and Yvon, 2011] pour l'alignement statistique de tokens, que nous n'aborderons pas ici en détails.

de petite taille. Dans ces situations, on peut en effet avoir $p(f|e) = p(e|f) = 1$ ce qui correspond fréquemment à une mauvaise modélisation. Pour améliorer la qualité de ces estimations, le travail de Koehn et al. [2003] ajoute un score de *pondération lexicale* (*lexical weighting*) pour évaluer la qualité des alignements des mots qui composent le bi-segment :

$$lex(e|f, A) = \prod_{i=i_d}^{i_f} \frac{1}{F(j|A(i,j)=1)} \sum_{j|A(i,j)=1} \frac{F(e_i, f_i)}{F(f_i)'} \tag{5}$$

$F(e_i, f_j)$ et $F(f_j)$ correspondent respectivement aux fréquences d'alignement du mot e_i avec le mot f_i et la fréquence du mot f_j. i_d et i_f sont les indices dans la phrase du début (d) et de la fin (f) du segment. Enfin, A prend la valeur maximale de tous les alignements trouvés entre e et f. La Table 3 illustre des appariemments possibles entre segments et les différents scores qui leur sont associés.

Segments source	**Traductions**	$p(e\|f)$	$lex(e\|f)$	$p(f\|e)$	$lex(f\|e)$
le salaire moyen d' un	the average salary of a	0.2	0.0012	0.25	3.1405E-4
	the average wage of a ,	0.2	9.3451E-4	1.0	2.4754E-4
	an average	0.2	2.1122E-4	1.1457E-4	3.9517E-9
	average salary for an	0.2	9.6648E-5	1.0	2.3409E-5
le deuxième	The second a	0.0087	0.0016	0.0021	0.0464
	the second	0.4649	0.3328	0.0011	0.0863
	the second one	0.0087	6.7489E-4	0.0037	0.0863
	the third	0.0087	8.8716E-4	4.9935E-5	3.0798E-4
rétrogradé	demoted in rank	0.0204	2.4835E-4	1.0	0.098
	reduced	0.0204	0.0196	2.6524E-5	2.2014E-5
	stipulating that	0.0204	2.7048E-4	0.0030	0.0014
	demoted	0.204	0.2549	0.1785	0.1969
	reduction	0.0204	0.0196	2.0738E-5	1.9228E-5
temps .	weather .	0.0098	0.0072	0.0011	0.0936
	time .	0.3333	0.4276	8.5928E-4	0.2564

TAB. 3 – Exemple de bi-segments

La Figure 2 décrit l'ensemble des traitements permettant d'extraire un bi-segment : les phrases du corpus parallèle bilingue sont tout d'abord alignées au niveau des tokens, puis la matrice d'alignement obtenue est utilisée pour guider l'extraction heuristique de bi-segments. Des comptes au niveau du corpus complet permettent ensuite d'estimer différents modèles que nous décrivons ci-dessous. Des recherches actives sont menées pour améliorer cette étape, voir notamment [Tomeh, 2012].

2.4 Des modèles et une fonction de score à maximiser

Comme nous l'avons vu, un système de TA statistique cherche à maximiser un score tel que celui de l'équation 2. Les systèmes actuels décomposent typiquement ce problème en combinant plusieurs modèles, ce qui donne la fonction de score suivante :

$$score(f, e) = \lambda_{lm} \log p_{lm}(e) + \sum_i \lambda_{tm}^{(i)} \log p_{tm}^{(i)}(f|e) - \lambda_d p_d(f, e) - \lambda_w p_w \tag{6}$$

Cette fonction met en jeu le score d'un modèle de langue $p_{lm}(e)$, de modèles de traduction $p_{tm}^{(i)}(f|e)$, de modèles de réordonnancement $p_d(f, e)$, ainsi qu'une pénalité lexicale p_w. Un poids λ est associé à chaque modèle afin de régler leur contribution individuelle au calcul du score.
Le modèle de langue, typiquement de type n-gramme, tente d'évaluer la "qualité" de la phrase produite dans la langue cible. Il calcule la probabilité pour chaque mot de la phrase dans le contexte des $n-1$ mots qui le précèdent. Ceci est illustré dans l'exemple suivant de modèle trigramme ($n = 3$)[7] :

[7] `<s>` représentant le début de la phrase, et `</s>` la fin de la phrase.

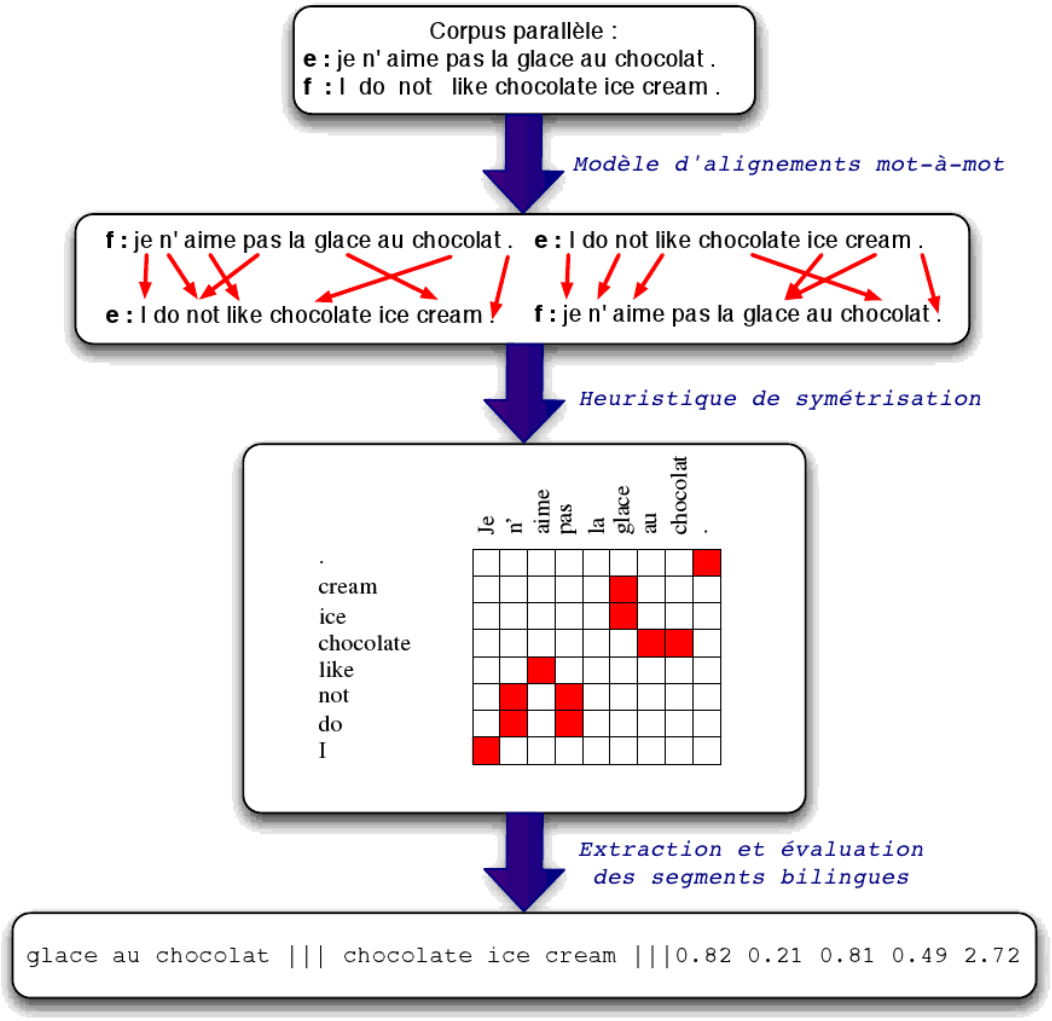

FIG. 2 – Traduction à base de segments (tiré de Allauzen and Yvon [2011])

$$\begin{aligned}\log P(I, saw, the, red, house) &= \log P(I| < s >, < s >) + \log P(saw| < s >, I) \\ &+ \log P(the|I, saw) + \log P(red|saw, the) \\ &+ \log P(house|the, red) + \log P(< /s > |red, house) \end{aligned} \quad (7)$$

Plus la phrase comptera de mots, et plus il est possible que la valeur du modèle de langue devienne faible[8]. Pour contrebalancer cet effet, la fonction de score utilise également une *pénalité lexicale* (ou bonus suivant le signe des poids associés) p_w, qui compte simplement le nombre de mots présents dans la phrase, ce qui permet d'apprendre un rapport de mots attendus entre traductions des langues source et cible.

Le modèle de traduction évalue quant à lui la qualité d'un appariemment entre une phrase source f et une phrase cible e. Celui-ci est calculé pour chaque segment en consultant les valeurs correspondantes dans une *table de traduction* construite lors de l'apprentissage du système. Le système que nous avons utilisé dans nos travaux (système Open Source `moses` [Koehn et al., 2007]) met en jeu les scores $lex(f|e)$, $p(e|f)$ et $lex(e|f)$ dont le calcul a été présenté dans la section 2.3. Nous n'avons pas utilisé $p(f|e)$, son absence est liée à l'estimation à la volée des tables de traduction [Gong et al., 2012]. Une pénalité de segmentation, qui donne une valeur constante ($\exp 1$) pour chaque bi-segment, permet de prendre en compte le nombre de segments source utilisés pour construire une traduction.

[8] La combinaison de l'équation 6 utilisant les logarithmes des scores, les valeurs données obtenues sont négatives.

Intervient également un modèle de distorsion p_d visant à autoriser des réordonnancements dans la traduction produite relativement à l'ordre de la phrase source. Dans sa forme la plus simple, il est calculé de la façon suivante :

$$p_d = d(a_i - b_{i-1} - 1) \tag{8}$$

où a_i représente la position du début du segment à traduire dans le i^{eme} segment de la phrase traduite, et b_{i-1} la position de la fin du segment à traduire dans le $(i-1)^{eme}$ segment de la phrase traduite.

Les poids λ de ces différents modèles sont optimisés sur un corpus de développement (distinct des corpus d'apprentissage et d'évaluation du système de traduction), par exemple par la technique de minimisation du taux d'erreur (*Minimum Error Rate Training* (MERT)) [Och, 2003]. Cette méthode cherche les paramètres λ qui permettent de traduire au mieux un corpus de développement D selon une des métriques d'évaluation automatique (voir section 2.6). Formellement on a donc :

$$\lambda^* = \arg\max_{\lambda} eval(\lambda, D) \tag{9}$$

Cette méthode est cependant imparfaite et peu robuste aux changements de conditions expérimentales. Ainsi, même après un changement mineur dans le système, il est conseillé de réexécuter la méthode MERT pour recalibrer en conséquence les poids λ.

2.5 Parcours de l'espace de recherche

Trouver l'hypothèse de traduction de meilleur score tel que défini par l'équation 6 est un problème NP-difficile [Knight, 1999], qui nécessite de chercher le meilleur compromis entre tous les modèles présents dans la fonction de score. Les tailles de la table de traduction, du modèle de langue et de l'espace des réordonnancements possibles étant particulièrement grandes, seul un algorithme de parcours heuristique de l'espace de recherche pourra produire une traduction en un temps acceptable durant la phase dite de *décodage*. La solution employée consiste à parcourir l'espace des préfixes en concaténant et réordonnant les segments au fur et à mesure que les hypothèses de traduction sont construites.

Dans le but de réduire le temps de calcul, une technique de recherche par faisceau (*beam search*) permet de ne conserver en mémoire que les k meilleures solutions trouvées jusque-là, focalisant ainsi la recherche sur un nombre limité d'hypothèses. Lorsque la phrase en entrée a été entièrement traduite, on peut extraire la meilleure hypothèse de l'espace de recherche effectivement parcouru selon la fonction de scores utilisée, voire les n meilleures (on parle alors informellement de *liste de n-bests*).

Parmi les défauts de cette approche, on note que ce type de recherche par faisceau supprime les hypothèses les moins prometteuses pour accélérer les calculs, alors qu'il se peut que celles-ci mènent à des hypothèses globalement meilleures, et que donc la meilleure hypothèse atteignable soit supprimée au cours de la recherche. En outre, le parcours de l'espace des préfixes produit des hypothèses partielles : il n'est donc pas possible de calculer des scores portant sur une hypothèse complète au cours de la recherche. Cela est typiquement fait lors d'une étape de *reclassement* (*reranking*) sur une liste de n-meilleures hypothèses (voir par exemple [Och et al., 2004]), ce qui limite donc les hypothèses considérées aux meilleures produites par la recherche initiale.

2.6 Des métriques automatiques pour l'évaluation de la performance des systèmes

La question de l'évaluation des systèmes de traduction s'avère être un problème relativement complexe [Koehn, 2010], qui génère régulièrement de nouvelles propositions. De nombreuses métriques d'évaluation automatiques de la performance des systèmes de traduction ont été développées. Celles-ci dispensent notamment du recours à des évaluations humaines, qui sont coûteuses, difficiles à mettre en place et non reproductibles. Les métriques BLEU (*BiLingual Evaluation Understudy* [Papineni et al., 2002] et TER (*Translation Error Rate*) [Snover et al., 2006] sont parmi les plus utilisées. BLEU mesure la ressemblance (à l'aide de précisions n-grammes et d'une *pénalité de concision*) entre la traduction

produite par un système et une ou plusieurs traductions humaines dites *de référence*. BLEU se calcule par l'équation 10. La pénalité de concision est donnée par l'équation 11, c et r étant respectivement la longueur (en nombre de tokens) de la traduction produite et de la traduction de référence, et p_n la précision n-gramme à laquelle est associée un poids positif w_n.

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{10}$$

$$BP = \begin{cases} 1 & \text{si } c > r \\ e^{1-r/c} & \text{si } c \leq r \end{cases} \tag{11}$$

La métrique TER mesure elle un coût de transformation d'une traduction en une traduction de référence à l'aide d'opérations d'édition élémentaires sur les tokens(substitution, insertion, suppression, déplacement). Le calcul d'un score TER est donné par l'équation 12.

$$TER = \frac{\text{nombre d'opérations effectuées}}{\text{nombre de mots de la traduction de référence}} \tag{12}$$

Les métriques automatiques partagent de nombreux défauts, notamment la contrainte de similarité très restrictive à des références déjà produites. En outre, l'interprétation de ces scores s'avère souvent difficile : un score BLEU élevé, par exemple, n'implique pas qu'une hypothèse sera correcte au niveau du sens (*adequacy*) ni de la grammaire (*fluency*). Cependant, ces mesures sont généralement conçues pour être appliquées et interprétées à l'échelle de corpus, et non à un niveau très local, comme celui des phrases. Il existe par exemple pour BLEU une variante appelée *smoothed* BLEU (sBLEU [Liang et al., 2006]) adapté à une évaluation à l'échelle de la phrase.

3 Motivations de ce travail

3.1 Mesures oracles

S'il est généralement admis que la Traduction Automatique Statistique a connu de grands progrès au cours des 15 dernières années, la qualité des traductions produites reste toujours globalement très imparfaite, et cette technologie ne peut pas encore être utilisée dans de nombreux contextes. Pour déterminer le potentiel d'amélioration des traductions relativement à un système particulier, il est possible de faire des mesures de performance dites *oracle*. Pour ce type de mesure, on cherche l'hypothèse d'un système qui, indépendamment de sa propre fonction de score, maximise le score d'une métrique d'évaluation particulière. Ce type de mesures a notamment permis de montrer un fort potentiel d'amélioration de la qualité des traductions des systèmes à base de segments selon les métriques traditionnelles [Wisniewski et al., 2010].
Afin de motiver nos travaux, nous avons souhaité observer certains types de transformations utiles pour améliorer les traductions d'un tel système en réalisant trois expériences oracle. Pour chacune d'elles, on suppose connu l'ordre des mots attendus dans la traduction de référence, mais chaque mot est remplacé par un token inconnu[9]. L'hypothèse ainsi obtenue a exactement la taille de la traduction de référence attendue, mais sa précision n-gramme est nulle. On considère alors une source lexicale particulière qui prédit un certain nombre de mots : pour chacun de ces mots, chaque emplacement où il apparaît dans la traduction de référence est découvert, signifiant que le mot apparaît en clair et pourra donc être reconnu comme tel par le calcul d'évaluation de cette nouvelle hypothèse. Nous comparons alors la meilleure hypothèse issue du décodage avec les trois variantes de l'oracle. Nous présentons des résultats pour nos systèmes français → anglais et anglais → français dans la Table 4.

Une première expérience oracle considère comme source lexicale l'hypothèse du système de traduction elle-même, ce qui correspond donc à utiliser les mots de l'hypothèse à l'emplacement où ils sont attendus dans la traduction de référence, ainsi qu'à les utiliser le nombre de fois nécessaire. Des gains importants d'environ 10 et 12 points BLEU, respectivement pour les sens de traduction français → anglais et anglais → français, montrent qu'un fort potentiel d'amélioration réside dans une meilleure organisation des

[9]Ceci est garanti en préfixant simplement chaque tokens par un tiret bas.

	français → anglais				anglais → français			
Oracles	BLEU	Δ BLEU	TER	Δ TER	BLEU	Δ BLEU	TER	Δ TER
-	47.95	-	32.13	-	23.11	-	60.32	-
1	58.19	+10.24	21.27	-10.86	35.63	+12.52	39.15	-21.17
2	70.60	+22.65	13.70	-18.43	46.72	+23.61	29.22	-31.10
3	87.42	+39.47	5.15	-26.98	76.09	+52.98	10.26	-50.06

TAB. 4 – Scores des métriques BLEU et TER pour la meilleure hypothèse issue du décodage et des expériences oracles pour le sens de traduction avec les sources lexicales suivantes : [1] tokens présents dans la meilleure hypothèse ; [2] tokens présents dans les 1000 meilleures hypothèses pour cette phrase ; [3] tokens présents dans les 1000 meilleures hypothèses des phrases appartenant au même document que la phrase à traduire. Les colonnes Δ donnent les écarts entre les scores du système de base et ceux de l'oracle.

mots de la meilleure hypothèse.
La Table 5 présente deux exemples dans lesquels la source lexicale (ici, la meilleure hypothèse de traduction) contient plusieurs mots de la traduction de référence. Ces mots sont conservés par l'oracle et réordonnés, puis les mots manquants sont ajoutés et préfixés par un "_". Ainsi, nous pouvons voir dans le deuxième exemple que le mot "sociétés" présent dans la source lexicale, a été utilisé par l'oracle pour former un 4-gramme de la traduction de référence, "pour les petites sociétés", qui n'était pas présent dans notre hypothèse de traduction.

Source lexicale	l' utilisation d' Internet est extrêmement limité sur l' île .
Oracle 1	sur l' île _, l' utilisation d' Internet est extrêmement _LIMITéE .

Source lexicale	1p hausse de l' impôt sur les sociétés pour les petites entreprises mis au rebut
Oracle 1	l' _AUGMENTATION de l' impôt sur les _BéNéFICES de _LA _SOCIéTé pour les petites sociétés _A _éTé _ABANDONNéE

TAB. 5 – Exemples produits par notre première expérience oracle

Dans une seconde expérience, nous avons considéré l'ensemble des mots appartenant aux 1 000 meilleures hypothèses produites par le système pour la phrase à traduire. Ici, des gains additionnels de 12 et 11 points BLEU sont obtenus (soit 22 et 23 par rapport au score initial). Ceci indique qu'un nombre important de mots attendus dans la traduction de référence n'apparaissent pas dans la meilleure hypothèse mais sont présents dans les meilleures hypothèses suivantes. Ce résultat, qui montre tout d'abord que les fonctions de score utilisées pour la recherche d'une meilleure hypothèse sont imparfaites, montre également ici qu'il est donc possible de se rapprocher de la traduction de référence en remplaçant des mots de la meilleure hypothèse par d'autres mots issus des meilleures hypothèses suivantes, donc atteignables par le système.

Une dernière expérience a été effectuée en considérant cette fois-ci tous les mots appartenant aux 1 000 meilleures hypothèses produites pour toutes les phrases du même document qu'une phrase à traduire. Ici les gains additionnels sont de 17 et 30 points BLEU (soit 40 et 53 par rapport au score initial). Il est donc possible d'obtenir une hypothèse de traduction très proche de la traduction de référence à l'aide des traductions produites par le système pour d'autres phrases dans le même contexte, puisqu'au final peu de mots attendus dans une traduction de référence semblent inatteignables pour le système considéré[10]. Cela met notamment en évidence l'importance d'une modélisation du contexte discursif d'une traduction (voir par exemple [Gong et al., 2012]).

[10]Il faut toutefois remarquer que la présence de mots inconnus en langue source ou de traductions non atteignables pose une vraie difficulté dont ne rendent pas compte nos mesures oracles : ces parties seront particulièrement difficiles à traduire puisque la présence de mots inconnus en langue cible ne permettra pas une prise en compte efficace du contexte.

3.2 Les limites de modélisation de la grammaticalité

Nous avons vu dans la section 2.4 que les modèles de langue typiquement utilisés étaient de type n-gramme. Un tel modèle ne peut donc pas évaluer la probabilité d'un mot au-delà des n-1 mots qui le précédent. Cette valeur de n étant généralement limitée à 4 ou 5 pour des raisons de place et de rareté des données, le contexte ainsi pris en compte est très local. Prenons l'exemple d'une traduction réalisée par un système automatique[11] :

- Phrase à traduire :
  The reference is used to evaluate machine translation and is written by a translator.
- Traduction :
  La référence est utilisée pour évaluer la traduction automatique et est ***écrit*** *par un traducteur.*

On constate ici que le mot "écrit" n'a pas été accordé en genre avec son sujet "la référence". Un modèle de langue ne peut souvent pas capturer ce genre d'erreur. En particulier ici, un modèle 5-gramme ne considère comme contexte pour le mot "écrit" le segment "traduction automatique et est", ce qui ne permet pas de modélisation très fine de la grammaticalité des énoncés produits. Aucun modèle décrit dans l'équation 6 ne modélise donc la grammaticalité globale des énoncés[12].
Tester la bonne formation d'une hypothèse partielle est donc un problème difficile, notamment parce que la grammaticalité ne peut être typiquement évaluée que sur la base d'un énoncé complet, qui n'est pas disponible en cours de décodage. Un travail récent [Schwartz et al., 2011] a proposé d'intégrer des connaissances syntaxiques en langue cible pendant la construction des hypothèses par le biais de modèles de langue syntaxiques incrémentaux, calculant des scores de dérivations syntaxiques pour des préfixes d'hypothèses. L'implémentation décrite s'avère cependant peu performante [13], et la validation expérimentale n'a pu être faite que pour des données de petite taille. Une autre approche déjà évoquée consiste à réévaluer une liste de meilleures hypothèses à l'aide de critères grammaticaux (voir par exemple [Carter and Monz, 2011]), ce qui à nouveau restreint les hypothèses considérées et ne mène en pratique pas à de fortes améliorations.

4 La recherche locale

4.1 État de l'art

Les premières expériences oracle présentées dans la section 3.1 ont permis de mettre en évidence qu'avec un meilleur parcours de l'espace de recherche il peut être possible d'améliorer une traduction de façon significative selon les métriques BLEU et TER. Une approche possible consiste à apprendre un système de traduction statistique apprenant à corriger les hypothèses d'un système particulier (symbolique ou statistique) [Simard et al., 2007]. Une autre approche, que nous étudions ici, consiste à améliorer itérativement une solution par recherche locale [Langlais et al., 2007]. Un algorithme simple (4.1) prend en entrée une *amorce*, ici la meilleure hypothèse produite par un système, et parcourt son voisinage en lui appliquant différentes opérations de transformation. Parmi l'ensemble des opérations possibles, l'algorithme applique celle qui maximise à chaque itération la fonction de score du décodeur (équation 6), et s'arrête dès qu'aucune amélioration n'est mesurée.

Le travail de Langlais et al. [2007] propose 6 opérations qui ont pour objectif de visiter des hypothèses proches qui sont susceptibles de contenir des améliorations ou corrections d'une hypothèse de départ.

[11] Traduction produite par le système Google Translate (`http://translate.google.fr/`) en juin 2012
[12] et *a fortiori* encore moins la cohérence discursive entre phrases.
[13] Par exemple, le temps de décodage moyen pour une phrase de 10 mots passe de 0.21s pour `moses` à 533s avec ce modèle additionnel.

Les opérations `move` (rapprochement de deux segments adjacents dans la phrase source mais traduits de manière distante) et `swap` (inversion de deux segments) déplacent des segments dans une hypothèse. Ces deux opérations réorganisent donc les segments d'une hypothèse dans le but d'améliorer son score, ce qui a un certain potentiel d'après notre première expérience oracle (cf. section 3.1). D'autres opérations (`split`, `merge`, `replace` et `bi-replace`) permettent quant à elles de remplacer certains segments de l'hypothèse par d'autres segments présents dans la table de traduction. Celles-ci présentent également le potentiel d'améliorer la traduction comme le montre notre seconde expérience oracle.
La complexité de ces opérations ainsi qu'une brève description sont indiquées dans la Table 6. Une trace de recherche locale illustrant quelques itérations est présentée dans la Figure 3. Sur la base de cet exemple on peut rapidement constater qu'une amélioration de la fonction de score ne correspond pas nécessairement à une amélioration de la traduction et peut même la dégrader.

Algorithme 1 Algorithme de recherche locale

ENTRÉE : *source* une phrase à traduire.

courant ← AMORCE(*source*)
boucler
  sCourant ← SCORE(*source*)
  s ← *sCourant*
  pour tout *h* ∈ VOISINAGE(*courant*) **faire**
    c ← SCORE(*h*)
    si *c* > *s* **alors**
      s ← *c*
      meilleur ← *h*
    fin si
    si *s* = *sCourant* **alors**
      retourner *courant*
    sinon
      courant ← *meilleur*
    fin si
  fin pour
fin boucle

FIG. 3 – Trace d'une recherche locale

Dans leur travail, Langlais et al. [2007] ont observé une légère amélioration des traductions obtenues. Le décodeur `Pharaoh`, système état de l'art de l'époque, était utilisé pour produire les amorces servant de

Opération	Complexité	Description
`move`	O(N)	rapproche deux segments cibles distants si les deux segments sources correspondant sont adjacents
`swap`	O(N)	inverse deux segments cibles adjacents
`replace`	O(N x T)	change la traduction d'un segment source par une autre traduction présent dans la table de traduction
`merge`	O(T x N)	fusionne deux segments sources et remplace les deux segments cibles corresopndants par une traduction du nouveau segment source créé
`split`	O(N x S x T^2)	scinde un segment source en deux parties et remplace la segment cible correspondant par des traductions de deux nouveaux segments créés
`bi-replace`	O(T^2 x N)	change la traduction de deux segments adjacents simultanément

TAB. 6 – Opérations utilisées par la recherche locale de Langlais et al. [2007]. **N** : nombre de segments dans la phrase source ; **T** : nombre maximum de traductions par segment ; **S** : nombre moyen de tokens par segment

point de départ pour la recherche locale. Un travail plus récent [Monty, 2010] a cependant montré une dégradation assez nette des traductions obtenues par ce biais en utilisant un système état de l'art actuel (`moses`) pour produire les amorces, ce qui démontre notamment les progrès effectués par ces systèmes relativement aux fonctions de score et au parcours de l'espace de recherche. Les dégradations observées mettent donc en évidence les limites de la fonction de score maximisée (de façon heuristique) durant la recherche locale, laquelle pourrait notamment être enrichie par de nouveaux modèles rendant mieux compte de la qualité d'une traduction.

4.2 Description du cadre expérimental

Dans notre travail présenté ici, nous avons réimplémenté un décodeur réalisant de la recherche locale telle que décrite dans Langlais et al. [2007]. En revanche, nous avons choisi une implémentation du `move` différente. Le `move` défini dans Langlais et al. [2007] avait pour contrainte que si deux segments source contigus étaient traduits de façon distante, cette opération avait pour possibilité de rapprocher les deux segments cible. Le `move` que nous avons implémenté n'a pas cette contrainte : un segment cible peut être déplacé n'importe où dans l'hypothèse. De cette façon, notre opération `move` inclut les déplacements possibles engendrés par l'opérateur `swap`, notre recherche locale ne reprend donc pas cette opération qui était de plus très peu utile dans le travail de Langlais et al. [2007].

Nous nous sommes concentré sur la paire de langues anglais ↔ français, en réutilisant les données d'apprentissage et d'évaluation fournies pour la campagne d'évaluation internationale WMT'12[14]. Les données d'évaluation sont composées de conditions hétérogènes, dans lesquelles les textes à traduire proviennent eux-mêmes de traductions depuis d'autres langues. Nous n'avons retenu que la partie des corpus d'évaluation pour lesquelles les phrases en anglais à traduire ont été originellement écrites en anglais, et avons fait de même pour le français[15]. Ainsi les corpus utilisés pour la traduction anglais → français et français → anglais sont deux corpus différents. Le corpus de développement quant à lui rassemble des documents pour lesquels nous n'avons pas tenu compte de la langue d'origine. Les données utilisées sont décrites dans les Tables 7 et 8.
Pour les pré-traitements, nous avons réutilisés ceux effectués pour les soumissions du LIMSI à la campagne WMT'2012 [Le et al., 2012]. Les tables de traduction ont été estimées par échantillonnage aléatoire

[14] `http://http://www.statmt.org/wmt12/`

[15] On constate des performances très différentes si l'on traduit par exemple un texte originellement écrit en anglais, et si l'on traduit un texte originellement en tchèque puis traduit en anglais. Nous nous sommes donc initialement placé dans des conditions expérimentales plus interprétables, mais dans lesquelles les performances du système seront plus difficiles à améliorer.

	anglais → français		
		source	cible
	# phrases	# tokens	# tokens
apprentissage	10M	317M	382M
développement	505	12 736	15 534
évaluation	370	8 778	10 073

TAB. 7 – Nombre de phrases et de tokens pour les corpus d'apprentissage, développement et évaluation pour notre tâche de traduction anglais → français.

	français → anglais		
		source	cible
	# phrases	# tokens	# tokens
apprentissage	10M	382M	317M
développement	505	15 534	12 736
évaluation	352	12 216	9 388

TAB. 8 – Nombre de phrases et de tokens pour les corpus d'apprentissage, développement et évaluation pour notre tâche de traduction français → anglais.

de 1 000 exemples par segment à traduire, ce qui permet d'obtenir des performances très proches d'un système utilisant l'ensemble des données d'apprentissage [Callison-Burch et al., 2005], mais en un temps considérablement réduit (1h10 contre 206h). L'optimisation des modèles est réalisée par minimisation du taux d'erreur (MERT), et la construction des hypothèses de traduction utilise le système état de l'art fondé sur les segments `moses` [Koehn et al., 2007].

Le modèle de langue que nous avons utilisé a été appris sur un très grand corpus de données : 6 milliards de tokens pour l'anglais et 2.5 milliards de tokens pour le français. Il est de type 4-gramme avec un lissage Knesser-Ney [Chen and Goodman, 1998]. Ce grand modèle de langue a ralenti de façon très significative nos expériences mais nous a donné l'assurance d'avoir un système de base très compétitif et donc suffisamment difficile à améliorer.

Dans notre travail, nous avons utilisé pour évaluer nos résultats les 2 métriques TER et BLEU avec une seule traduction de référence, celle fournie pour WMT'12.

4.3 Résultats des expériences

Nous avons mené de premières expériences dans le but d'essayer de reproduire les résultats de l'état de l'art en recherche locale. Pour ces expériences, nous avons essayé chaque opération seule et la combinaison de toutes les opérations. Nous n'avons pas utilisé l'opérateur `bi-replace` que nous avons choisi de traiter directement comme l'une des opérations jointes évoquées dans la section 6.2. Les amorces utilisées dans nos expériences sont des traductions produites par le système `moses`. Les scores obtenus par ce système apparaissent sous l'appellation "*baseline*" dans les différents tableaux de résultats.

Pour ces expériences, la recherche locale trouve un maximum local à la fonction de score après 8 itérations en moyenne pour le décodage d'une phrase. Le modèle de langue étant appris sur un grand nombre de données, et les voisinages des hypothèses étant particulièrement grands, il a fallu environ 53 heures pour décoder notre corpus anglais→français et 45 heures pour notre corpus français→anglais dans une configuration n'utilisant qu'un seul *thread* d'exécution. Le calcul des scores du modèle de langue occupe environ 86% du temps du décodage. Le voisinage d'une hypothèse peut compter jusqu'à plusieurs centaines de milliers d'hypothèses qui doivent toutes être évaluées par le modèle langue, une évaluation qui peut durer plus d'une heure (la durée dépend majoritairement de la taille des phrases) pour certaines itérations. Notre décodeur met en cache le score du modèle de langue pour le dernier million d'hypothèses calculé de façon à ce que le modèle de langue ne recalcule pas plus d'une fois le score d'une hypothèse

du voisinage proche.

Les résultats que nous avons obtenus sont présentés dans les Tables 9 et 10.

	français→anglais					
	Scores		Répartition des opérations effectuées			
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`
baseline	47.95	32.13	-	-	-	-
`move`	47.80	32.22	100.00% (36)	-	-	-
`replace`	47.14	32.99	-	100.00% (467)	-	-
`merge`	47.08	32.68	-	-	100.00% (153)	-
`split`	48.07	32.26		-	-	100.00% (1658)
`move+replace +merge+split`	45.91	33.63	4.22% (93)	12.44% (274)	10.26% (226)	73.07% (1609)

TAB. 9 – Résultats de notre décodeur par recherche locale pour le sens de traduction français→anglais

	anglais→français					
	Scores		Répartition des opérations effectuées			
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`
baseline	23.11	60.31	-	-	-	-
`move`	22.98	60.52	100.00% (61)	-	-	-
`replace`	22.67	59.64	-	100.00% (650)	-	-
`merge`	23.09	60.27	-	-	100.00% (16)	-
`split`	21.67	61.29	-	-	-	100.00% (2394)
`move+replace +merge+split`	21.57	60.83	5.18% (158)	15.62% (476)	0.79% (24)	78.41% (2390)

TAB. 10 – Résultats de notre décodeur par recherche locale pour le sens de traduction anglais→français

Ces premiers résultats nous montrent une assez nette dégradation des traductions produites par le système `moses` comme le constatait déjà avant nous Monty [2010]. En effet, la plupart des configurations que nous avons testée dégradent les scores BLEU et TER de la *baseline*. Les pertes vont jusqu'à 2 points BLEU dans la configuration `move+replace+merge+split` pour le sens de traduction français→anglais. Toutefois, quelques expériences ont montré de petites améliorations, c'est le cas notamment du `replace` seul (anglais→français) pour laquelle le score TER a baissé de 0,6 point et pour le `split` seul (français→anglais) où l'on observe un gain de 0,13 point BLEU.
Les configurations où les scores sont les moins pénalisés sont celles où très peu d'itérations sont effectuées par la recherche locale. C'est notamment le cas lorsque seuls les déplacements de segments avec l'opérateur `move` sont permis. L'opérateur de resegmentation `split` est de loin le plus utilisé. Cependant en analysant les traces de nos expériences il apparaît qu'un grand nombre d'opérations `split` effectuées ne changent pas la traduction (voir les Figure 4). Cela signifie que la fonction de score préfère les traductions composées de petits segments, ce qui explique également le peu d'opérations `merge` observées.

En analysant la trace produite par notre décodeur, il est possible d'observer des améliorations effectuées par recherche locale. La Figure 5 présente une suite d'améliorations réalisées sur plusieurs itérations. Dans cet exemple, la première itération présentée fait une opération `split` qui ajoute à la traduction le mot "dans". L'ajout de cette préposition a été permis par la recherche locale grâce à un gain pour le score du modèle de traduction et l'ajout d'un mot qui fait gagner une unité à la pénalité lexicale. Cette itération rapproche l'hypothèse de traduction de la traduction de référence, elle fait donc augmenter le score BLEU.
Dans l'itération suivante une opération `move` est effectuée, il améliore le score du modèle de langue au détriment de la distorsion. L'opération réalisée ici rend directement possible l'itération suivante qui ajoute un "de" entre "note" et "confession" rendant la phrase grammaticalement plus correcte. Après cette dernière itération, il ne reste que très peu de différences avec la traduction de référence. Notam-

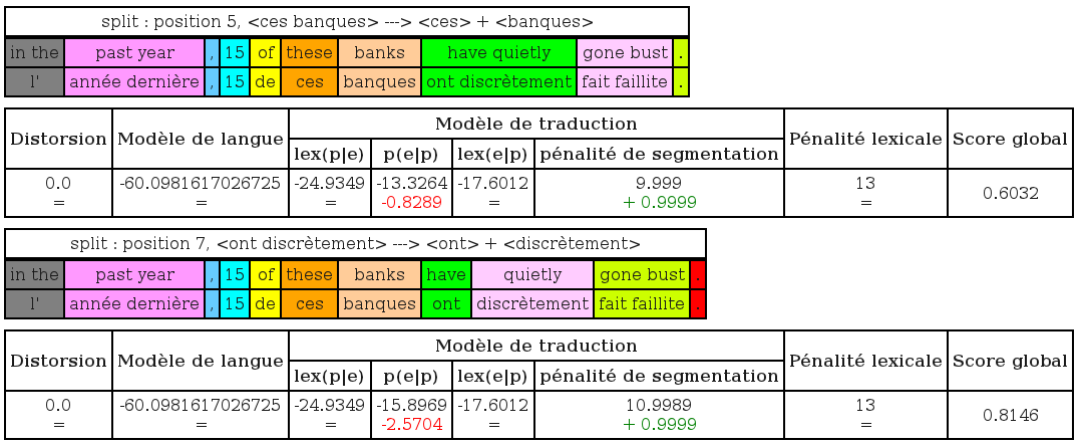

FIG. 4 – Exemples de `split` qui ne changent pas la traduction

ment, l'hypothèse de traduction utilise la mot "note" tandis que la traduction de référence préfère le mot "lettre" sans doute plus approprié.

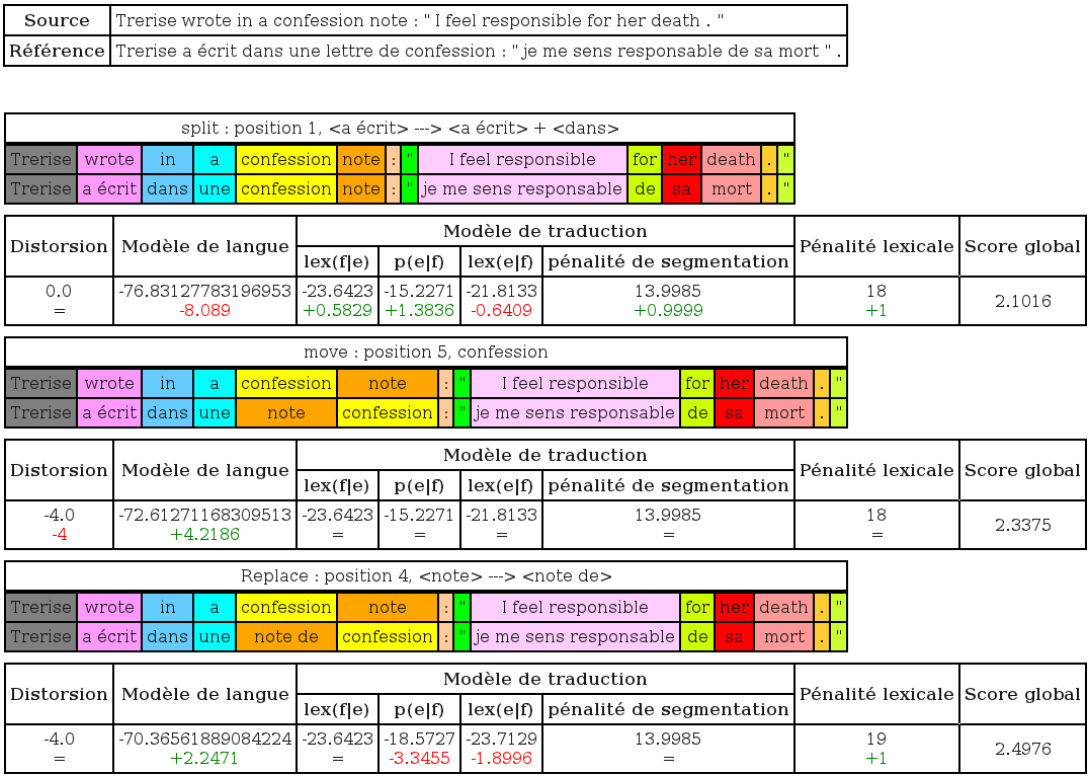

FIG. 5 – Exemple d'améliorations effectuées par la recherche locale (les premières itérations ont été masquées pour plus de lisibilité)

Certains exemples mettent aussi en évidence les limites des métriques automatiques BLEU et TER. En effet, dans l'exemple de la Figure 6, on peut voir que l'amorce est grammaticalement incorrecte. Le verbe "s'éléveront" est au pluriel alors que son sujet est au singulier. La recherche locale choisie de le remplacer par l'intermédiaire d'une opération `split` par "sera". La traduction produite devient de cette façon plus correcte. Cependant on remarque que le mot "s'éléveront" était présent dans la traduction de référence, en remplaçant ce mot la recherche locale s'est donc éloignée de la traduction de référence. Ici

nous sommes donc dans une situation où malgré l'amélioration de l'hypothèse de traduction, les scores des métriques BLEU et TER ont été dégradés.

Source	the Treasury estimates that the interest on the loan will amount to £450m in 2009 .
Référence	le Trésor estime que les intérêts du prêt s' élèveront en 2009 , à £450millions .

Amorce							
the Treasury	estimates that	the interest	on	the loan	will amount to	£450m	in 2009 .
le Trésor	estime que	l' intérêt	sur	l' emprunt	s' élèveront à	£450m	en 2009 .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-64.40814047973875	-19.7363	-9.6303	-28.7235	8.9991	16	0.3994

split : position 5, <s' élèveront à> ---> <sera> + <de>								
the Treasury	estimates that	the interest	on	the loan	will amount	to	£450m	in 2009 .
le Trésor	estime que	l' intérêt	sur	l' emprunt	sera	de	£450m	en 2009 .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-59.734353257979436 +4.6738	-24.1394 -4.4031	-11.6084 -1.9781	-20.2381 +8.4854	9.999 +0.9999	15 -1	1.2463

FIG. 6 – Exemple d'amélioration d'une traduction qui n'améliore pas les scores BLEU et TER

Nous venons donc de voir que la recherche locale au cours de ses itérations peut passer par des phases d'améliorations des traductions. Pour certaines phrases la traduction de référence est cependant inatteignable. La Figure 7 présente une exemple où la traduction de référence ne traduit pas une partie la phrase source : "on the recent evening". Or un système de traduction automatique cherchera à traduire cette partie[16].

Source	with a few exceptions , the street looked much the same on a recent evening .
Référence	à peu d' exceptions près , la rue a toujours la même allure .

Amorce						
with a few exceptions , the	street looked	much the	same	on a recent	evening	.
à quelques exceptions près , les	passants regardaient	le	même	lors d' une récente	soirée	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-61.25382916084621	-27.9935	-10.0931	-40.8566	6.9993	16	-1.3489

FIG. 7 – Exemple de traduction de référence inatteignable

Nous venons de voir quelques exemples d'améliorations de traductions produites par la recherche locale. Celles-ci sont toutefois minoritaires par rapport au dégradations. La Figure 8 présente un exemple d'importantes dégradations syntaxiques opérées par la recherche locale.
Ainsi ces résultats mettent en évidence que malgré un meilleur score obtenu par la fonction 6, les traductions n'ont globalement pas été améliorées, au contraire pour la plupart des configurations celles-ci sont dégradées.
Afin de mieux comprendre ces résultats nous avons analysés les traces produites lors de nos expériences, notamment à l'aide de nouvelles expériences oracle, dans la section 4.4.1.

[16] L'identification de groupes de mots à supprimer ou optionnels pour une traduction n'est pas un problème actuellement étudié à notre connaissance.

Source	" they will have thought this was being dealt with internally at the MoD and Cabinet Office , " he said .
Référence	" ils auront pensé que cette affaire serait restée entre les murs du Ministère de la Défense et du Cabinet Office " a-t-il déclaré .

split : position 5, <traitée> ---> <traitée> + <avec>																
"	they will	have	thought this	was	being dealt	with	internally	at the	MoD	and	Cabinet Office	,	"	he	said	.
"	ils	ont	pensé que c'	était	traitée	avec	à l' intérieur	du	MoD	et	chef de cabinet	,	"	qu' il	dit	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-98.33788311755251 -6.5946	-40.4064 +1.2279	-34.27 -1.3097	-41.4946 -0.7547	16.9982 +0.9999	24 +1	0.3601

split : position 5, <traitée> ---> <d' être> + <traité>																
"	they will	have	thought this	was	being	dealt	with	internally	at the	MoD	and	Cabinet Office	,	"	he	said	.
"	ils	ont	pensé que c'	était	d' être	traité	avec	à l' intérieur	du	MoD	et	chef de cabinet	,	"	qu' il	dit	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-101.49887193321473 -3.161	-43.4137 -3.0073	-36.9085 -2.6385	-47.2925 -5.7979	17.9981 +0.9999	26 +2	0.4618

FIG. 8 – Exemple de dégradations syntaxiques (les premières itérations ont été masquées pour plus de lisibilité)

4.4 Expériences oracles

4.4.1 Résultats sur la recherche locale

Pour voir l'impact de la recherche locale nous avons utilisé la métrique sBLEU comme fonction de score au niveau des phrases. Les Figure 9 et 10 montrent les évolutions du sBLEU entre phrases successives de documents (séparés par des barres verticales rouges).

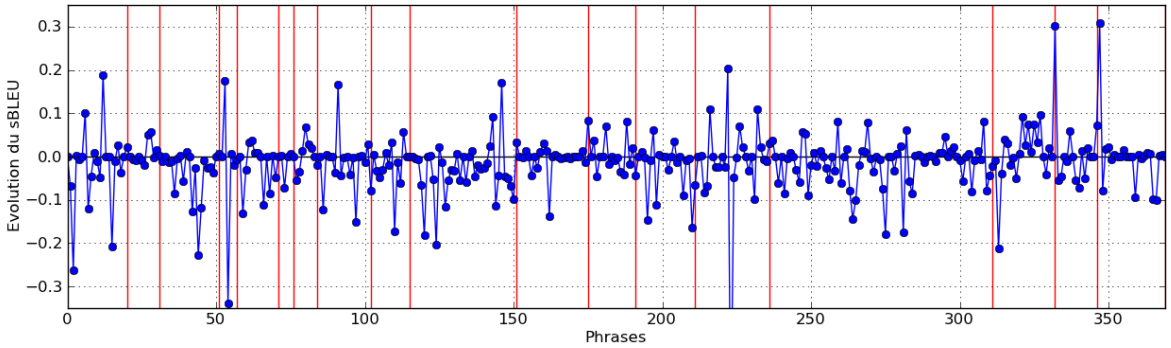

FIG. 9 – Évolution du sBLEU pour chaque phrase du corpus dans le sens de traduction français→anglais. Les barres verticales rouges indiquent la fin de chaque document du corpus.

On constate tout d'abord que la recherche locale n'effectue pas que des dégradations. En effet, pour de nombreuses phrases le score sBLEU n'évolue pas de manière significative et se trouve même parfois amélioré. Cependant, les dégradations sont plus nombreuses et de plus forte amplitude, ce qui explique la baisse du score BLEU sur l'ensemble de chaque corpus. À l'échelle des documents présents dans nos corpus (délimités par les lignes rouges) il est possible d'observer que l'impact de la recherche locale sera différent suivant le document traité. Pour certains documents la recherche locale améliorera plutôt la qualité de la traduction, mais, au niveau discursif, les dégradations ou améliorations opérées par la recherche locale ne semblent pas dépendre du document.

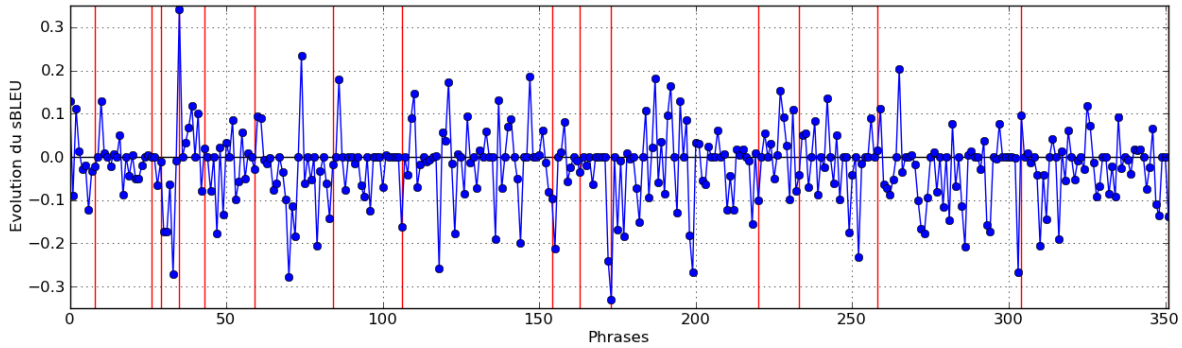

FIG. 10 – Évolution du sBLEU pour chaque phrase du corpus dans le sens de traduction anglais→français. Les barres verticales rouges indiquent la fin de chaque document du corpus.

4.4.2 Sélection d'une hypothèse issue de la recherche locale

Dans le but de mieux évaluer le potentiel de la recherche locale nous avons réalisé un oracle capable de choisir la meilleure traduction (selon la métrique sBLEU) entre l'amorce et sa modification par la recherche locale, pour chaque phrase. Nos résultats sont présentés dans les Table 11 et 12.

	français→anglais				
	Scores		Taux de phrases améliorées		
Configuration	BLEU	TER	Amélioration	dégradation	sans impact
baseline	47.95	32.13	-	-	-
`move`	47.99	32.09	1.42%	3.41%	95.17%
`replace`	49.51	31.20	26.99%	36.65%	36.36%
`merge`	48.47	31.69	9.66%	21.02%	69.32%
`split`	49.65	31.07	30.68%	24.43%	44.89%
`move+replace+merge+split`	49.54	30.97	28.12%	45.45%	26.42%

TAB. 11 – Résultats oracle validant la recherche locale uniquement sur les phrases ayant obtenu un score sBLEU supérieur à celui de l'amorce

	anglais→français				
	Scores		Taux de phrases améliorées		
Configuration	BLEU	TER	Amélioration	dégradation	sans impact
baseline	23.11	60.31	-	-	-
`move`	23.24	60.31	1.89%	5.41%	92.70%
`replace`	24.19	58.71	30.27%	37.30%	32.43%
`merge`	23.16	60.21	3.24%	1.08%	95.68%
`split`	24.02	59.27	29.46%	42.97%	27.57%
`move+replace+merge+split`	24.48	58.79	33.24%	49.73%	17.03%

TAB. 12 – Résultats oracle validant la recherche locale uniquement sur les phrases ayant obtenu un score sBLEU supérieur à celui de l'amorce

Ces expériences montrent une amélioration significative pouvant dépasser 1,5 point BLEU. Dans le détail on peut voir que plus les configurations font d'itérations (par exemple `move+replace+merge+split` et `split`) et plus le potentiel d'amélioration sera grand. Dans ces configurations la recherche locale améliore le score sBLEU pour environ 30% des phrases.
Nous avons aussi souhaité observer si la recherche locale visitait des états intermédiaires pouvant être meilleurs que l'état final. Pour cela nous avons fourni à l'oracle toutes les traductions calculées à chaque itération intermédiaire par la recherche locale en plus de l'amorce et de l'itération finale. Nos résultats (Table 13 et 14) dévoilent un gain potentiel encore plus important pouvant aller jusqu'à 2,6 points BLEU.

	français→anglais	
	Scores	
Configuration	BLEU	TER
baseline	47.95	32.13
`move`	48.00	32.09
`replace`	49.85	30.87
`merge`	48.50	31.68
`split`	49.84	30.90
`move+replace+merge+split`	50.61	30.36

TAB. 13 – Résultats oracle choisissant l'amorce ou un meilleur état calculé par la recherche locale pour chaque phrase dans le sens de traduction français→anglais.

	anglais→français	
	Scores	
Configuration	BLEU	TER
baseline	23.11	60.31
`move`	23.27	60.33
`replace`	24.32	58.38
`merge`	23.16	60.21
`split`	24.32	58.71
`move+replace+merge+split`	25.12	57.89

TAB. 14 – Résultats oracle choisissant l'amorce ou un meilleur état calculé par la recherche locale pour chaque phrase dans le sens de traduction anglais→français.

Ainsi nous avons pu montrer à l'aide de ces expériences que si la recherche locale telle que décrite par Langlais et al. [2007] et par Monty [2010] dégrade les traductions, elle posséde néanmoins un fort potentiel d'amélioration.

4.4.3 Expériences oracle sur le voisinage des hypothèses

Nous avons aussi cherché à comparer le voisinage de la meilleure hypothèse calculée par `moses` avec celui de la meilleure hypothèse trouvée par la recherche locale. Ici nous avons exécuté nos expériences oracle sur les 1 000 meilleures hypothèses trouvées pour chaque phrase par `moses` et notre décodeur. Ces expériences retiennent donc la meilleure hypothèse selon la métrique sBLEU parmi les 1 000 hypothèses calculées. L'expérience oracle faite sur la liste produite par `moses` montre une amélioration pouvant aller jusqu'à 10 points BLEU (voir les Tables 25 et 24). Ce potentiel d'amélioration montre les limites des modèles utilisés par la fonction de score à maximiser. En effet, la meilleure hypothèse de traduction trouvée par `moses` (celle de score le plus élevé) n'est que rarement la meilleure traduction selon la métrique sBLEU.

	anglais→français	
Configuration	BLEU	TER
baseline produite par `moses`	23.11	60.31
oracle (`moses`)	31.24	52.67
baseline produite par recherche locale	21.57	60.83
oracle (recherche locale)	32.20	53.10

TAB. 15 – Résultats oracle obtenus sur la liste des 1000-meilleures hypothèses produite par `moses` et la recherche locale pour le sens de traduction anglais→français.

Les mêmes expériences oracle réalisées cette fois sur le voisinage des hypothèses de traduction produites par la recherche locale montrent là aussi une amélioration pouvant aller jusqu'à 10 points BLEU si l'on considère la meilleure hypothèse selon sBLEU. `moses` et notre décodeur produisent donc des voisinages pour lesquels nos expériences oracle donnent des performances similaires. Cependant, en comparant le contenu des listes des 1 000 meilleures hypothèses produites par les 2 décodeurs, on constate des

	français→anglais	
Configuration	BLEU	TER
baseline produite par `moses`	47.95	32.13
oracle (`moses`)	58.33	25.25
baseline produite par recherche locale	45.91	33.63
oracle (recherche locale)	58.34	26.29

TAB. 16 – Résultats oracle obtenus sur la liste des 1000-meilleures hypothèses produite par `moses` et la recherche locale pour le sens de traduction français→anglais.

différences importantes. En effet, par exemple pour le sens de traduction français→anglais, sur la totalité des 352 000 hypothèses contenues pour chacune des deux listes (1 000 hypothèses pour chacune des 352 phrases du corpus), seulement 8,5% des hypothèses de la liste calculées par `moses` se trouvent dans la liste produite par la recherche locale. La recherche locale produit donc un voisinage très différent de celui de `moses` mais met aussi en évidence les limites des modèles de la fonction de score utilisée : là aussi, parmi les 1 000 hypothèses calculées pour chaque phrase, la meilleure traduction n'est pas celle de meilleur score.

4.4.4 Recherche locale oracle guidée par la métrique sBLEU

Dans les expériences présentées ici nous avons remplacé la fonction de score à maximiser par la métrique sBLEU. La recherche locale oracle cherchera donc à effectuer des opérations directement dans le but de rendre l'hypothèse de traduction similaire à la traduction de référence, en étant localement guidée par sBLEU.

Les résultats décrits dans les Tables 17 et 18 montrent de fortes améliorations des scores BLEU et TER. Ainsi, pour les meilleures configurations nous obtenons un gain de respectivement 35,16 et 37,09 points BLEU pour nos configurations français → anglais et anglais → français. Ces gains très importants mettent en évidence qu'avec une fonction de score mieux définie il est possible d'améliorer nettement les traductions produites selon les métriques BLEU et TER.
De façon plus précise, on constate que dans l'expérience où seul l'opérateur `split` est activé les gains sont beaucoup plus forts que dans l'expérience avec l'opérateur `merge` seul. L'écart entre ces 2 expériences est de 20,77 points BLEU pour le sens de traduction français → anglais et 14,33 points BLEU pour le sens de traduction anglais → français. Notre oracle a donc plus de chances d'améliorer la traduction en augmentant le nombre de segments qu'en le diminuant. Cela signifie qu'un certain nombre de longs segments de la table de la traduction qui pourraient être obtenus par une ou des opérations `merge` ne sont généralement pas de très bonne qualité.

On peut aussi noter dans les expériences avec l'opérateur `move` seul un gain d'environ 3 points BLEU. Sans changement de vocabulaire, puisque le move ne permet que des déplacements, il donc est aussi possible d'améliorer les traductions, ce qui est cohérent avec notre première expérience oracle de la section 3.1. Néanmoins, on peut constater une différence de près de 7 points BLEU entre les deux oracles. Cet écart met en évidence la sous-optimalité de la recherche locale oracle qui agit sur des segments et ne permet pas en l'état des transformations plus précises de l'hypothèse en agissant directement sur les mots.
Pour la plupart des phrases notre oracle trouve une ou plusieurs améliorations possibles du score sBLEU. Cependant chacune de ces améliorations entraîne majoritairement une baisse significative du score de la fonction maximisée (équation 6) par notre système de traduction (voir Figure 11). Les traductions produites par la recherche locale oracle ne sont donc pas atteignables en maximisant cette fonction. Ces expériences mettent en évidence les limites de cette fonction de score et des modèles utilisés, qui ne permettent pas d'identifier de meilleures traductions se trouvant dans notre espace de recherche[17].
En observant les traductions produites par cet oracle, nous avons pu détecter quelques abérrations dues à l'utilisation de la métrique BLEU comme fonction de score. Notamment la pénalité de concision a poussé notre oracle à raccourcir l'hypothèse de traduction pour que sa taille se rapproche de celle de la

[17] Il faut toutefois garder à l'esprit que l'utilisation d'une unique traduction de référence peut avoir mal récompensé des hypothèses pourtant acceptables.

	français→anglais					
	Scores		Répartition des opérations effectuées			
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`
baseline	47.95	32.13	-	-	-	-
`move`	51.04	31.68	100.00% (297)	-	-	-
`replace`	66.73	19.44	-	100.00% (1470)	-	-
`merge`	50.18	30.62	-	-	100.00% (257)	-
`split`	70.95	18.10	-	-	-	100.00% (1209)
`move+replace +merge+split`	83.11	10.54	15.55% (325)	28.52% (596)	9.33% (195)	46.60% (974)

TAB. 17 – Résultats d'une recherche locale guidée par la métrique BLEU

	anglais→français					
	Scores		Répartition des opérations effectuées			
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`
baseline	23.11	60.31	-	-	-	-
`move`	26.22	59.58	100.00% (382)	-	-	-
`replace`	44.46	37.88	-	100.00% (2863)	-	-
`merge`	28.28	54.17	-	-	100.00% (301)	-
`split`	42.61	42.30	-	-	-	100.00% (1504)
`move+replace +merge+split`	60.20	26.84	17.00% (622)	51.86% (1897)	10.52% (385)	20.61% (754)

TAB. 18 – Résultats d'une recherche locale guidée par la métrique BLEU

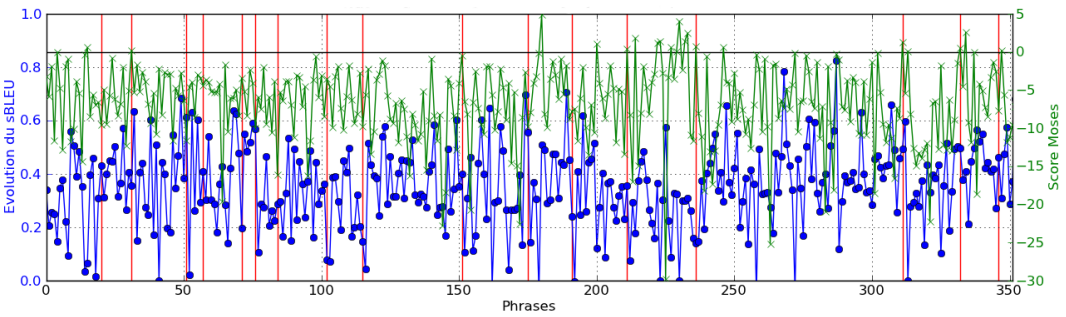

FIG. 11 – Comparaison entre l'évolution du sBLEU et le score `moses`

traduction de référence. L'oracle aura également tendance à déplacer en début de traduction un segment qui s'intercale entre deux n-grammes dont la concaténation correspondrait à un plus grand n-gramme présent dans la traduction de référence. Ces déplacements ont pour conséquences de rendre les phrases produites grammaticalement très incorrectes en juxtaposant par exemple plusieurs prépositions ou signes de ponctuation. Quelques exemples de ces effets sont présentés dans la Table 19. La Figure 12 présente des exemples de dégradations grammaticales opérées par la recherche locale. L'amorce, relativement correcte grammaticalement, est itérativement dégradée : tous les scores des différents modèles sont tour à tour pénalisés malgré une augmentation du sBLEU. Cet exemple particulier met en évidence les défauts des métriques automatiques couramment utilisées qui privilégient un rapprochement surfacique de la traduction de référence sans considérer, à l'échelle de la phrase, les problèmes syntaxiques que cela peut poser.

4.5 Détection automatique des cas de succès de la recherche locale

Nos expériences oracle présentées dans la section 4.4.1 ont montré que la recherche locale pouvait améliorer environ 30% des phrases de nos corpus. Nous avons donc essayé de développer un classifieur dans

Source	après la faillite de Washington Mutual vendredi , les autorités ont organisé le rachat des activités bancaires de Wachovia par sa rivale Citigroup .
Amorce	" it is us , but I will be there , " he told his supporters , bringing home the applause .
Recherche locale	"] " between us , but I plan to be there , " he told advocates , bringing them of applause .
Référence	" between us , but I plan to be there , " he told activists , eliciting from them a huge round of applause .

Source	new probe into US attorney affair
Amorce	une nouvelle sonde américaine dans l' affaire de la Justice
Recherche locale	. nouvelle enquête dans l' affaire conseil
Référence	nouvelle enquête dans l' affaire des procureurs américains .

Source	" off the road on the left , " she wrote , " is the brown-with-white-trim modern public school , with its well-kept yards and playgrounds , which Howard Miller always looks after , though he can scarcely read and write . "
Amorce	" la route à la gauche " , elle écrit " le brown-with-white-trim de l' école publique moderne , avec ses chantiers et abords bien entretenus , Howard Miller a toujours l' air après qu' il peut à peine lire et écrire " .
Recherche locale	la route sur la gauche " , écrit " brown-with-white-trim moderne école publique , chantiers et abords bien entretenus dont Howard Miller a toujours , " après , bien qu' il de à peine lire et écrire .
Référence	" tu n' embrasses pas ton professeur blanc parce qu' il est blanc - je veux dire qu' il y a ici , une ligne de démarcation culturelle " a déclaré Melle Nathiri .

TAB. 19 – Exemples de traduction par la recherche locale oracle

l'objectif de repérer automatiquement les dégradations effectuées par la recherche locale afin de déterminer automatiquement quand utiliser l'amorce ou la sortie de la recherche locale.
Nous avons pour cela défini différents traits pour caractériser une itération de la recherche locale. Ces traits sont calculés sur la trace du décodage effectué par la recherche locale :

- nombre d'itérations effectuées
- nombre d'itérations effectuées / nombre de mots
- nombre de `split`
- nombre de `split` / nombre de mots
- nombre de `move`
- nombre de `move` / nombre de mots
- nombre de `merge`
- nombre de `merge` / nombre de mots
- nombre de `replace`
- nombre de `replace` / nombre de mots
- nombre de segments
- nombre de segments / nombre de mots
- nombre de segments de l'amorce - nombre de segments de l'hypothèse
- (nombre de segments de l'amorce - nombre de segments de l'hypothèse) / nombre de mots
- score de l'amorce - score de l'hypothèse
- (score de l'amorce - score de l'hypothèse) / nombre de mots
- (score obtenu à l'itération précédente - score de l'hypothèse) / nombre de mots
- (score de l'amorce - score de l'hypothèse) / nombre d'itérations
- 4 dernières opérations effectuées
- nombre de `split identité`

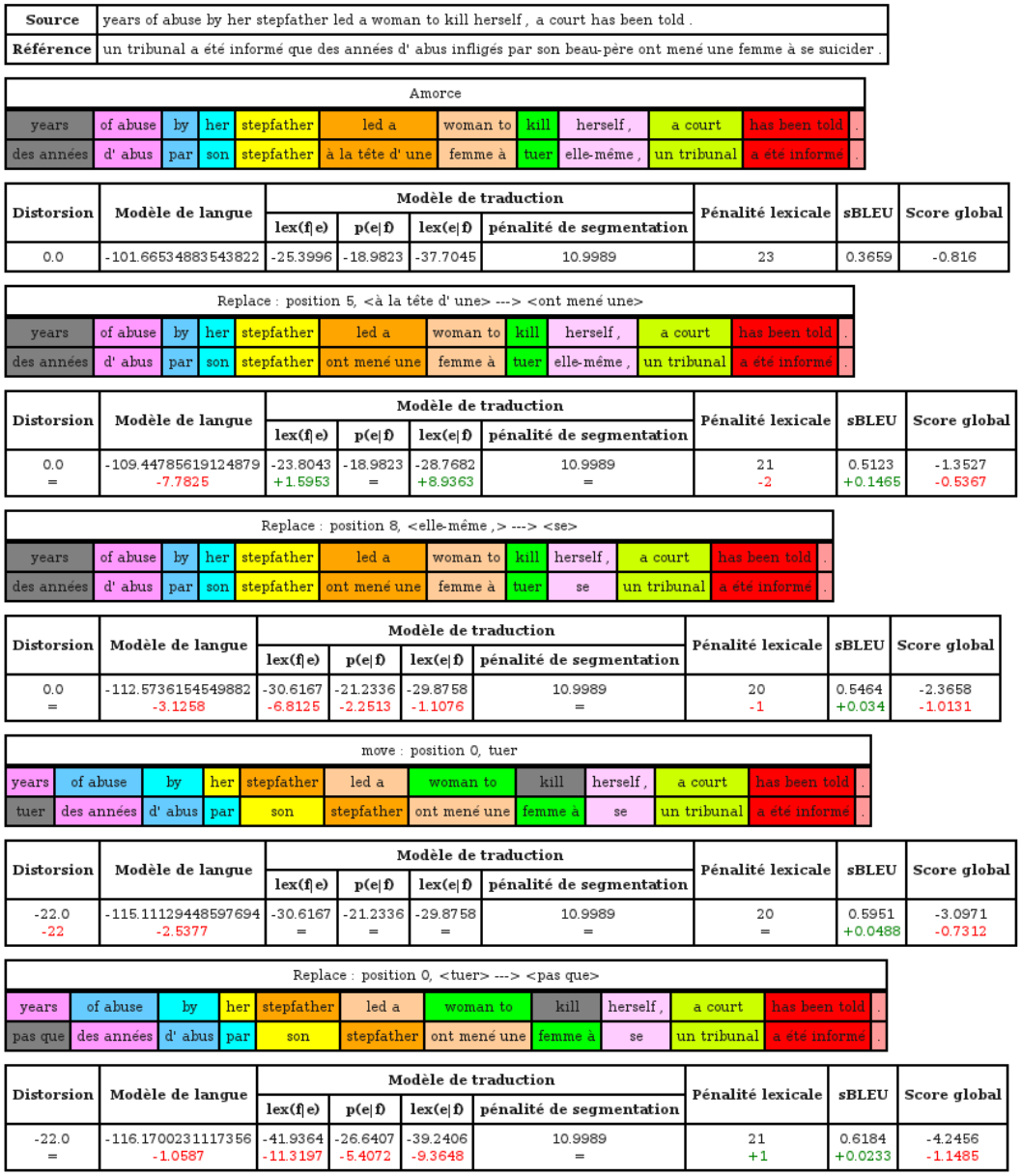

FIG. 12 – Exemple de recherche locale oracle

– nombre de `split identité` / nombre de mots

Les 4 dernières opérations effectuées sont réparties en 4 traits différents. Chaque trait a pour valeur un label identifiant l'opération effectuée : `split`, `move`, `merge`, `replace` ou `split identité`. L'opération `split identité` identifie une sous-classe du `split` qui indique qu'un `split` a été fait mais sans modifier la traduction obtenue (voir Figure 4). Un label identifiant qu'aucune itération n'a été effectuée a aussi été ajouté pour les itérations précédées de moins de 3 itérations. Par exemple, s'il s'agit pour une phrase de la première itération effectuée par la recherche locale parmi les 4 traits, 3 seront initialisés avec ce label tandis que seul le trait correspondant à la dernière opération effectuée aura pour valeur le label d'une opération.
Le classifieur utilisé est un classifieur binaire de type SVM[18]. Chaque itération est étiquetée 1 ou -1. L'étiquette 1 indique une amélioration du score sBLEU par l'itération comparé à l'amorce, l'étiquette -1

[18] Nous avons utilisé le classifieur SVM^{light} : `http://svmlight.joachims.org/`

indique une dégradation ou aucune évolution du score sBLEU. Les itérations avec l'étiquette -1 étant plus nombreuses nous en avons pour l'apprentissage supprimé aléatoirement de sorte qu'il y ait autant d'exemples pour les deux classes. Pour l'apprentissage du classifieur nous avons utilisé 2 corpus (français→anglais et anglais→français) distincts des 2 corpus d'évaluation utilisés jusqu'ici dans nos expériences, mais optimiser sur le même corpus de développement et avec les mêmes données d'apprentissage. Nos données d'apprentissage sont composées de 14 060 exemples correspondant chacun à une itération effectuée par notre décodeur.

	français→anglais		
Configuration	Taux d'erreurs	Rappel	Précision
[1]	49.91%	24.62%	73.46%
[2]	39.20%	22.22%	26.51%
[3]	39.20%	22.22%	26.51%

TAB. 20 – Performances du classifieur pour le sens de traduction français→anglais

	anglais→français		
Configuration	Taux d'erreurs	Rappel	Précision
[1]	54.00%	18.89%	66.93%
[2]	42.43%	36.59%	36.29%
[3]	51.62%	78.05%	36.92%

TAB. 21 – Performances du classifieur pour le sens de traduction anglais→français

Les résultats de notre classifieur sont présentés dans les Tables 20 et 21. Les configurations utilisées sont définies comme suit :

1. l'apprentissage utilise tous les exemples (14 060) et la classification se fait sur toutes les itérations du corpus d'évaluation ;
2. l'apprentissage utilise tous les exemples (14 060) et la classification se fait uniquement sur les itérations finales du corpus d'évaluation ;
3. l'apprentissage utilise uniquement les itérations finales de la recherche locale comme exemples (1 056) et la classification se fait uniquement sur les itérations finales du corpus d'évaluation.

On constate que nos résultats restent approximativement tous au-dessus des 40% d'erreurs, ce qui ne permet pas une utilisation fiable de ce classifieur. Avec un tel taux d'erreur un classement au hasard des hypothèses serait tout aussi performant. Ainsi, avec les traits que nous avons choisis, un classifieur de type SVM ne peut pas faire la distinction entre une amélioration ou une dégradation effectuée par la recherche locale. Pour améliorer cette situation, il faudrait vraisemblablement développer de nouveaux traits plus spécifiques aux transformations effectuées par la recherche locale, qui plus précisément caractérisent les segments touchés par la recherche locale.

5 Meilleur parcours de l'espace de recherche

Nos premiers résultats montrent d'assez nettes dégradations des traductions par la recherche locale. Maximiser la fonction de score du décodeur ne permet donc pas d'obtenir de meilleures traductions par recherche locale et met en évidence la nécessité de mieux parcourir l'espace de recherche pour obtenir des améliorations. Nous proposons donc d'ajouter de nouveaux modèles à cette fonction de score dans le but de mieux l'informer et de la rendre plus efficace pour guider la recherche locale (section 5.1). De plus les dégradations de la recherche locale nous indique que des zones pourtant correctes de la traduction sont modifiées. Afin que la recherche locale se concentre en priorité sur les zones incorrectes de la traduction, nous présentons la mise en place d'une mesure de confiance au niveau des segments qui identifierait les zones de la traduction à ne pas transformer (section 5.2). Nous introduisons en outre un nouvel opérateur capable d'agrandir notre espace de recherche en transformant la phrase source pour nous permettre d'atteindre de nouvelles traductions (section 5.3).

5.1 Introduction de nouveaux modèles

Nous avons vu que la recherche locale permettait la mise en place de modèles pouvant évaluer des hypothèses complètes. Ainsi nous avons donc étudié l'introduction de nouveaux modèles pour mieux évaluer, en particulier, la grammaticalité d'une hypothèse.
Deux nouveaux modèles que nous avons introduits reposent sur les catégories morpho-syntaxiques (*POS*, pour *part-of-speech*) et la forme lemmatisée des mots de l'hypothèse. Pour extraire les lemmes et les POS d'une hypothèse nous avons utilisé le parseur robuste XIP [Ait-Mokhtar et al., 2002].
XIP permet aussi de décomposer une hypothèse en *chunks* sous la forme d'un arbre. L'arbre créé reflète la capacité du parseur à relier les différents chunks entre eux. Plus on aura de chunks de premier niveau (nommés *flc* pour *first level chunk*), plus cela signifiera que XIP a eu des difficultés à relier les chunks entre eux. Intuitivement si nous avons un grand nombre de chunks de premier niveau, c'est donc que l'hypothèse est probablement mal-formée. Nous avons exploité cette information donnée par XIP pour d'une part avoir un nouveau modèle reposant sur la séquence de chunks de premier niveau de l'hypothèse et d'autre part calculer un score sur l'hypothèse nommé *flc-ratio*. Le *flc-ratio* est calculé par la formule suivante :

$$\text{flc-ratio} = \frac{\text{nombre de mots de l'hypothèse}}{\text{nombre de chunks de premier niveau}} \tag{13}$$

La Table 22 présente des exemples d'analyses *flc* effectuées par XIP. On peut voir qu'entre la première et la deuxième hypothèse, le mot "puis" a été retiré rendant la deuxième hypothèse grammaticalement incorrecte. XIP pour cette phrase choisi donc de rajouter un chunk de premier niveau ce qui coïncide avec notre intuition selon laquelle plus on aura de chunks de premier niveau, et plus l'hypothèse sera potentiellement agrammaticale[19]. De plus, comme le montre l'exemple présenté dans la Table 23, XIP sera limité dans sa capacité à réduire le nombre de *flc* même sur les zones correctes d'une hypothèse. En effet, le segment "les avocats américains" est ici décomposé en deux chunks, un chunk nominal ("NP") et un chunk adjectival ("AP"). Si l'on conjugue correctement le verbe "enquêter", ces deux chunks sont fusionnés en un "SC", le raccourcissement de la séquence des *flc* témoigne bien ici d'une amélioration de la grammaticalité.

Hypothèse	les	enfants	ont	mangé	puis	ils	sont	partis	.
flc	SC					SC			SENT
Hypothèse	les	enfants	ont	mangé		ils	sont	partis	.
flc	SC				NP			FV	SENT

TAB. 22 – Exemples de séquences *flc*

Nous avons également utilisé l'analyseur robuste en entités nommées NCA (Non-Contextual Analysis)[20] pour extraire des informations sémantiques d'une hypothèse. Ces informations sont données sous la forme d'une ou plusieurs étiquettes associées à chaque mot ou groupe de mots de l'hypothèse. Les informations fournies par les différents modèles sont illustrées Table 23.

Hypothèse	les	avocats	américains	enquêter	les	affaires	.
POS	DET	NOUN	ADJ	VERB	DET	NOUN	SENT
lemmes	le	avocat	américain	enquêter	le	affaire	.
nca	det	NN	subs orig	action	det	subs	punc
flc	NP		AP	IV	NP		SENT
flc-ratio	1.4						

TAB. 23 – Étiquettes et score donnés par les nouveaux modèles

Les nouveaux modèles *POS*, *lemmes*, *flc* et *nca* sont appris et calculés pour une hypothèse de la même manière que le modèle de langue présenté dans la section 2.4, les mots étant simplement remplacés par

[19] Il est important de noter que cette hypothèse ne tient que parce que nous ne comparons entre elles que des hypothèses produites à partir de la même phrase source.
[20] Cet analyseur est décrit dans Rosset et al. [2008] et utilise le système `wmatch` [Galibert, 2009]

les étiquettes des différents modèles. Ces modèles ont été estimés sur les mêmes données d'apprentissage que le système de traduction. Ainsi, lors du calcul des scores donnés par les différents modèles, les n-grammes d'un type particulier observés pendant l'apprentissage seront favorisés. Le vocabulaire des modèles *POS*, *flc* et *nca* étant constitués d'un petit nombre d'étiquettes, il a été possible d'estimer des modèles 6-grammes avec un lissage de type Witton-Bell. Un modèle 4-grammes a lui été estimé pour le modèle de *lemmes* avec un lissage de type Knesser-Ney.
Nous avons donc pu enrichir notre fonction de score par ces nouveaux modèles :

$$\begin{aligned} score(f,e) &= \lambda_{lm} \log P_{lm}(e) + \sum_i \lambda_{tm}^{(i)} \log p_{tm}^{(i)}(e|f) - \lambda_d p_d(f,e) - p_w \\ &+ \lambda_{pos} P_{pos}(e) \\ &+ \lambda_{lem} P_{lem}(e) \\ &+ \lambda_{nca} P_{nca}(e) \\ &+ \lambda_{flc} P_{flc}(e) \\ &+ \lambda_{\text{flc-ratio}} \text{flc-ratio} \end{aligned} \tag{14}$$

La procédure d'optimisation MERT a été utilisée pour recalibrer les poids λ associés à chaque modèle. Pour cela nous sommes repartis de la liste des 200 meilleures hypothèses de notre corpus de développement et avons recalculé les scores de chaque modèle pour toutes les hypothèses. MERT a ensuite pu estimer les nouveaux poids en utilisant la liste des 200 meilleures hypothèses enrichies des nouveaux scores. Les nouveaux poids obtenus par MERT ont ensuite été appliqués à nos corpus d'évaluation : les scores des 1 000 hypothèses de traduction pour chaque phrase ont été recalculés avec ces nouveaux poids, puis les hypothèses ont été reclassées en fonction de leur nouveau score. On obtient ainsi une nouvelle meilleure traduction selon notre fonction de score. Avec nos nouveaux modèles, MERT doit au moins donner un score BLEU pour notre corpus de développement équivalent à celui obtenu sans nos nouveaux modèles. Si MERT ne parvient pas à utiliser les nouveaux modèles, leurs poids seront déterminés de sorte qu'ils n'affectent pas la performance des autres modèles. Si MERT obtient une amélioration significative du score BLEU pour la traduction du corpus de développement, le reclassement de la liste des 1 000 meilleures hypothèses de notre corpus d'évaluation donnera lui aussi, le plus souvent, une amélioration du score BLEU pour la traduction du corpus d'évaluation. En effet, pour deux tâches de traduction très proches à l'aide d'un même système, on s'attend à ce qu'une augmentation du score BLEU pour l'une (sur le corpus de développement) entraîne une augmentation du score BLEU sur l'autre (corpus d'évaluation). Il faut cependant noter que nous ne nous sommes pas mis dans la situation optimale où le corpus de développement et le corpus d'évaluation proviennent de la même langue d'origine (cf. section 4.2). Nous avons également testé nos nouveaux modèles en appliquant MERT directement sur nos corpus d'évaluation (*self-tuning*), de la même manière que pour notre corpus de développement. Nous nous sommes mis ici dans la situation où l'adéquation entre corpus de développement et corpus d'évaluation serait parfaite.

	français→anglais					
	Développement		Évaluation		self-tuning	
Modèles additionnels activés	BLEU	TER	BLEU	TER	BLEU	TER
baseline	28.29	54.94	47.95	32.13	50.29	31.04
POS	28.97	54.31	46.14	33.37	50.29	31.24
lemmes	28.93	54.73	45.32	33.88	50.31	31.06
nca	28.98	54.36	45.89	33.56	50.28	31.24
flc	29.05	54.20	45.69	33.73	50.33	31.00
flc-ratio	28.98	54.28	45.75	33.59	50.29	31.12
Tous	29.10	54.54	45.63	33.87	50.34	31.02

TAB. 24 – Résultats obtenus après optimisation par MERT après avoir ajouté nos nouveaux modèles pour le sens de traduction français→anglais

Nos résultats présentés dans les Tables 24 et 25 indiquent que MERT a pu utiliser nos nouveaux modèles pour améliorer la traduction de notre corpus de développement. Avec l'ensemble de nos nouveaux modèles

	anglais→français					
	Développement		Évaluation		self-tuning	
Modèles additionnels activés	BLEU	TER	BLEU	TER	BLEU	TER
baseline	27.88	57.39	23.11	60.31	24.27	59.43
POS	27.91	56.99	23.31	59.87	24.26	59.40
lemmes	28.13	57.10	23.29	59.93	24.32	59.44
nca	28.03	57.06	23.27	59.72	24.24	59.42
flc	28.16	57.14	23.14	59.81	24.37	59.36
flc-ratio	28.11	56.94	23.43	59.52	24.30	59.30
Tous	28.23	57.07	23.13	60.20	24.52	59.33

TAB. 25 – Résultats obtenus après optimisation par MERT après avoir ajouté nos nouveaux modèles pour le sens de traduction anglais→français

nous obtenons des améliorations de 0,81 et 0,35 point BLEU sur nos corpus de développement. Nous ne retrouvons pas ces améliorations dans l'expérience en *self-tuning* pour laquelle l'ajout de nos nouveaux modèles n'a aucun impact sur le score des métriques BLEU et TER. Cette absence d'améliorations peut notamment s'expliquer par le fait que notre corpus d'évaluation ne contient que des documents dont la langue d'origine est aussi la langue depuis laquelle ont traduit, ce qui n'est pas le cas pour notre corpus de développement.
En revanche, l'utilisation des nouveaux poids donnés par MERT ne permet pas d'améliorer les scores BLEU et TER de façon significative sur nos corpus d'évaluation ; nous perdons même plus de 2 points BLEU pour notre de tâche de traduction français→anglais. Plusieurs raisons peuvent expliquer cette situation. Il est possible par exemple que les poids trouvés par MERT lors du décodage par `moses` du corpus de développement soit, par chance, l'un des ensembles de poids qui permettent une amélioration de la traduction du corpus de développement et une grande amélioration de la traduction du corpus d'évaluation. Chaque exécution de MERT atteint approximativement le même score BLEU pour la traduction du corpus de développement, en revanche les poids obtenus sont parfois très différents d'une exécution à l'autre. Ainsi, un ensemble de poids donnés peut, par hasard, mieux convenir au corpus d'évaluation. De cette façon, à moins de retrouver des poids optimaux pour notre corpus d'évaluation lors de notre exécution de MERT avec ajout de nos nouveaux modèles, retrouver le score équivalent ou supérieur à 47,95 points BLEU peut être difficile sans pour autant que nos modèles soient à mettre en cause. En outre, comme le montre la Table 26, nous obtenons assez peu de variété dans les scores de nos nouveaux modèles. Par exemple, les scores de *flc* et *flc-ratio* sont identiques pour de nombreuses hypothèses. Ce type de situation ne permet pas à MERT d'optimiser efficacement les poids de ces modèles.

Toutefois, hormis cette situation particulière qui donne une baisse du score BLEU pour notre corpus d'évaluation dans le sens de traduction français→anglais, nos nouveaux modèles, et plus particulièrement leur combinaison, permettent d'obtenir des améliorations des traductions sur nos corpus de développement et d'évaluation (pour notre tâche de traduction anglais→français). Ils peuvent donc être utilisés pour reclasser des hypothèses et obtenir une traduction améliorée. Le score de nos nouveaux modèles étant calculé sur la base d'hypothèses complètes (i.e. déjà construites), ces modèles se prêtent particulièrement bien à une utilisation par la recherche locale pour détecter de meilleures traductions dans le voisinage d'une hypothèse.

5.2 Réécriture des zones de faible confiance

Nos expériences oracle de la section 4.4 ont montré que si, en moyenne, la recherche locale dégrade les traductions, elle peut néanmoins passer par des étapes intermédiaires qui correspondent à des améliorations. Afin d'empêcher autant que possible la recherche locale d'effectuer des dégradations, nous avons mené de nouvelles expériences oracle dans lesquelles les segments de la meilleure hypothèse qui apparaissent dans la traduction de référence ne peuvent pas être modifiés. Ces segments, que nous appelons *segments de confiance*, ne peuvent donc pas subir une opération de la recherche locale, à l'exception d'un déplacement par l'opération `move`. Cela limite donc la recherche locale à tenter d'améliorer les fragments d'une hypothèse qui ne correspondent pas exactement à la traduction de référence attendue.

Obama was rather reticent at this price , at the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-8.89872	-24.813	-11.333	-4.93512	2.0
Obama was rather reluctant to price in the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-6.88889	-21.408	-8.93431	-4.00455	2.0
Obama was rather reluctant at that price at the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-6.45507	-24.5594	-8.61643	-3.25423	2.2
Obama was rather reluctant to that price at the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-6.45507	-23.5123	-8.61643	-3.25423	2.2
Obama was rather reluctant at this price in the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-6.45507	-25.366	-8.61643	-3.25423	2.2
Obama was rather reticent at this price at the beginning .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-6.45507	-23.7268	-8.61643	-3.25423	2.2
Obama was rather reluctant at that price .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-5.40551	-20.0475	-7.49903	-2.95741	2.0
Obama was rather reluctant to that price .				
POS	**lemmes**	**nca**	**flc**	**flc-ratio**
-5.40551	-19.0004	-7.49903	-2.95741	2.0

TAB. 26 – Extrait d'une liste des meilleures hypothèses pour une phrase décodée par `moses` enrichies des scores donnés par les nouveaux modèles.

Les Tables 27 et 28 présentent les résultats de nos expériences oracle avec figement des segments de confiance dans la configuration `move+replace+merge+split`. Dans ces expériences, les segments apparaissant dans la référence et contenant au moins n mots sont figés. De plus, les nouveaux segments de confiance créés par la recherche locale sont figés à leur tour à chaque itération. Si une séquence de segments correspond exactement à une suite d'au moins n mots présente dans la traduction de référence, ces segments de l'hypothèse sont liés entre eux de sorte qu'ils ne puissent pas être séparés par une opération `move` ultérieure et ne puissent plus être modifiés par l'une des trois autres opérations.

Dans l'exemple présenté Figure 13, on peut constater le figement de deux segments présents dans la traduction de référence : "ils peuvent être" et "pour des raisons politiques .". Ces deux segments ne peuvent être que déplacés et ne seront pas modifiables de toute autre manière par la recherche locale.

		français→anglais					
		Scores		Précision n-gram de BLEU			
Configuration	Tokens figés	BLEU	TER	1	2	3	4
baseline	-	47.95	32.13	0.7617	0.6387	0.5505	0.4795
n≥1	79.10%	49.94	31.09	0.7696	0.6534	0.5684	0.4994
n≥2	28.29%	48.78	32.03	0.7683	0.6475	0.5591	0.4878
n≥3	8.03%	47.37	32.83	0.7618	0.6376	0.5469	0.4737
n≥4	3.02%	46.41	33.33	0.7585	0.6314	0.5386	0.4641
n=∞	-	45.91	33.63	0.7563	0.6280	0.5343	0.4591

TAB. 27 – Résultats des expériences oracle avec figement des segments de confiance dans la configuration `move+replace+merge+split` pour le sens de traduction français→anglais. n représente le nombre de tokens minimal qu'un segment ou une séquence de segments contigus doit contenir pour être figé.

Source	they can be dismissed at any time , but may not be removed for political reasons .
Référence	il est certain qu' ils peuvent être licenciés à n' importe quel moment , mais ne peuvent l' être pour des raisons politiques .

Amorce						
they can be	dismissed	at any time ,	but	may not be	removed	for political reasons .
ils peuvent être	renvoyés	à tout moment ,	mais	ne peuvent pas être	supprimées	pour des raisons politiques .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-59.524818014516974	-20.5984	-17.4319	-31.6126	6.9993	19	0.3348

split : position 2, <à tout moment ,> ---> <à tout moment> + <,>							
they can be	dismissed	at any time	,	but	may not be	removed	for political reasons .
ils peuvent être	renvoyés	à tout moment	,	mais	ne peuvent pas être	supprimées	pour des raisons politiques .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-59.524818014516974 =	-20.5984 -0	-17.5978 -0.1658	-31.6126 =	7.9992 +0.9999	19 =	0.6982

FIG. 13 – Exemple de segments figés pour $n \geq 3$ pour le sens de traduction anglais→français. Les segments figés apparaissent sur fond blanc.

		anglais→français					
		Scores		Précision n-gram de BLEU			
Configuration	Tokens figés	BLEU	TER	1	2	3	4
baseline	-	23.11	60.31	0.5679	0.4092	0.3056	0.2311
n≥1	46.81%	24.79	57.99	0.5824	0.4271	0.3232	0.2479
n≥2	11.15%	23.74	59.20	0.5773	0.4171	0.3123	0.2374
n≥3	3.75%	22.79	60.07	0.5704	0.4083	0.3029	0.2279
n≥4	0.84%	22.16	60.53	0.5671	0.4031	0.2968	0.2216
n=∞	-	21.57	60.83	0.5646	0.3994	0.2918	0.2157

TAB. 28 – Résultats des expériences oracle avec figement des segments de confiance dans la configuration `move+replace+merge+split` pour le sens de traduction anglais→français. n représente le nombre de tokens minimal qu'un segment ou une séquence de segments contigus doit contenir pour être figé.

Nos résultats (Tables 27 et 28) montrent qu'en contraignant la recherche locale à effectuer des opérations sur les zones de faible confiance, il est possible d'obtenir des améliorations des traductions pouvant aller jusqu'à 2 points BLEU. Cependant, pour n strictement supérieur à 2, la recherche locale dégrade globalement à nouveau les traductions. Ces expériences mettent aussi en évidence les similarités lexicales entre les amorces et les traductions de référence. En effet, dans le sens de traduction français→anglais, pour n supérieur ou égal à 1, 79,10% des tokens sont figés. Ainsi seulement 20% des tokens n'apparaissent pas dans les traductions de référence et pourront subir une opération de la recherche locale. En plus de commettre très peu d'erreurs la recherche locale se trouve également nettement accélérée. En effet, étant donné le figement d'un grand nombre de segments, le voisinage des hypothèses est considérablement réduit et ne contient plus les états correspondant aux opérations qui auraient pu être appliquées à ces segments.
Les Figures 14 et 15 présentent respectivement des figements de segments de confiance et une dégradation effectuée par la recherche locale qui a pu être évitée par le figement de ces segments. Dans l'exemple correspondant à un cas de dégradation, on peut voir que la recherche locale remplace "this plan" par "this" or "this plan" apparaît dans le référence. Le figement par notre expérience oracle montrée dans la Figure 14 nous permet d'observer que cette dégradation aurait pu être évitée.
On peut conclure de ces observations qu'une bonne mesure de confiance au niveau des segments peut donc permettre à la recherche locale d'être mieux guidée. Plusieurs améliorations de cette méthode de figement sont envisageables. L'implémentation des figements présentée dans cette section peut notamment

Source	" cela montre à quel point le marché comptait sur ce plan " , a -t-elle ajouté , craignant qu' un projet adopté dans plusieurs jours n' arrive " trop tard " .
Référence	" this shows how much the market relied on this plan , " she added , fearing that a draft adopted in several days time can be " too late " .

Amorce													
"	cela montre	à quel point le marché	comptait sur	ce plan	" , a -t-elle	ajouté ,	craignant qu'	un	projet adopté	dans	plusieurs jours	n' arrive	" trop tard " .
"	this shows	how the market	was counting on	this plan	, " she	added ,	fearing that	a	draft adopted	in	several days	was	" too late " .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-128.50059654322803	-59.9975	-20.5704	-38.8916	13.9985	30	-7.2917

Replace : position 12, <was> ---> <is>													
"	cela montre	à quel point le marché	comptait sur	ce plan	" , a -t-elle	ajouté ,	craignant qu'	un	projet adopté	dans	plusieurs jours	n' arrive	" trop tard " .
"	this shows	how the market	was counting on	this plan	, " she	added ,	fearing that	a	draft adopted	in	several days	is	" too late " .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-132.71409700489784 -4.2135	-59.2512 +0.7462	-19.0663 +1.5041	-37.0604 +1.8312	13.9985 =	30 =	-7.2076

split : position 3, <was counting on> ---> <had> + <on>													
"	cela montre	à quel point le marché	comptait	sur	ce plan	" , a -t-elle	ajouté ,	craignant qu'	un	projet adopté	dans	plusieurs jours	n' arrive	" trop tard " .
"	this shows	how the market	had	on	this plan	, " she	added ,	fearing that	a	draft adopted	in	several days	was	" too late " .

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-131.71753817665 +0.9966	-60.2283 -0.9771	-20.2625 -1.1962	-32.1427 +4.9177	14.9984 +0.9999	29 -1	-7.1398

FIG. 14 – Exemple de segments figés pour $n \geq 2$ pour le sens de traduction français→anglais. Les segments figés apparaissent sur fond blanc.

Replace : position 4, <this plan> ---> <this>														
"	cela montre	à quel point le marché	comptait sur	ce plan	" , a -t-elle	ajouté	,	craignant qu'	un	projet adopté	dans	plusieurs jours	n' arrive	" trop tard "	.
"	this shows	how the market	was counting on	this	, " she	added	,	fearing that	a	draft adopted	in	several days	was	" too late "	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-120.94489381907736 +7.5557	-67.5444 -7.5469	-22.0725 -1.2408	-37.88 +1.0116	15.9983 =	29 -1	-6.7176

FIG. 15 – Exemple d'une dégradation évitable par figement des segments de confiance

amener notre décodeur à figer un même n-gramme apparaissant plusieurs fois dans l'hypothèse même si ce n-gramme n'apparaît qu'une seule fois dans la traduction de référence. De cette façon, si celle-ci contient par exemple une seule fois le bigramme "the house" et qu'il apparaît deux fois dans l'hypothèse, ces 2 occurrences seront figées alors qu'il est probable qu'une seule ne soit valide.

En outre, le n-gramme figé doit correspondre exactement à un segment ou une séquence de segments contigus, ce qui n'est pas optimal. Une méthode plus appropriée serait donc d'essayer de figer des n-grammes pouvant chevaucher plusieurs segments. Par exemple, si dans l'hypothèse de traduction nous avons deux segments "he is in" et "the kitchen ." et que le 4-gramme "in the kitchen ." est dans la traduction de référence, nous souhaiterions pouvoir figer ce 4-gramme (correct) afin que la recherche locale ne le modifie pas. Une solution possible serait d'effectuer une opération `split` aux frontières des segments. Dans notre exemple cela nous permettrait d'extraire le mot "in" du segment "he is in" et avec un `split` nous aurions donc au total trois segments : "he is", "in" et "the kitchen .". Cette technique requiert cependant que les segments "he is" et "in" soient dans la table de traduction pour que l'opération `split` soit rendue possible. Notre implémentation des figements deviendrait alors capable de figer "in" et "the kitchen .".

Nous souhaiterons par la suite pouvoir repérer automatiquement ces segments de confiance, en s'inspirant de travaux sur des mesures de confiance *a posteriori* au niveau sous-phrastique [Ueffing and Ney, 2007, Gispert et al., 2012]. Le champ de recherche sur le repérage des erreurs dans les sorties de traduction, par exemple fondé sur des critères linguistiques [Xiong et al., 2010], propose lui aussi des pistes intéres-

santes. L'objectif sera de ne pas remettre en cause des fragments d'hypothèses relativement sûrs, et donc indirectement de s'appuyer sur eux pour guider l'amélioration des autres fragments. Ainsi, un segment qui contiendrait une grande proportion de mots reconnus comme incorrects serait à traiter en priorité durant la recherche locale.

Nous avons testé une simple mesure de confiance automatique, n'ayant pas accès à la traduction de référence, en prenant pour référence les travaux de Blackwood et al. [2010]. Cette mesure repose sur le modèle de langue que nous avons utilisé : un n-gramme de l'hypothèse reconnu par le modèle de langage sera figé et la recherche locale ne pourra plus le modifier. Comme pour notre expérience précédente, ce n-gramme dans l'hypothèse doit correspondre à un segment ou une séquence de segments contigus contenant au moins n tokens. Les résultats des Tables 29 et 30 montrent que la recherche locale dégrade ici nettement les traductions mais améliore tout de même les précisions unigramme et bigramme de BLEU pour $n \leq 3$ dans le sens de traduction français→anglais. Cette mesure de confiance très naïve ne permet pas une amélioration des traductions. Les amorces sont toutefois moins dégradées, le figement de certaines zones de la traduction minimise en effet le risque que la recherche locale ne dégrade trop les traductions en trasnformant ces zones.

		français→anglais							
		Scores		Précision n-gram de BLEU				oracle	
Configuration	Tokens figés	BLEU	TER	1	2	3	4	BLEU	TER
baseline	-	47.95	32.13	0.7617	0.6387	0.5505	0.4795	-	-
n≥2	58.88%	47.64	32.59	0.7670	0.6411	0.5495	0.4764	48.78	32.03
n≥3	14.49%	46.71	33.26	0.7630	0.6350	0.5417	0.4671	47.37	32.83
n≥4	1.86%	46.15	33.55	0.7587	0.6305	0.5367	0.4615	46.41	33.33
n=∞	-	45.91	61.15	0.7563	0.6280	0.5343	0.4591	-	-

TAB. 29 – Résultats des expériences utilisant le modèle de langue pour le figement des segments de confiance dans la configuration `move+replace+merge+split` pour le sens de traduction français→anglais. n représente le nombre de tokens minimal qu'un segment doit contenir pour être figé. Les colonnes oracle reprennent les scores obtenus par figement des zones de confiances de la référence.

		anglais→français							
		Scores		Précision n-gram de BLEU				oracle	
Configuration	Tokens figés	BLEU	TER	1	2	3	4	BLEU	TER
baseline	-	23.11	60.31	0.5679	0.4092	0.3056	0.2311	-	-
n≥2	61.05%	22.62	60.47	0.5675	0.4057	0.3010	0.2262	23.74	59.20
n≥3	17.12%	22.10	60.93	0.5636	0.4013	0.2959	0.2210	22.79	60.07
n≥4	4.44%	21.63	61.15	0.5626	0.3984	0.2919	0.2163	22.16	60.53
n=∞	-	21.57	60.83	0.5646	0.3994	0.2918	0.2157	-	-

TAB. 30 – Résultats des expériences utilisant le modèle de langue pour le figement des segments de confiance dans la configuration `move+replace+merge+split` pour le sens de traduction anglais→français. n représente le nombre de tokens minimal qu'un segment doit contenir pour être figé. Les colonnes oracle reprennent les scores obtenus par figement des zones de confiances de la référence.

Cependant, étant donné que de nombreux segments sont figés, la recherche locale trouve assez rapidement un maximum local à la fonction de score et s'arrête. L'ajout d'un nouvel opérateur (`paraphrase`) est une première réponse que nous proposons dans la section suivante pour agrandir l'espace de recherche en apportant de nouvelles solutions à la recherche locale.
Nous pouvons conclure que notre mesure de confiance automatique présentée ici, même si elle permet de diminuer les risques que la recherche locale produise de dégradations, n'est pas une mesure de confiance suffisamment fiable et informée. Ce type de mesures fait actuellement l'objet de nombreux travaux, et nous présentons dans la section 6.1 quelques pistes d'améliorations envisageables.

5.3 Un nouvel opérateur : `paraphrase`

Au cours de nos expériences précédentes nous avons pu voir que la recherche locale pouvait être limitée à un espace de recherche trop restreint. Dans le but d'agrandir cet espace nous avons ajouter un nouvel opérateur que nous avons appelé `paraphrase`. Avec cet opérateur, la recherche locale peut modifier la phrase source à traduire en remplaçant l'un de ses segments par un autre segment qu'on suppose de sens équivalent. Ce nouveau segment aura des traductions pouvant être différentes de celles du segment remplacé. La recherche locale aura donc accès à ces nouvelles traductions. Chaque opération `paraphrase` provoque ainsi 2 remplacements de segments : un remplacement dans la phrase source et un remplacement dans la phrase cible. Un exemple de ce que peut réaliser l'opérateur `paraphrase` est présenté dans la Figure 16.
Les segments permettant de paraphraser un segment source sont contenus dans une table de paraphrase. Ils sont trouvés et estimés par la technique du *pivot* [Bannard and Callison-Burch, 2005], de ce fait ils partagent au moins une traduction en commun avec le segment source d'origine. Dans cette table le segment source et le segment cible sont dans la même langue et sont accompagnés d'un score estimant la probabilité que le segment cible puisse remplacer le segment source. Pour la majorité des segments il existe une entrée où le segment source et le segment cible sont identiques, dans cette situation le segment source est remplacé par lui-même (nous avons appelé cette opération `paraphrase identité`). Celle-ci ne donne donc pas accès à de nouvelles traductions puisque le segment source reste le même, ce type de `paraphrase` peut donc être assimilé à un simple `replace`.

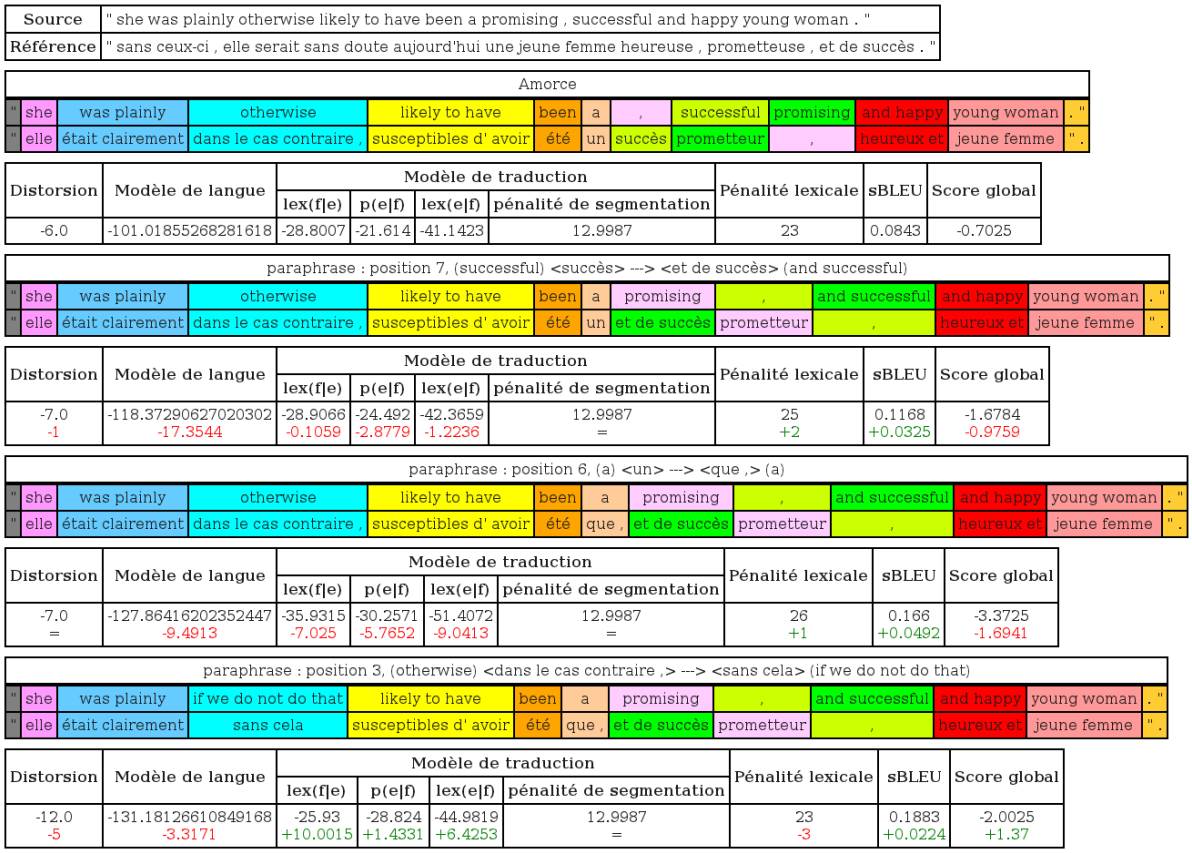

FIG. 16 – Exemples d'opérations `paraphrase`

La Figure 16 illustre différents types de paraphrases. La première itération remplace dans la phrase source "a" par "a" : il s'agit d'une `paraphrase identité`. Il aurait été possible d'atteindre la même traduction "que ," via un `replace`. L'itération suivante, en revanche, présente bien une opération `paraphrase` modifiant la phrase source : "otherwise" est remplacé par "if we do not do that" pour pouvoir atteindre la traduction "sans cela".

Notre table de traduction filtrée ne contenant que des segments source présents dans nos corpus, il nous a fallu ajouter dans la table les traductions de tous les segments source de la table de paraphrases, et leurs traductions en langue cible, pouvant venir remplacer l'un des segments source de la table de traduction. Cette démarche nous limite aux paraphrases que nous appelons *de premier niveau* : notre table de traduction ne contient pas (dans le cas général) les traductions des paraphrases de paraphrases. En conséquence, l'opérateur `paraphrase` ne pourra pas s'appliquer deux fois sur un même segment, sauf pour revenir au segment source d'origine.

Pour mesurer le potentiel de ce nouvel opérateur nous avons refait des expériences oracle maximisant sBLEU en suivant le même principe que les expériences présentées dans la section 4.4.4. Nos résultats sont présentés dans les Tables 31 et 32.

	français→anglais						
	Scores		Répartition des opérations effectuées				
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`	`paraphrase`
baseline	47.95	32.13	-	-	-	-	-
`replace`	66.73	19.44	-	100.00%	-	-	-
`paraphrase`	63.81	21.89	-	-	-	-	100.00%, identité : 55.19%
`move+replace +merge+split`	83.11	10.54	15.55%	28.52%	9.33%	46.60%	-
`move+replace +merge+split +paraphrase`	83.56	10.24	15.28%	8.16%	9.39%	46.02%	21.16%, identité : 55.16%

TAB. 31 – Résultats d'une recherche locale guidée par la métrique BLEU avec l'opérateur `paraphrase` pour le sens de traduction français→anglais

	anglais→français						
	Scores		Répartition des opérations effectuées				
Configuration	BLEU	TER	`move`	`replace`	`merge`	`split`	`paraphrase`
baseline	23.11	60.31	-	-	-	-	-
`replace`	44.46	37.88	-	100.00%	-	-	-
`paraphrase`	41.96	39.62	-	-	-	-	100.00%, identité : 71.15%
`move+replace +merge+split`	60.20	26.84	17.00%	51.86%	10.52%	20.61%	-
`move+replace +merge+split +paraphrase`	61.59	25.69	16.67%	6.36%	15.02%	29.82%	32.14%, identité : 72.66%

TAB. 32 – Résultats d'une recherche locale guidée par la métrique BLEU avec l'opérateur `paraphrase` pour le sens de traduction anglais→français

Ces résultats montrent que malgré un espace de recherche potentiellement plus grand l'opérateur `paraphrase` activé seul obtient un score BLEU moins élevé que `replace`. Dans toutes les configurations présentées ici, plus de la moitié des opérations `paraphrase` correspondent à des `paraphrase identité`.
Le score moins élevé de `paraphrase` par rapport à `replace` peut avoir plusieurs origines. Tout d'abord, la table de paraphrases ne contient pas pour chaque segment source la paraphrase identité. Dans cette situation, on se prive donc des traductions qu'aurait pu donner un simple `replace`. L'espace de recherche du `replace` n'est donc pas entièrement inclu dans celui de `paraphrase`. De plus, l'espace de recherche de `paraphrase` étant nettement plus grand que celui de `replace`, l'opérateur `paraphrase` maximise plus

vite le score sBLEU de l'hypothèse en trouvant plus rapidement les meilleures traductions de segments. Dans nos expériences, `replace` fait ainsi environ 200 itérations de plus que `paraphrase` et permet une exploration plus en profondeur de l'espace de recherche.

En revanche, comme le montrent nos résultats avec la configuration `move+replace+merge+split+paraphrase`, l'opérateur `paraphrase` peut apporter une contribution utile en association avec les autres opérateurs qui améliore la performance de l'oracle de 1,4 point BLEU. La recherche locale oracle trouve donc grâce à l'opérateur `paraphrase` de nouvelles solutions pour se rapprocher de la traduction de référence. L'opérateur `paraphrase` peut donc potentiellement améliorer les traductions produites par la recherche locale. Ce résultat est particulièrement intéressant, car il valide des hypothèses sur le potentiel de la traduction indirecte via des paraphrases [Onishi et al., 2010, Resnik et al., 2010].

6 Discussion et perspectives

6.1 Amélioration de la détection des zones de faible confiance

Au cours de nos expériences, nous avons vu qu'en disposant d'une mesure de confiance parfaite (section 5.2) la recherche locale réalisait correctement son travail d'amélioration en laissant intacts les n-grammes déjà reconnus comme corrects. En utilisant une mesure de confiance automatique mais peu informée reposant sur le figement des n-grammes de l'hypothèse reconnus par le modèle de langue, nous avons montré qu'il était possible de réduire les dégradations opérées par la recherche locale, sans néanmoins obtenir d'améliorations relativement à la traduction amorce.

Il est possible d'améliorer la mesure de confiance automatique utilisée. Cela peut se fonder, par exemple, sur l'identification automatique des erreurs présentes dans la traduction et leur catégorisation [Popović and Ney, 2011] pour ensuite sélectionner les opérations de recherche locale les plus appropriées (types et segments). Le travail de Bach et al. [2011] propose également une mesure de confiance reposant sur celles développées par Ueffing and Ney [2007] et Xiong et al. [2010], utilisant notamment des informations contenues dans la phrase source et les probabilités d'alignement entre segments source et cible. Leur mesure fournit des estimations de confiance au niveau des mots et de la phrase et pourrait donc mieux guider la recherche locale dans son travail de correction. En se basant sur une mesure de confiance donnée au niveau de la phrase, la recherche locale pourra en outre choisir des hypothèses qui auront une mesure de confiance dépassant un certain seuil en plus d'améliorer la fonction de score.

6.2 Opérations jointes et séquences d'opérations

Nous avons pu également constater que la recherche locale atteignait en quelques itérations un maximum de la fonction de score. Nos expériences montrent qu'en moyenne, sur nos 2 corpus d'évaluation, la recherche locale n'arrive plus à augmenter le score de l'hypothèse après 8 itérations, ce qui ne permet pas d'effectuer souvent des transformations importantes. Ce maximum n'est que local, d'autres hypothèses avec un score plus élevées sont formulables mais inatteignables avec l'ensemble des opérations et la stratégie *gloutonne* utilisés. Nous avons introduit pour cela l'opérateur `paraphrase`, qui a montré son potentiel lors d'expériences oracle (voir section 5.3). Dans leur travail, Langlais et al. [2007] utilise un opérateur appelé `bi-replace` qui effectue simultanément 2 remplacements de segments dans l'hypothèse. Cet opérateur permet de trouver de nouveaux états dans l'espace de recherche en agrandissant le voisinage de l'hypothèse calculé par la recherche locale.

Nous pouvons ainsi envisager d'autres opérations jointes telles que `replace+move`. Cet opérateur tenterait tout d'abord de remplacer un segment cible par un autre segment de la table de traduction et traduisant le même segment source, et serait déplacé dans toutes les positions possibles *via* un `move`. De nouvelles hypothèses, jusque-là inatteignables, seraient ainsi générées dans le voisinage mais au prix d'une très forte augmentation de la complexité de la recherche : l'opérateur `replace+move` aurait ainsi par exemple une complexité de $O(N^2 \times T)$, correspondant à N fois plus d'hypothèses qu'avec l'opération `replace` seule et donc des temps de calculs rédhibitoires. Une solution réside donc dans un élaguage du voisinage en ne calculant le score que des hypothèses les plus prometteuses. Cela peut se faire par

exemple en faisant en sorte que la recherche locale se concentre en priorité sur les segments appartenant à des zones de faible confiance (cf. sections 5.2 et 6.1). Une fois ce problème de complexité contourné, il devient possible d'essayer d'autres opérations jointes telles que `move+replace`, `move+move`, `merge+split`, `split+merge`, etc.

Cependant, même en utilisant des opérations jointes, il reste difficile d'atteindre les hypothèses de l'espace de recherche qui sont très éloignées de l'amorce. Typiquement, même en utilisant des méthodes d'élaguage du voisinage, la complexité d'opérations jointes combinant plus de 3 ou 4 opérations devient bien trop grande. L'utilisation d'une recherche par faisceau pendant la recherche locale, telle qu'essayée par Langlais et al. [2007], ne permet pas non plus de supprimer ce défaut (et elle n'avait, en outre, pas permis de montrer des gains dans nos expériences). Une façon prometteuse d'améliorer cette situation consisterait à réaliser de nombreux tirages pour déterminer quelles opérations mènent généralement à de meilleures hypothèses après application d'un certain nombre d'opérations. Une telle technique de type Monte Carlo a déjà été proposée et évaluée avec succès dans le cadre des systèmes de réécriture monolingue [Chevelu et al., 2009] ou pour la tâche de traduction elle-même [Arun et al., 2010].

6.3 Meilleure prise en compte du contexte

Une autre piste d'amélioration consisterait à exploiter des informations contextuelles extraites du document à traduire lors du décodage *a posteriori* d'une phrase de ce document. En effet, les systèmes de Traduction Automatique Statistique actuels restent très limités car ils réalisent leurs traductions phrase par phrase, ce qui rend difficile l'intégration d'informations contextuelles. Le travail récent de Hardmeier et al. [2012] explore ce type d'approche en utilisant un algorithme de recherche locale, et montre que des modèles sont capables de capturer des informations contextuelles utiles au décodage d'une phrase telles que la résolution des problèmes d'anaphore pronominale, de choix du temps à utiliser pour la conjugaison des verbes ou encore d'établir une certaine cohésion lexicale dans le décodage des phrases d'un même document. Ces problèmes sont très difficiles, voire impossibles à résoudre, en se limitant aux seules informations contenues au niveau de la phrase.

6.4 Diversification de l'espace de recherche

Toutes les expériences décrites dans ce mémoire ont utilisé comme source de réécriture pour les hypothèses de traduction la table de traduction d'un unique système, correspondant à celui ayant permis d'obtenir l'amorce. L'introduction de notre opération `paraphrase` avait notamment pour objectif d'atteindre de nouvelles hypothèses potentiellement très éloignées de l'hypothèse courante par l'effet de la traduction indirecte résultant de la réécriture d'un segment source. Il est également possible d'utiliser une table de traduction provenant de plusieurs systèmes. Un nouveau système peut ainsi être défini, correspondant dans les faits à un système de combinaison de systèmes (voir par exemple Matusov et al. [2009]). Les différentes hypothèses formulées par un nombre relativement important de systèmes peuvent constituer un corpus d'apprentissage pour un système de traduction statistique. La table de traduction obtenue peut alors soit servir à guider la recherche locale d'un système individuel particulier, soit à guider la recherche locale de la combinaison des systèmes. Nous menons actuellement de telles expériences à l'aide des données fournies pour la campagne d'évaluation WMT'11[21].

[21] `http://www.statmt.org/wmt11`

7 Conclusion

Ce travail a porté sur l'amélioration d'hypothèses de traduction automatique par une nouvelle recherche d'hypothèses. Dans ce mémoire, nous avons tout d'abord pu mettre en évidence les limites actuelles des systèmes de Traduction Automatique Statistique et plus particulièrement des modèles qu'ils utilisent. Nous avons pu montrer à l'aide d'expériences oracle qu'il était possible d'améliorer significativement les traductions produites. Pour tenter d'exploiter ce potentiel d'amélioration, nous avons eu recours à la recherche locale comme technique d'amélioration *a posteriori* des traductions déjà produites par un système. Cependant, comme le montrait déjà avant nous l'état de l'art en recherche locale, nous avons pu constater que dans sa formulation la plus simple cette technique ne permet pas d'améliorer la qualité de traduction au niveau d'un corpus, et peut même introduire de fortes dégradations. En analysant les résultats produits par nos expériences, nous avons montré que ces dégradations au niveau du corpus s'accompagnent toutefois d'améliorations sur environ un tiers des phrases de nos corpus d'évaluation, validant le potentiel de cette approche. De plus, notre travail a montré, à nouveau par l'intermédiaire d'expériences oracle, que la recherche locale guidée par une fonction de score très bien informée pouvait fortement améliorer une traduction. Nous avons donc essayé l'enrichissement de la fonction de score guidant la recherche locale en introduisant de nouveaux modèles calculés sur des hypothèses de traduction complètes. Un reclassement d'hypothèses à l'aide de la méthode d'optimisation MERT a permis de mettre en évidence que nos nouveaux modèles peuvent identifier de meilleures traductions, ce qui a suggéré leur utilisation durant la recherche locale.

Une autre piste que nous avons étudiée concerne les mesures de confiance calculées sur une hypothèse de traduction au niveau sous-phrastique. Nous avons vu qu'avec une mesure de confiance parfaite, permettant de figer les zones d'une hypothèse correspondant à la traduction de référence, la recherche locale pouvait améliorer une traduction par la modification de zones moins sûres. Nous avons essayé une mesure de confiance automatique très simple qui fige les n-grammes reconnus par le modèle de langue utilisé. Cette mesure, même si elle diminue effectivement les risques que la recherche locale fasse des erreurs, ne s'est pas montrée suffisamment efficace pour permettre une amélioration des hypothèses du système initial.

Enfin, afin d'agrandir l'espace de recherche parcouru par la recherche locale et d'éviter les maximums locaux rencontrés par la fonction de score, nous avons introduit un nouvel opérateur, `paraphrase`, capable de changer la phrase source pour atteindre de nouvelles hypothèses de traduction potentiellement plus distantes. Cet opérateur s'est révélé utile lors de nos expériences oracle, mais l'utilisation que nous en faisons pendant la recherche locale non informée ne nous a pas permis d'obtenir des améliorations jusqu'à présent.

Les recherches présentées dans ce travail débouchent sur des perspectives prometteuses, notamment en terme d'introduction d'informations contextuelles au niveau du document à traduire pour la production de traductions plus cohérentes, d'élargissement de l'espace de recherche *via* de nouveaux opérateurs et des techniques de type Monte Carlo, ou encore de développement de mesures de confiance susceptibles de guider la recherche locale pour que celle-ci s'attaque en priorité aux zones contenant des erreurs.

Références

S. Ait-Mokhtar, J.-P. Chanod, and C. Roux. Robustness beyond shallowness : incremental dependency parsing. *Natural Language Engineering*, 8(2–3) :121–144, 2002.

A. Allauzen and F. Yvon. Méthodes statistiques pour la traduction automatique. In E. Gaussier and F. Yvon, editors, *Modèles statistiques pour l'accès à l'information textuelle*, chapter 7, pages 271–356. Hermès, Paris, 2011.

A. Arun, B. Haddow, P. Koehn, A. Lopez, C. Dyer, and P. Blunsom. Monte carlo techniques for phrase-based translation. *Machine Translation*, 24(2) :103–121, 2010.

N. Bach, F. Huang, and Y. Al-Onaizan. Goodness : a method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies - Volume 1*, HLT '11, pages 211–219, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002500`.

C. J. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *ACL*, 2005.

G. Blackwood, A. de Gispert, and W. Byrne. Fluency constraints for minimum bayes-risk decoding of statistical machine translation lattices. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 71–79, Beijing, China, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1873781.1873790`.

C. Callison-Burch, C. Bannard, and J. Schroeder. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 255–262, Ann Arbor, USA, June 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P05-1032`.

S. Carter and C. Monz. Syntactic discriminative language model rerankers for statistical machine translation. *Machine Translation*, 25(4) :317–339, Dec. 2011. ISSN 0922-6567. URL `http://dx.doi.org/10.1007/s10590-011-9108-7`.

S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard Un iversity, 1998.

J. Chevelu, T. Lavergne, Y. Lepage, and T. Moudenc. Introduction of a new paraphrase generation tool based on monte-carlo sampling. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 249–252, 2009. URL `http://www.aclweb.org/anthology/P/P09/P09-2063.pdf`.

D. Déchelotte, G. Adda, A. Allauzen, H. Bonneau-Maynard, O. Galibert, J.-L. Gauvain, P. Langlais, and F. Yvon. Limsi's statistical translation systems for WMT'08. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 107–110, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0310`.

O. Galibert. *Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre interactif en domaine ouvert*. PhD thesis, Université Paris Sud, Orsay, 2009.

A. Gispert, G. Blackwood, G. Iglesias, and W. Byrne. N-gram posterior probability confidence measures for statistical machine translation : an empirical study. *Machine Translation*, pages 1–30, 2012. ISSN 0922-6567. URL `http://dx.doi.org/10.1007/s10590-012-9132-2`.

L. Gong, A. Max, and F. Yvon. Towards contextual adaptation for any-text translation. IWSLT '12, Hong Kong, 2012.

C. Hardmeier, J. Nivre, and J. Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea,

July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1108.

W. Hutchins and H. Somers. *An introduction to machine translation*. Academic Press, 1992. ISBN 9780123628305. URL http://books.google.fr/books?id=0ZhrAAAAIAAJ.

K. Knight. Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4) : 607–615, Dec. 1999. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=973226.973232.

P. Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *HLT-NAACL*, Edmonton, Canada, 2003.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses : Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P07-2045.

D. Kurokawa, C. Goutte, and P. Isabelle. Automatic detection of translated text and its impact on machine translation. Ottawa, Canada, 2009. MT Summit.

P. Langlais, F. Gotti, and A. Patry. A Greedy Decoder for Phrase-Based Statistical Machine Translation. In *11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, pages 104–113, Skovde, Sweden, 2007.

H.-S. Le, T. Lavergne, A. Allauzen, M. Apidianaki, L. Gong, A. Max, A. Sokolov, G. Wisniewski, and F. Yvon. Limsi @ wmt12. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 330–337, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3141.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 761–768, Sydney, Australia, 2006. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220175.1220271.

E. Matusov, G. Leusch, and H. Ney. *Learning To Combine Machine Translation Systems*, chapter 13, pages 257–276. Neural Information Processing Series. MIT Press, 2009.

P. P. Monty. Traduction statistique par recherche locale, 2010.

F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Sapporo, Japan, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1075096.1075117.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. A smorgasbord of features for statistical machine translation. In D. M. Susan Dumais and S. Roukos, editors, *HLT-NAACL 2004 : Main Proceedings*, pages 161–168, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

T. Onishi, M. Utiyama, and E. Sumita. Paraphrase lattice for statistical machine translation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 1–5, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-2001.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu : a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P02-1040.

M. Popović and H. Ney. Towards automatic error analysis of machine translation output. *Comput. Linguist.*, 37(4) :657–688, Dec. 2011. ISSN 0891-2017. URL http://dx.doi.org/10.1162/COLI_a_00072.

P. Resnik, O. Buzek, C. Hu, Y. Kronrod, A. Quinn, and B. B. Bederson. Improving translation via targeted paraphrasing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 127–137, Cambridge, MA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1013.

S. Rosset, O. Galibert, G. Bernard, E. Bilinski, and G. Adda. The limsi participation to the qast track. In *In Working Notes of CLEF 2008 Workshop*, 2008.

L. Schwartz, C. Callison-Burch, W. Schuler, and S. Wu. Incremental syntactic language models for phrase-based translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, pages 620–631, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1063.

M. Simard, C. Goutte, and P. Isabelle. Statistical phrase-based post-editing. In *Human Language Technologies 2007 : The Conference of the North American Chapter of the Association for Computational Linguistics ; Proceedings of the Main Conference*, pages 508–515, Rochester, New York, April 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N07/N07-1064.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, , and J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA)*, Cambridge, USA, 2006.

N. Tomeh. *Discriminative Alignment Models For Statistical Machine Translation*. PhD thesis, Université Paris Sud, Orsay, 2012.

N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1), 2007.

G. Wisniewski, A. Allauzen, and F. Yvon. Assessing phrase-based translation models with oracle decoding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 933–943, Cambridge, MA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1091.

D. Xiong, M. Zhang, and H. Li. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 604–611, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1062.

Annexes

A Exemples de recherche locale pour le sens de traduction anglais → français

Source	years of abuse by her stepfather led a woman to kill herself , a court has been told .
Référence	un tribunal a été informé que des années d' abus infligés par son beau-père ont mené une femme à se suicider .

Amorce											
years	of abuse	by	her	stepfather	led a	woman to	kill	herself ,	a court	has been told	.
des années	d' abus	par	son	stepfather	à la tête d' une	femme à	tuer	elle-même ,	un tribunal	a été informé	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-101.66534883543822	-25.3996	-18.9823	-37.7045	11.9988	23	-0.4421

split : position 10, <a été informé> ---> <a été> + <dit>												
years	of abuse	by	her	stepfather	led a	woman to	kill	herself ,	a court	has been	told	.
des années	d' abus	par	son	stepfather	à la tête d' une	femme à	tuer	elle-même ,	un tribunal	a été	dit	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-104.58042156316867 -2.9151	-24.7545 +0.6451	-19.846 -0.8637	-34.0852 +3.6193	12.9987 +0.9999	23 =	-0.0374

split : position 6, <femme à> ---> <femme> + <de>													
years	of abuse	by	her	stepfather	led a	woman	to	kill	herself ,	a court	has been	told	.
des années	d' abus	par	son	stepfather	à la tête d' une	femme	de	tuer	elle-même ,	un tribunal	a été	dit	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-103.59307307529282 +0.9873	-25.896 -1.1416	-20.6105 -0.7645	-34.2841 -0.1988	13.9985 +0.9999	23 =	0.3191

split : position 5, <à la tête d' une> ---> <,> + <une>														
years	of abuse	by	her	stepfather	led	a	woman	to	kill	herself ,	a court	has been	told	.
des années	d' abus	par	son	stepfather	,	une	femme	de	tuer	elle-même ,	un tribunal	a été	dit	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-96.59758930426761 +6.9955	-31.8467 -5.9507	-22.5981 -1.9877	-23.789 +10.4951	14.9984 +0.9999	20 -3	0.7708

Source	at the time , the department said that the attorneys had been fired for poor performance .
Référence	à l' époque , le département avait déclaré que les procureurs furent licenciés pour cause de prestations médiocres .

Amorce							
at the time ,	the department	said that	the attorneys	had been fired	for	poor performance	.
à l' époque ,	le département	a indiqué que	les avocats	avaient été lancées	pour	ses piètres performances	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-69.10725613752099	-24.3461	-13.8117	-41.8545	7.9992	19	-0.7611

split : position 6, <ses piètres performances> ---> <mauvais> + <résultats>								
at the time ,	the department	said that	the attorneys	had been fired	for	poor	performance	.
à l' époque ,	le département	a indiqué que	les avocats	avaient été lancées	pour	mauvais	résultats	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-64.54744687786489 +4.5598	-29.984 -5.6379	-16.8992 -3.0875	-33.8422 +8.0124	8.9991 +0.9999	18 -1	-0.0654

split : position 2, <a indiqué que> ---> <a dit> + <que>									
at the time ,	the department	said	that	the attorneys	had been fired	for	poor	performance	.
à l' époque ,	le département	a dit	que	les avocats	avaient été lancées	pour	mauvais	résultats	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-69.20695807204764 -4.6595	-28.9139 +1.0701	-14.0778 +2.8213	-30.4512 +3.3909	9.999 +0.9999	18 =	0.4274

split : position 0, <à l' époque ,> ---> <à l' époque> + <,>										
at the time	,	the department	said	that	the attorneys	had been fired	for	poor	performance	.
à l' époque	,	le département	a dit	que	les avocats	avaient été lancées	pour	mauvais	résultats	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-69.20695807204764 =	-28.9139 =	-14.735 -0.6572	-30.4512 =	10.9989 +0.9999	18 =	0.7598

B Exemples de recherche locale pour le sens de traduction français → anglais

Source	la Dre Décary garde toutefois en tête que " la confiance est quelque chose d' extrêmement fragile " .
Référence	Dr. Décary , nonetheless , thinks " confidence is something extremely fragile " .

Amorce										
la	Dre	Décary	garde toutefois	en tête que	"	la confiance est	quelque chose d'	extrêmement fragile	" .
the	Dre	Décary	however	in mind that	"	confidence is	something	extremely fragile	. "

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-102.4012550311591	-32.0442	-10.0277	-13.2472	9.999	15	-5.9759

split : position 9, <. "> ---> <"> + <.>										
la	Dre	Décary	garde toutefois	en tête que	"	la confiance est	quelque chose d'	extrêmement fragile	"	.
the	Dre	Décary	however	in mind that	"	confidence is	something	extremely fragile	"	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-104.54288942615287 -2.1416	-32.0442 =	-9.3256 +0.7021	-13.2472 =	10.9989 +0.9999	15 =	-5.7435

split : position 8, <extremely fragile> ---> <extremely> + <fragile>											
la	Dre	Décary	garde toutefois	en tête que	"	la confiance est	quelque chose d'	extrêmement	fragile	"	.
the	Dre	Décary	however	in mind that	"	confidence is	something	extremely	fragile	"	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-104.54288942615287 =	-32.0442 =	-9.8745 -0.5489	-13.2472 =	11.9988 +0.9999	15 =	-5.6106

merge : position 7, <something> + <extremely> ---> <extremely>										
la	Dre	Décary	garde toutefois	en tête que	"	la confiance est	quelque chose d' extrêmement	fragile	"	.
the	Dre	Décary	however	in mind that	"	confidence is	extremely	fragile	"	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-96.94159551716092 +7.6013	-43.0596 -11.0154	-9.2642 +0.6103	-12.9911 +0.2561	10.9989 -0.9999	14 -1	-5.396

Source	et surtout , il a reconquis la confiance du public qui avait été sérieusement ébranlée par le scandale du sang contaminé dans les années 1980 et 1990 .
Référence	and most importantly , it has regained the confidence of the public who had been seriously shaken by the tainted blood scandal in the 1980s and 1990s .

Amorce									
et surtout ,	il a	reconquis la	confiance	du public qui	avait	été sérieusement	ébranlée par le	scandale du sang contaminé	dans les années 1980 et 1990	.
and , above all ,	he has	regained the	confidence	of the public who	had	been seriously	damaged by the	contaminated blood scandal	in the 1980s and 1990s	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0	-85.94352807898417	-40.3844	-14.0314	-30.4304	10.9989	29	-2.19

split : position 1, <he has> ---> <it> + <has>										
et surtout ,	il	a	reconquis la	confiance	du public qui	avait	été sérieusement	ébranlée par le	scandale du sang contaminé	dans les années 1980 et 1990	.
and , above all ,	it	has	regained the	confidence	of the public who	had	been seriously	damaged by the	contaminated blood scandal	in the 1980s and 1990s	.

Distorsion	Modèle de langue	Modèle de traduction				Pénalité lexicale	Score global
		lex(f|e)	p(e|f)	lex(e|f)	pénalité de segmentation		
0.0 =	-86.7418343307252 -0.7983	-40.9987 -0.6143	-14.0024 +0.029	-28.5559 +1.8745	11.9988 +0.9999	29 =	-1.8864