Projet ANR 2009 CORD 023

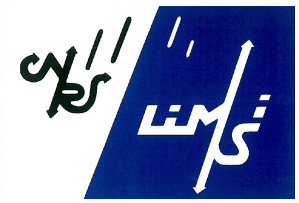
TRACE

TRADUCTION ROBUSTE PAR ANALYSE ET CORRECTION D'ERREURS

AMÉLIORATION EX-POST DE LA QUALITÉ DE TRADUCTION PAR RECHERCHE LOCALE

Juillet 2013

Benjamin Marie - Aurélien Max



Résumé

Ce rapport synthétise l'ensemble des travaux effectués, dans le cadre du projet TRACE, sur les méthodes de récriture de fragments d'énoncés *en langue cible* en vue d'améliorer la traduction. Ces travaux ont été conduits dans le cadre du lot 5.2.

Partant d'hypothèses de traduction produites par un système tiers, on peut exploiter de nouvelles informations difficiles d'accès au cours du décodage grâce à des techniques de recherche *a posteriori*, en particulier des méthodes de *recherche locale*. On espère, ce faisant, pouvoir corriger des erreurs de recherche qui auraient été faites lors du décodage. La stratégie utilisée est une stratégie *gloutonne* qui peut améliorer la meilleure hypothèse d'un système si et seulement si une hypothèse meilleure dans l'absolu, inédite et de meilleur score, est rencontrée durant l'exploration de l'espace des traductions possibles.

Des expériences oracle et de décodage, réalisées pour un ensemble de langues européennes, permettent de mieux évaluer le réel potentiel de ces techniques.

Amélioration ex-post de la qualité de traduction par recherche locale

Benjamin Marie Aurélien Max

Introduction

Dans ce rapport, nous décrivons nos tentatives pour améliorer la qualité d'une traduction en la modifiant *a posteriori*. En partant d'hypothèses de traduction déjà produites par un système tiers, on peut exploiter de nouvelles informations difficiles d'accès au cours du décodage grâce à des techniques de recherche *a posteriori*, par exemple la recherche locale [Langlais et al., 2007]. Avec une telle technique, on espère pouvoir corriger des erreurs de recherche qui auraient été faites lors d'un premier décodage. Néanmoins, cette approche se fonde sur le fait que la fonction de score utilisée pour guider la recherche modélise bien ce qu'est une bonne traduction.

Sous sa forme la plus simple, l'algorithme se décrit comme suit pour un système de traductio fondé sur les segments (*phrase-based*) : étant donnée une hypothèse de traduction *amorce*, un ensemble d'opérations de transformation est essayé sur l'ensemble des segments de l'hypothèse, chacune donnant lieu à la génération d'une hypothèse *voisine* de l'amorce. Le score de chacune de ces nouvelles hypothèses est calculé. Si le score de l'amorce est amélioré, la meilleure hypothèse sert à son tour d'amorce pour l'itération suivante, sinon la recherche d'une meilleure hypothèse s'arrête. Il s'agit donc d'une stratégie *gloutonne* qui peut améliorer la meilleure hypothèse d'un système si et seulement si une hypothèse meilleure dans l'absolu, inédite et de meilleur score, est rencontrée durant cette exploration de proche en proche de l'espace des traductions possibles.

Ce travail propose un regard complémentaire à ce qui est abordé ailleurs dans le Lot 4 de ce projet, où il est décrit comment proposer, avant toute traduction, des récritures possibles des phrases à traduire représentées sous forme de treillis. Nous nous intéressons donc ici à la possibilité de réviser une première hypothèse de traduction, mais considérerons néanmoins la possibilité de modifier également les phrases à traduire durant ce traitement.

Ce rapport est organisé comme suit : nous étudions dans un premier temps le *potentiel d'amélioration* de la recherche locale en calculant la meilleure hypothèse atteignable, compte tenu des opérations disponibles. Nous effectuons ensuite des expériences de décodage, en étudiant en particulier l'apport de fonctions caractéristiques complexes, et en évaluant également une stratégie mixte qui reposerait sur des mesures de confiance.

1 Évaluer le potentiel d'amélioration des traductions

1.1 Recherche locale *oracle*

Nous avons dans un premier temps réalisé des expériences *oracle*, afin de vérifier qu'une traduction est effectivement améliorable via une technique de recherche locale. Dans

cette partie du travail, nous mettons de côté la fonction de score initialement utilisée par le décodeur et utilisons à la place, pour guider la recherche, une mesure d'évaluation automatique comparant l'hypothèse de traduction produite à une traduction de référence. Pour guider la recherche locale nous avons utilisé la mesure sBLEU initialement utilisée dans [Liang et al., 2006], qui permet de rendre compte de la qualité d'une traduction au niveau de la phrase.

À chaque itération de la recherche, une opération améliorant le score sBLEU de l'hypothèse courante est appliquée. La recherche s'arrête lorsqu'il n'existe plus d'hypothèse ayant un meilleur score dans le voisinage, tel que décrit dans l'Algorithme 1.1. Cet algorithme peut être trivialement modifié pour permettre de conserver des hypothèses dans un *faisceau* (*beam search*), permettant une plus large exploration de l'espace de recherche.

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
```

Notre algorithme de recherche utilise les opérations suivantes (pour chaque opération, nous indiquons également le nombre de voisins évalués, en utilisant les notations suivantes : N correspond au nombre de bisegments, T au nombre maximum d'entrées par segment source dans la table de traduction, P au nombre maximum d'entrées par segment source dans une table de paraphrases locales, et S au nombre moyen de tokens par segment) :

1. `replace` ($\mathcal{O}(NT)$) : la traduction d'un segment source est remplacée par une traduction de la table de traduction ;
2. `split` ($\mathcal{O}(NST^2)$) : divise un segment source en deux segments contigus (présents dans la table de traduction), et utilise `replace` sur les segments produits ;
3. `merge` ($\mathcal{O}(NT)$) : fusionne deux segments source contigus si le segment résultant appartient à la table de traduction et utilise `replace` sur ce segment ;
4. `move` ($\mathcal{O}(N)$) : déplace le segment cible d'un bisegment vers toutes les positions inter-segments possibles dans l'hypothèse de traduction ;

5. `remove` ($\mathcal{O}(N)$) : supprime la traduction d'un bisegment de l'hypothèse de traduction (qui reste néanmoins disponible pour une récriture au cours des itérations suivantes) ;
6. `rewrite` ($\mathcal{O}(NP)$) : le segment source d'un bisegment est remplacé par un autre segment source, et sa traduction est remplacée par la traduction de ce nouveau segment source.

Un tel oracle fondé sur un algorithme de recherche nous permettra d'évaluer le potentiel d'amélioration des traductions, sans nous offrir toutefois la possibilité de toujours identifier la meilleure traduction pour un système donné. En effet l'algorithme employé ne peut pas réaliser une exploration complète de l'espace de recherche et s'arrêtera sur un maximum local de la fonction de score. De plus, du fait des limites de sBLEU dans sa capacité à modéliser une « bonne » traduction, une amélioration de sBLEU lors d'une itération de notre oracle ne signifie pas pour autant une amélioration de la qualité réelle de la traduction. Ainsi, par exemple, certaines suppressions ou déplacements de mots peuvent améliorer le score de traduction si ces mots séparaient deux séquences de mots qui sont contiguës dans la traduction de référence, alors que de telles transformations arbitraires sont susceptibles de rendre la phrase agrammaticale. Une solution possible serait d'utiliser une combinaison de plusieurs métriques d'évaluation au lieu d'une seule pour guider la recherche, ce qui présente l'inconvénient de faire augmenter les temps de calcul. Nous nous en sommes donc tenus à l'utilisation de sBLEU, en partant du principe qu'une forte amélioration du score correspond en effet à une amélioration *globale* effective de la qualité traduction.

La Figure 1 montre une trace d'exécution du programme de recherche locale *oracle* pour une traduction du français vers l'anglais. La recherche effectue tout d'abord une substitution de traduction (`replace`), qui permet de remplacer un seul token n'appartenant pas tel quel à la référence (*support*) par un trigramme appartenant à la référence (*will be supporting*). L'opération suivante sépare en deux (`split`) un segment source (*la ligne*), ce qui permet de faire apparaître une préposition manquante dans la traduction du second segment (*line of*[1]). La troisième opération consiste à supprimer (`remove`) la traduction d'un mot source ($donc_3$), qui dans cet exemple précis ne trouve effectivement pas de correspondant dans la traduction de référence. Finalement, une dernière opération de substitution de traduction (`replace`) permet d'obtenir l'article attendu pour la tête d'un groupe nominal (*the majority*).

La Figure 2 illustre les autres opérations sur différentes phrases. La première déplace (`move`) la traduction d'un segment, ce qui permet d'obtenir un plus grand nombre de n-grammes appartenant à la traduction de référence et de corriger localement la grammaticalité de la traduction (*a very long rail route*). La seconde récrit un segment source (`rewrite`) et permet d'atteindre une traduction précédemment non atteignable depuis l'état courant (*discussion on*[2]). La dernière opération correspond au regroupement de deux segments source contigus (`merge`), qui permet ici l'introduction d'un verbe non accessible précédemment (*permitted*[3]).

1. Bien entendu, on remarquera ici que cet alignement ne correspond pas à celui qu'aurait produit un annotateur humain, qui aurait plus naturellement associé *of the* comme la traduction du segment source du_6. Cela suggère qu'à scores oracles équivalents ou proches l'on préfère les hypothèses de traduction les mieux évaluées par le système automatique, lequel aurait ici préféré cette dernière solution.
2. On remarquera que l'opération de récriture des segments de la phrase n'introduit aucune contrainte de grammaticalité pour la phrase source récrite, comme le montre l'exemple obtenu : *il est évident que débat d'intermodalité...*
3. De nombreuses raisons peuvent être à l'origine de telles configurations où certains mots cibles ne peuvent être obtenus que par traduction d'un segment plus grand.

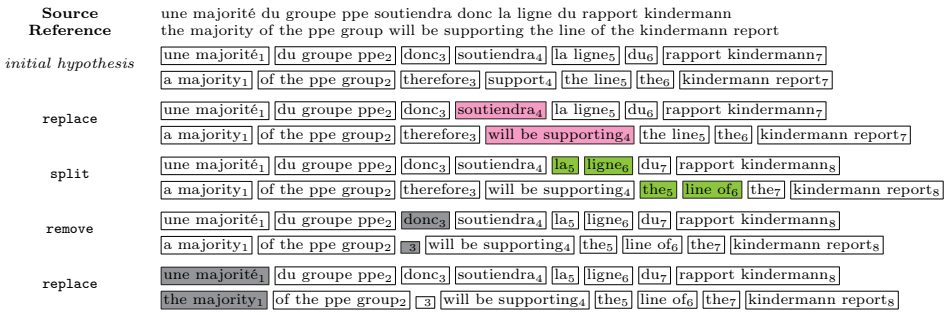


FIGURE 1 – Trace d'exécution du programme de recherche locale oracle pour une traduction du français vers l'anglais.

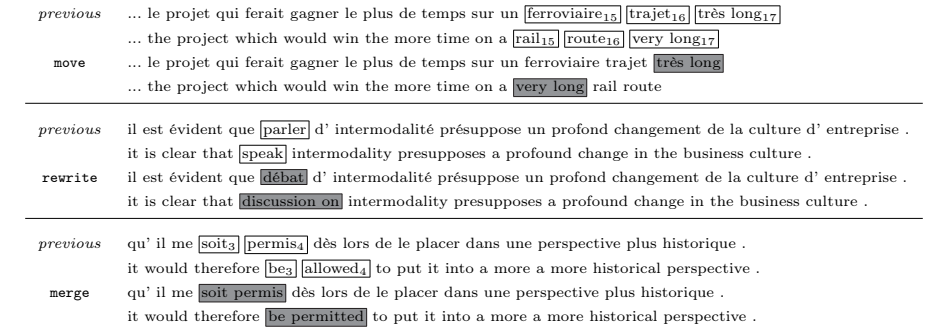


FIGURE 2 – Exemples d'applications d'opérations de récriture n'étant pas déjà illustrées sur la Figure 1.

1.2 Cadre expérimental

Nous avons utilisé le corpus Europarl [4] et calculé l'intersection pour 11 langues en utilisant l'anglais comme langue pivot. À partir de ces données nous avons extrait trois sous-corpus dédiés à l'apprentissage, au développement et à l'évaluation du décodeur. La composition de ces corpus est décrite dans le Tableau 1. Nous avons utilisé le français comme langue source et toutes les autres langues comme langues cibles, en étudiant plus en détails la traduction vers l'anglais. Pour ce sens de traduction, nous avons notamment souhaité observer l'impact individuel des opérations en les activant indépendamment les unes des autres. Pour voir dans quelle mesure le décodeur fait des erreurs de recherche, nous avons également fait varier la taille du faisceau de meilleures hypothèses considérées à chaque itération.

Nous avons entraîné un système de traduction état de l'art à l'aide de `moses` [5] [Koehn et al., 2007] en utilisant les paramètres et modèles standards ainsi que MERT [Och, 2003] pour l'optimisation du système sur le corpus de développement. Le modèle de langue a été estimé sur les trigrammes de la partie cible du corpus d'apprentissage avec un lissage Kneser-Ney [Chen and Goodman, 1998].

4. `http://statmt.org/europarl/`
5. `http://www.statmt.org/moses`

Langues	**apprentissage**	**développement**	**évaluation**		
	# M-tok.	# K-tok.	# K-tok.	BLEU	TER
fr	10,2	32,8	32,8	-	-
en	8,8	28,3	28,6	29,1	54,0
da	8,4	27,0	27,2	23,2	61,3
de	8,4	27,1	27,1	17,0	68,0
el	8,8	28,5	28,5	23,5	62,2
es	9,2	29,5	29,7	35,9	49,7
fi	6,4	20,6	20,5	11,2	79,7
it	10,2	28,9	29,0	31,6	55,3
nl	8,9	28,2	28,7	21,2	64,6
pt	9,1	29,4	29,3	33,4	52,8
sv	7,9	25,7	25,8	21,0	62,7

TABLE 1 – Statistiques sur la composition des corpus utilisés. Les scores BLEU et TER ont été calculés pour le décodage initial effectué par le décodeur `moses`.

Quelques modifications et adaptations ont été apportées à l'algorithme de recherche locale pour accélérer les décodages. En particulier, pour les opérations `replace`, `split` et `merge` nous n'avons considéré que les segments cible qui avaient au moins un token en commun avec la traduction de référence, à l'exception des 50 tokens les plus fréquents dans le corpus d'apprentissage.

Pour l'opération `rewrite`, les segments source candidats ont été extraits via la technique d'acquisition de paraphrase par pivot [Bannard and Callison-Burch, 2005] sur la table de traduction pour chaque paire de traduction considérée. Considérant la possibilité de paraphraser un segment par lui-même, une telle opération couvrirait alors un surensemble des hypothèses atteignables par l'opération `rewrite`. Nous avons préféré une formulation beaucoup plus limitée qui mettra mieux en évidence l'intérêt d'avoir recours à une telle *traduction indirecte* lorsque une traduction particulière ne peut être obtenue par une voie plus directe : les récritures possibles sont ainsi limitées aux traductions qui permettent de générer des n-grammes non atteignables *via* une application de `rewrite` et constitué d'au moins 6 caractères.

Enfin, la métrique sBLEU a été utilisée comme fonction objectif pour guider l'oracle, et les scores BLEU et TER ont été calculés pour évaluer la qualité des traductions avant et après décodage *oracle*.

1.3 Expériences

1.3.1 Opérations de récriture et taille de faisceau

Les détails des résultats pour les décodages *oracle* en traduction français vers anglais sont donnés dans le Tableau 2. On remarque que les deux opérations menant individuellement à la plus grande amélioration sont les opérations `replace` et `split`. `split` effectuant une segmentation puis un `replace` sur les deux nouveaux segments créés, l'oracle a donc, avec la seule opération `split`, directement accès à un plus grand nombre de traductions de segments que les autres opérations. En comparaison, `merge` mène à des gains beaucoup plus modestes mais reste tout de même utile lors de la recherche. `remove` est récompensé par sBLEU uniquement pour son impact sur la pénalité de concision de sBLEU et par le possible rassemblement de deux n-grammes joints dans la référence

	BLEU					TER	#. itérations
	score	1g	2g	3g	4g	score	**(moy. par phrase)**
baseline	29,0	63,2	35,5	22,6	14,6	54,0	-
taille du faisceau = 1							
`merge`	31,8	65,3	38,3	25,2	16,9	51,7	0,75
`move`	32,0	63,2	39,1	25,8	17,3	53,3	1,01
`rewrite`	29,8	64,5	36,2	23,0	14,0	53,5	0,38
`remove`	29,7	67,1	39,2	25,6	16,9	50,0	1,03
`replace`	42,1	73,9	48,8	34,8	25,1	42,5	4,40
`split`	45,7	74,3	52,7	39,1	28,6	41,3	4,46
Toutes	66,5	88,2	73,8	62,6	53,0	23,1	11,04
taille du faisceau = 2							
Toutes	66,6	88,1	73,9	62,8	53,2	23,0	11,19
taille du faisceau = 5							
Toutes	67,8	88,5	74,9	64,3	55,0	22,3	11,26

TABLE 2 – Effets individuels des opérations et de la taille du faisceau.

mais qui étaient séparés par le segment cible supprimé dans l'hypothèse. `move` mène à une amélioration modeste en permettant parfois la reconstitution de n-grammes présents dans la traduction de référence. Enfin, l'opération `rewrite` apporte des gains relativement faibles (+0,8 BLEU) mais les obtient avec très peu d'applications (0,38 récritures en moyenne par phrase).

En activant toutes les opérations, on obtient une amélioration de 37,5 points BLEU avec une moyenne de 11,04 itérations par phrase. Cette augmentation met clairement en évidence qu'il est possible d'améliorer de façon très importante une hypothèse de traduction construite par un système état de l'art, si l'on dispose, bien entendu, d'une fonction de score reflétant mieux la qualité réelle d'une hypothèse de traduction.

On constate par ailleurs qu'en augmentant la taille du faisceau à 5 on obtient un gain supplémentaire de +1,3 BLEU, ce qui met donc en évidence le fait que des erreurs de recherche sont commises au cours du décodage glouton. Cependant, l'augmentation de la taille du faisceau dégrade de façon importante le temps de calcul : en passant d'un faisceau de taille 1 à 5 la durée du décodage est par exemple multipliée par 3. L'ensemble des expériences décrites par la suite utiliseront un faisceau de taille 1.

1.3.2 Langues cibles

Nous avons aussi voulu évaluer le potentiel d'amélioration d'une traduction en fonction de la langue cible, en utilisant comme langues cibles certaines langues assez proches de la langue source (par exemple l'espagnol pour la traduction depuis le français) et des langues plus éloignées (par exemple le finnois). Nos résultats, pour les 10 langues cibles étudiées, sont présentés dans le Tableau 3.

On observe que pour des langues proches du français telles que l'espagnol et le portuguais l'amélioration relative du score BLEU est plus faible (+106% pour l'espagnol) que pour les langues plus éloignées (+311% pour le finnois). Pour le score TER, *a contrario*, l'amélioration relative est meilleure pour les langues proches du français (-63% pour l'espagnol, -52% pour le finnois). Cette différence vient probablement du fait que l'ordre des mots est similaire entre la traduction de référence et la traduction produite par l'oracle

		BLEU						TER	
		score		1g	2g	3g	4g	score	
da	*basel.*	23,2		57,2	28,9	17,3	10,7	61,3	
	oracle	58,4	↑+151%	84,0	66,5	53,8	43,6	29,5	↑-52%
de	*basel.*	17,0		52,6	22,1	11,6	6,3	68,0	
	oracle	55,1	↑+224%	83,7	64,2	50,1	39,0	32,0	↑-53%
el	*basel.*	23,5		53,8	28,9	17,8	11,1	62,2	
	oracle	62,8	↑+167%	85,2	70,3	58,8	49,3	26,5	↑-57%
en	*basel.*	29,0		63,2	35,5	22,6	14.7	54,0	
	oracle	66,5	↑+129%	88,2	73,8	62,6	53,0	23,1	↑-57%
es	*basel.*	35,9		65,3	41,5	29,4	21,2	49,7	
	oracle	74,0	↑+106%	90,6	79,8	70,3	62,5	18,2	↑-63%
fi	*basel.*	11,2		40,1	15,5	7,2	3,5	79,7	
	oracle	46,1	↑+311%	77,0	55,7	41,4	30,7	38,1	↑-52%
it	*basel.*	31,6		60,1	37,2	25,5	17,9	55,2	
	oracle	71,2	↑+125%	89,3	77,6	66,3	57,7	20,4	↑-63%
nl	*basel.*	21,2		56,4	26,8	15,4	9,5	64,6	
	oracle	56,3	↑+165%	83,7	64,8	51,5	41,0	32,4	↑-50%
pt	*basel.*	33,4		62,3	38,7	26,9	19,1	52,8	
	oracle	69,8	↑+109%	88,6	76,3	66,3	57,7	21,5	↑-59%
sv	*basel.*	21,0		55,0	26,3	15,2	9,2	62,7	
	oracle	59,9	↑+185%	85,0	67,7	55,3	45,2	27,8	↑-55%

TABLE 3 – Résultats de la recherche locale oracle en fonction de la langue cible, avec le français comme langue source.

pour les langues cible proches de la langue source.

La Figure 3 montre la distribution des opérations en fonction de la langue cible. On peut remarquer que `replace` est à peu près utilisé dans les mêmes proportions pour toutes les langues. En revanche `split` est beaucoup plus utilisé pour des langues cibles dont le système initial (*baseline*) est de meilleure qualité, ce qui pourrait être attribué au fait que pour de telles configurations une trop forte confiance est attribuée, à tort, pour les longs segments. De façon peu surprenante, nous pouvons observer une plus grande utilisation du `move` pour les langues cible ayant une syntaxe très différente du français, comme c'est le cas notamment de l'allemand et du néerlandais. Finalement, l'opération `remove` est beaucoup plus utilisée pour le finnois que pour d'autres langues, ce qui témoigne de la difficulté à tokeniser et aligner les langues à forte morphologie compositionnelle comme le finnois.

2 Fonction de score et recherche locale standard

Nous avons vu dans la section 1 qu'il était effectivement possible d'améliorer une traduction *a posteriori* si l'on disposait d'une fonction de score appropriée, ce qui est bien entendu au cœur des difficultés des recherches actuelles en Traduction Automatique (voir également Wisniewski and Yvon [2013]). Dans cette section, nous allons étudier les effets d'une recherche locale guidée par la fonction de score d'un système de traduction état de l'art, à la manière des travaux de Langlais et al. [2007] et proposer quelques pistes d'amélioration.

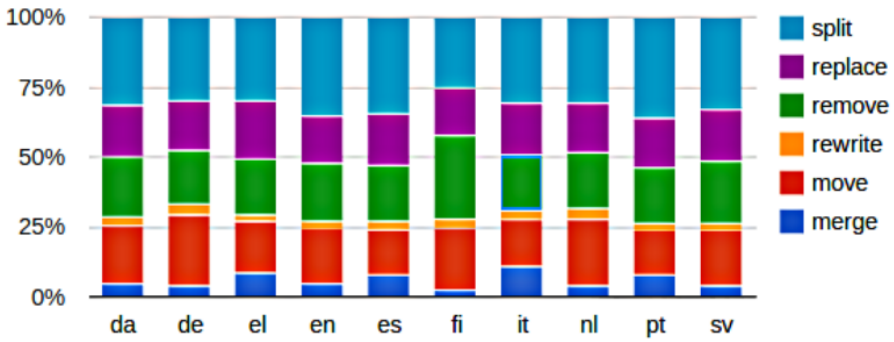


FIGURE 3 – Distribution des opérations en fonction de la langue cible avec le français comme langue source.

2.1 Cadre expérimental

Pour ces expériences, nous nous sommes concentrés 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 [6]. 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 laquelle les phrases en anglais à traduire ont été originellement écrites en anglais, et avons fait de même pour le français [7]. Ainsi, les corpus utilisés pour la traduction anglais → français et français → anglais sont, par construction, deux corpus différents. Le corpus de développement rassemble quant à lui des documents pour lesquels nous n'avons pas tenu compte de la langue d'origine, ce qui n'est pas un choix optimal [8]. Les données utilisées sont décrites dans les Tableaux 4 et 5.

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

TABLE 4 – Nombre de phrases et de tokens pour les corpus d'apprentissage, développement et évaluation pour notre tâche de traduction anglais → français.

Pour les pré-traitements, nous avons réutilisés ceux effectués pour les soumissions du LIMSI à la campagne WMT'12 [Le et al., 2012]. Les tables de traduction ont été estimées par échantillonnage aléatoire 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, Le et al., 2012], mais en un temps

6. `http://http://www.statmt.org/wmt12/`

7. 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 placés dans des conditions expérimentales plus interprétables, mais dans lesquelles les performances du système seront plus difficiles à améliorer. Les applications de récriture en présence de textes source traduits apparaissent comme une perspective intéressante au travail décrit ici.

8. Voir [Allauzen et al., 2013] pour une étude de l'impact de la prise en compte de la langue d'origine dès la phase d'optimisation des systèmes.

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

TABLE 5 – Nombre de phrases et de tokens pour les corpus d'apprentissage, développement et évaluation pour notre tâche de traduction français → anglais.

considérablement réduit. L'optimisation des modèles est réalisée par minimisation du taux d'erreur (MERT), et la construction des hypothèses de traduction utilise à nouveau le système état de l'art `moses`, dans la configuration fondée sur les segments. Le modèle de langue utilisé a été appris sur un très grand corpus de données (cf. [Le et al., 2012]) : 6 milliards de tokens pour l'anglais et 2,5 milliards de tokens pour le français. Il s'agit d'un modèle de langue 4-gramme avec lissage Kneser-Ney [Chen and Goodman, 1998].

2.2 Expériences de recherche locale

Dans le cadre de ces expériences, nous nous sommes limités aux opérations `move`, `replace`, `merge` et `split` telles qu'elles sont décrites dans la section 1.1. Les résultats que nous avons obtenus sont présentés dans les Tableaux 6 et 7.

	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)

TABLE 6 – 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)

TABLE 7 – Résultats de notre décodeur par recherche locale pour le sens de traduction anglais→français

Ces premiers résultats montrent une assez nette dégradation des traductions produites

par le système `moses`. En effet, la plupart des configurations que nous avons testée dégradent les scores BLEU et TER du système initial (*baseline*).[9] 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 pour lesquelles les scores sont les moins pénalisés sont celles pour lesquelles 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, il ressort de l'analyse des traces de nos expériences qu'un grand nombre d'opérations `split` effectuées ne changent pas la traduction (uniquement un changement de segmentation, donc). Cela signifie que la fonction de score préfère les traductions composées de petits segments, ce qui explique également le peu d'opérations `merge` observées.

2.3 Introduction de nouveaux modèles

Nos premiers résultats montrent d'assez nettes dégradations des traductions par la recherche locale avec un décodeur fondé sur les segment état de l'art, ce qui correspond aux observations précédemment décrites par Monty [2010]. Maximiser la fonction de score du décodeur ne permet donc pas d'obtenir de meilleures traductions par recherche locale[10] et met en évidence la nécessité de mieux parcourir l'espace de recherche pour obtenir des améliorations. Une solution intéressante consiste en l'ajout de nouveaux modèles dans la fonction de score afin de mieux l'informer sur la qualité des hypothèses et ainsi de la rendre plus efficace pour guider la recherche locale.

Les 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], disponible en anglais et en français. L'analyseur XIP permet également de décomposer une hypothèse en *chunks* sous la forme d'un arbre. L'arbre créé reflète la capacité de l'analyseur 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 donc exploité cette information donnée par XIP pour avoir un nouveau modèle reposant sur la séquence de chunks de premier niveau de l'hypothèse.

Nous avons par ailleurs utilisé l'analyseur robuste en entités typées NCA (Non-Contextual Analysis) décrit par Rosset et al. [2008] 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 nouveaux modèles *POS*, *lemmes*, *flc* et *nca* ont été estimés comme des modèles de langue sur les mêmes données d'apprentissage que le système de traduction. Ainsi, lors

9. Nous attribuons principalement cela aux différents types d'élagages de l'espace de recherche effectués par le décodeur utilisé.

10. Nous notons que le décodeur état de l'art utilisé dans les travaux de Langlais et al. [2007] permettait lui des gains dans cette situation, ce qui nous renseigne sur les progrès effectués à différents niveaux par les systèmes de traduction statistique en l'espace de 5 ans.

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 Witten-Bell. Un modèle 4-grammes a lui été estimé pour le modèle de *lemmes* avec un lissage de type Kneser-Ney.

La procédure d'optimisation MERT a été utilisée pour recalibrer les poids associés à chaque modèle. Pour cela nous sommes repartis de la liste des 200 meilleures hypothèses du 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 devrait elle aussi mener à 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). Nous pouvons cependant regretter le fait de ne pouvoir considérer la situation optimale dans laquelle le corpus de développement et le corpus d'évaluation proviendraient de la même langue d'origine.

	français→anglais			
	Développement		Évaluation	
Modèles additionnels activés	BLEU	TER	BLEU	TER
baseline	28,29	54,94	47,95	32,13
POS	28,97	54,31	46,14	33,37
lemmes	28,93	54,73	45,32	33,88
nca	28,98	54,36	45,89	33,56
flc	29,05	54,20	45.69	33,73
Tous	29,10	54,54	45.63	33,87

TABLE 8 – Résultats obtenus après optimisation par MERT après avoir ajouté nos nouveaux modèles pour le sens de traduction français→anglais.

Les résultats présentés dans les Tableaux 8 et 9 indiquent que MERT a pu utiliser nos nouveaux modèles pour améliorer la traduction du corpus de développement. Avec l'ensemble de nos nouveaux modèles nous obtenons des améliorations de 0,81 et 0,35 point BLEU sur ces 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 la 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 permettent, par chance, une forte amélioration

	anglais→français			
	Développement		Évaluation	
Modèles additionnels activés	BLEU	TER	BLEU	TER
baseline	27,88	57,39	23,11	60,31
POS	27,91	56,99	23,31	59,87
lemmes	28,13	57,10	23,29	59,93
nca	28,03	57,06	23,27	59,72
flc	28,16	57,14	23,14	59,81
Tous	28,23	57,07	23,13	60,20

TABLE 9 – Résultats obtenus après optimisation par MERT après avoir ajouté nos nouveaux modèles pour le sens de traduction anglais→français

de la traduction du corpus de développement et soient au contraire très peu adaptés au 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, nous obtenons assez peu de variété dans les scores de nos nouveaux modèles. Par exemple, les scores de *flc* 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 le corpus d'évaluation dans le sens de traduction français→anglais, les nouveaux modèles, et plus particulièrement leur combinaison, permettent d'obtenir des améliorations (certes modestes) des traductions sur les corpus de développement et d'évaluation pour la tâche de traduction anglais→français. Ils peuvent donc être utilisés pour reclasser des hypothèses et obtenir une traduction améliorée. Le score de ces 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. Cependant, il faut pouvoir limiter la taille du voisinage considéré, le calcul de ces modèles s'avérant relativement coûteux. Dans la section suivante, nous proposons de limiter l'espace de recherche considéré en limitant la récriture de segments appartenant à des zones de plus faible confiance.

2.4 Récriture des zones de faible confiance

Afin d'empêcher autant que possible la recherche locale de dégrader l'hypothèse amorce, 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. [11]

11. Notons toutefois que notre définition des segments de confiance est ici rigide, car il n'est par exemple par possible de les fusionner avec les segments de leur contexte. Ceci aura possiblement moins d'impact dans le cadre d'expériences oracles, mais pourrait s'avérer très limitant pour des configurations

Les Tableaux 10 et 11 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.

		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

TABLE 10 – 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.

		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

TABLE 11 – 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.

Les résultats obtenus montrent qu'en contraignant la recherche locale à effectuer des opérations sur les zones de faible confiance oracle, il est désormais 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 également 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

où la mesure de confiance dans la traduction des segments serait automatique.

être transformés par une opération de recherche locale.

On peut conclure de ces observations qu'une bonne mesure de confiance au niveau des segments pourrait permettre à la recherche locale d'être mieux guidée en se concentrant sur les zones qui ont le plus à bénéficier d'une récriture. Plusieurs améliorations de cette méthode de figement sont envisageables. L'implémentation des figements présentée dans cette section peut notamment 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 deux 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 à 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 ..*

3 Conclusions

Ce travail a montré qu'une recherche locale oracle guidée par une métrique d'évaluation telle que sBLEU permet d'atteindre une amélioration très forte des traductions. L'ajout d'un opérateur `rewrite`, modifiant la phrase source pour atteindre de nouvelles traductions, a permis à la recherche locale d'améliorer un peu plus les traductions en dépit de sa faible utilisation. Nous avons aussi montré que le potentiel d'amélioration des traductions est aussi très dépendant de la paire de langue utilisée.

La réutilisation de la fonction de score d'un système de traduction état de l'art pour guider la recherche n'a cependant pas permis d'améliorer les traductions produites, et a, au contraire, eu un effet négatif sur celles-ci. Ces dégradations mettent en évidence les limites de la fonction de score d'un tel système pour modéliser une bonne traduction. Comme piste d'amélioration de cette fonction de score nous avons proposé l'ajout de modèles apportant de nouvelles informations sur la grammaticalité d'une hypothèse et avons validé le potentiel de cette approche sur une tâche de reclassement. De plus, afin d'éviter que la recherche locale ne dégrade des zones correctes de l'hypothèse, nous avons mis en place un figement *oracle* de ces zones, ce qui cette fois-ci a permis d'obtenir des améliorations des traductions en utilisant la fonction de score du système. Deux améliorations possibles se présentent : implémenter des mesures de confiance automatiques (ex. [Bach et al., 2011, de Gispert et al., 2012], ainsi que les travaux conduits dans le cadre du Lot 2 du projet), et réoptimiser les poids de la fonction de scores par niveaux de confiance.

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, N. Pécheux, Q. K. Do, M. Dinarelli, T. Lavergne, A. Max, H.-S. Le, and F. Yvon. LIMSI @ WMT13. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 62–69, Sofia, Bulgaria, 2013. URL `http://www.aclweb.org/anthology/W13-2204`.

N. Bach, F. Huang, and Y. Al-Onaizan. Goodness : A Method for Measuring Machine Translation Confidence. In *ACL*, Portland, USA, 2011.

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

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. 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.

A. de Gispert, G. Blackwood, G. Iglesias, and W. Byrne. N-gram posterior probability confidence measures for statistical machine translation : an empirical study. *Machine Translation*, 2012.

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 *Proc. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic, 2007.

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`.

P. P. Monty. Traduction statistique par recherche locale. Master's thesis, Université de Montréal, 2010.

F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, 2003.

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.

G. Wisniewski and F. Yvon. Oracle decoding as a new way to analyze phrase-based machine translation. *Machine Translation*, pages 1–24, 2013.