TALN 2008, Avignon, 9–13 juin 2008

Recherche locale pour la traduction statistique à base de segments

Philippe Langlais, Alexandre Patry et Fabrizio Gotti
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal,
C.P. 6128, succursale Centre-Ville
H3C 3J7, Montréal, Québec, Canada
{felipe,patryale,gottif}@iro.umontreal.ca

Résumé. Dans cette étude, nous nous intéressons à des algorithmes de recherche locale pour la traduction statistique à base de segments (*phrase-based* machine translation). Les algorithmes que nous étudions s'appuient sur une formulation complète d'un état dans l'espace de recherche contrairement aux décodeurs couramment utilisés qui explorent l'espace des préfixes des traductions possibles. Nous montrons que la recherche locale seule, permet de produire des traductions proches en qualité de celles fournies par les décodeurs usuels, en un temps nettement inférieur et à un coût mémoire constant. Nous montrons également sur plusieurs directions de traduction qu'elle permet d'améliorer de manière significative les traductions produites par le système à l'état de l'art `Pharaoh` (Koehn, 2004).

Abstract. Most phrase-based statistical machine translation decoders rely on a dynamic-programming technique for maximizing a combination of models, including one or several language models and translation tables. One implication of this choice is the design of a scoring function that can be computed incrementally on partial translations, a restriction a search engine using a complete-state formulation does not have. In this paper, we present experiments we conducted with a simple, yet effective greedy search engine. We report significant improvements in translation quality over a state-of-the-art beam-search decoder, for some configurations.

Mots-clés : Traduction statistique, recherche locale, post-traitement.

Keywords: Statistical Machine Translation, local search, post-processing.

1 Introduction

Au début des travaux sur la traduction statistique (TS), plusieurs chercheurs se sont intéressés au problème de la recherche d'une meilleure traduction, étant donné un modèle de traduction basé sur les mots (Berger *et al.*, 1994; Tillmann *et al.*, 1997; Wang & Waibel, 1997; Niessen *et al.*, 1998; García & Casacuberta, 2001). Avec la montée en popularité des approches à base de segments (Koehn *et al.*, 2003), des décodeurs dédiés ont commencé à voir le jour au sein de notre communauté, comme `Pharaoh` (Koehn, 2004), distribué sous forme d'un exécutable,

ainsi que différentes variantes *logiciel-libre* comme Moses [1] (Koehn *et al.*, 2006), Ramses [2] (Patry *et al.*, 2006), Phramer [3] (Olteanu *et al.*, 2006) ou Marie [4] (Crego & Marino, 2007) pour des modèles n-grammes bilingues. De nombreuses équipes utilisent ces boîtes à outils pour construire leurs propres systèmes de TS (Déchelotte *et al.*, 2007; Besacier *et al.*, 2007).

Tous ces décodeurs partagent la propriété de s'appuyer sur une fonction de score incrémentale de manière à pouvoir organiser l'espace de recherche efficacement à l'aide de la programmation dynamique (DP). Il n'est pas difficile d'imaginer des modèles de traduction où cette propriété n'est pas appropriée.

Le moteur de traduction ReWrite [5] (Germann *et al.*, 2001), qui utilise un modèle de traduction mot-à-mot (Brown *et al.*, 1993) est une exception notable dans ce paysage. Il s'agit d'un algorithme de recherche locale qui tente d'améliorer de manière itérative une traduction courante, en lui faisant subir un ensemble de perturbations. À chaque itération, la meilleure traduction issue de ces perturbations devient l'hypothèse courante. Le processus se termine lorsqu'il n'est plus possible d'améliorer cette dernière, ce qui arrive typiquement après quelques itérations. Une version rapide de cet algorithme est décrite dans (Germann, 2003). Il est cependant accepté que cet algorithme produit des traductions de qualité moindre que les décodeurs DP faisant usage des mêmes modèles de traduction (Foster *et al.*, 2003).

À notre connaissance, personne n'a fait l'étude d'algorithmes de recherche locale pour les modèles de traduction à base de segments. Cette étude est une réponse à cette lacune. Nous montrons qu'une implémentation simple de cette idée permet d'obtenir des traductions d'une qualité proche de celles produites par les décodeurs standards à un coût mémoire constant (alors qu'un décodeur standard requiert un espace mémoire à tout le moins linéaire avec la taille de la phrase à traduire) et en un temps de loin inférieur (quelques minutes contre quelques heures).

Nous montrons également que lorsqu'utilisé en cascade, à la sortie d'un décodeur à l'état de l'art, notre algorithme permet d'en améliorer les traductions. Différentes expériences illustrent à la fois la souplesse de l'approche et son potentiel comme méthode de post-traitement.

L'article est organisé comme suit. Dans la section 2, nous décrivons précisément notre approche. Le protocole expérimental est ensuite présenté en section 3. Nous décrivons les expériences réalisées en section 4 puis concluons cette étude et proposons des pistes de recherche en section 6.

2 Algorithme glouton

L'algorithme de recherche (voir figure 1) que nous étudions est une forme particulièrement simple de recherche locale souvent nommée recherche gloutonne. Il utilise une formulation complète, ce qui signifie qu'un état dans l'espace de recherche est une traduction possible, à contrario des décodeurs standards qui parcourent plutôt l'espace des préfixes de traductions possibles. Plus précisément, un état, que nous désignons de manière interchangeable par hypothèse, est la donnée d'une traduction du texte source et d'un alignement entre les segments (*phrases*) source et cibles.

1. http://www.statmt.org/moses/
2. http://smtmood.sourceforge.net
3. http://www.phramer.org
4. http://gps-tsc.upc.es/soft/marie
5. http://www.isi.edu/publications/licensed-sw/rewrite-decoder/index.html

L'algorithme (désigné par `feGreedy` dans la suite) dépend de la définition de trois opérateurs : le premier (`seed`) est en charge de produire la première hypothèse courante, le second (`score`) implémente la fonction de score que nous tentons d'optimiser, le dernier (`voisinage`) propose les hypothèses voisines explorées à partir de l'hypothèse courante.

Require: $source$ une phrase à traduire
  $courant \leftarrow$ `seed`($source$)
  loop
    $s\_courant \leftarrow$ `score`($courant$)
    $s \leftarrow s\_courant$
    for all $h \in$ `voisinage`($courant$) **do**
      $c \leftarrow$ `score`(h)
      if $c > s$ **then**
        $s \leftarrow c$
        $meilleur \leftarrow h$
    if $s = s\_courant$ **then**
      return $courant$
    else
      $courant \leftarrow meilleur$

FIGURE 1 – Algorithme de recherche locale glouton.

Ce type de recherche possède trois caractéristiques intéressantes. Premièrement, une quantité constante (et réduite) de mémoire est requise pour représenter l'espace de recherche. Il s'agit de l'espace nécessaire à l'encodage de l'hypothèse courante. Deuxièmement, ce type d'algorithme propose souvent des solutions raisonnables (en terme de la fonction de score que l'on cherche à optimiser), en un temps habituellement très court, à des problèmes nécessitant une recherche combinatoire (Russell & Norvig, 1995). Troisièmement, aucune hypothèse n'est nécessaire quant à la fonction de score optimisée. En particulier, elle n'a pas besoin d'être calculée de manière incrémentale. Bien sûr, cet inventaire de points positifs est contrebalancé par le fait que cet algorithme ne possède aucune propriété d'optimalité. Nous verrons que ce défaut n'est pas pénalisant dans notre cas.

2.1 La fonction de score

Dans ce travail, nous cherchons à maximiser la combinaison habituellement utilisée en TS à base de segments. En particulier, nous nous intéressons dans un premier temps à maximiser la même fonction que celle que le décodeur à l'état de l'art `Pharaoh` (Koehn, 2004) maximise :

$$Score(e,f) = \lambda_{lm} \log p_{lm}(f) + \textstyle\sum_i \lambda_{tm}^{(i)} \log p_{tm}^{(i)}(f|e) - \lambda_d p_d(e,f) - \lambda_w |f| \qquad (1)$$

où les λ sont des coefficients contrôlant la contribution de chaque modèle à la combinaison, p_{lm} est un modèle de langue (n-gramme), $p_{tm}^{(i)}$ représente différentes tables de transfert (qui dans nos expériences partagent les mêmes paramètres), $|f|$ représente la longueur comptée en mots de la traduction et $p_d(e,f)$ est un modèle appelé généralement modèle de distorsion (nous utilisons le modèle simple décrit dans (Koehn *et al.*, 2003)).

SRC : le groupe csu au parlement européen se réjouit que le présent projet de charte des droits fondamentaux rassemble et rende visibles les droits fondamentaux dont disposent les citoyens vis-à-vis des organes et institutions de l ' ue .
`Pharaoh` the csu group in the european parliament welcomes the draft charter of fundamental rights lumps together and make visible the fundamental rights enjoyed by the citizens towards the eu institutions and bodies that . *(-43.8823)*

FIGURE 2 – Exemple d'une traduction produite par `Pharaoh` pour une phrase française et son score. Deux segments sources adjacents sont traduits à tort de manière distante.

2.2 Fonction de voisinage

Par inspection de traductions produites par `Pharaoh`, nous avons défini six familles de perturbation d'une hypothèse courante. Cet ensemble n'est en aucun cas exhaustif. En particulier, nous n'autorisons pas encore qu'un mot ou un segment soit inséré ou bien détruit.

Move `Pharaoh` (comme nombre de ses clones) s'autorise à reporter à plus tard la traduction d'un segment source afin de traduire le segment qui le suit (traduction non monotone). Ce comportement est souhaitable pour rendre compte de certaines divergences locales entre deux langues ; il introduit cependant le problème fréquent où des segments adjacents sont traduits à tort par des segments distants (le plus souvent sur la recommandation du modèle de langue). C'est par exemple le cas des segments *se réjouit* et *que* de l'exemple de la figure 2 traduits respectivement par *welcomes* et *that*. Nous avons donc implémenté une opération qui autorise deux segments cibles distants [6] correspondant à la traduction de deux segments sources adjacents à être rapprochés (nous tentons tous les rapprochements possibles).

Swap Lorsqu'un segment du texte à traduire est absent du modèle de traduction, ce segment est traduit de manière compositionnelle à l'aide de segments plus petits. L'ordre des segments traductions est alors souvent un compromis fragile entre les recommandations du modèle de langue et du modèle de distorsion habituellement biaisé en faveur de traductions monotones. Dans le but de corriger certains ordonnancements, nous autorisons deux segments cibles adjacents à être inversés. La complexité [7] de cette opération est linéaire avec le nombre N de segments sources dans l'hypothèse courante.

Replace Cette opération permet de changer la traduction d'un segment source par une autre traduction validée par la table de transfert. Cette opération a une complexité en $\mathcal{O}(N \times T)$, où T est le nombre de traductions considérées pour une phrase source (valeur typique de 10).

Bi-replace De la même manière, nous autorisons deux segments à changer simultanément de traduction avec l'espoir que cela permettra à notre algorithme d'échapper à certains maxima locaux. La complexité de cette opération est quadratique en T.

Split Un segment source peut être scindé en deux parties, pour autant que les sous-parties soient présentes dans la table de transfert. Cette opération est d'une complexité en $\mathcal{O}(N \times S \times T^2)$, où S est le nombre de segments sources dans l'hypothèse courante.

Merge Il s'agit de l'opération inverse de la précédente. Il convient de noter que ces deux opérations s'accompagnent généralement d'un changement lexical de la traduction courante (d'où la dépendance à T).

6. Sont dits distants dans cette étude deux blocs séparés par au moins 3 mots.
7. Nous mesurons le nombre d'hypothèses voisines engendrées par une opération.

$\mathcal{M}$	de plus , ↔ furthermore , || in addition , … le bon exemple ↔ a good example être modernisés ↔ modernization || modernized … modernisés ↔ modernized	
F	de plus , nos systèmes administratifs doivent être modernisés . nous devons également donner le bon exemple .	
E	*in addition , our administrative systems must be modernised , and it is our duty to lead by example .*	
S_0	[de plus ,] [nos systèmes administratifs] [doivent] [être modernisés] [. nous devons également] [donner le bon exemple .]	
T_0	[furthermore ,] [our administrative systems] [must] [modernization] [and we also need] [set a good example .]	*-19.5068*
S_1	[de plus ,] [nos systèmes administratifs] [doivent] [être modernisés] **[.] [nous devons également]** [donner le bon exemple .]	
T_1	[furthermore ,] [our administrative systems] [must] [modernization] **[.] [we must also]** [set a good example .]	SPLIT *-17.4382*
S_2	[de plus ,] [nos systèmes administratifs] [doivent] **[être] [modernisés]** [.] [nous devons également] [donner le bon exemple .]	
T_2	[furthermore ,] [our administrative systems] [must] **[be] [modernized]** [.] [we must also] [set a good example .]	SPLIT *-15.8488*
S_3	[de plus ,] [nos systèmes administratifs] [doivent] [être] [modernisés] [.] [nous devons également] **[donner] [le bon exemple .]**	
T_3	[furthermore ,] [our administrative systems] [must] [be] [modernized] [.] [we must also] **[give] [a good example .]**	SPLIT *-15.5885*
S_4	**[de plus ,]** [nos systèmes administratifs] [doivent] [être] [modernisés] [.] [nous devons également] [donner] [le bon exemple .]	
T_4	**[in addition ,]** [our administrative systems] [must] [be] [modernized] [.] [we must also] [give] [a good example .]	REPLACE *-15.5199*

FIGURE 3 – Itérations impliquées dans la traduction par initialisation GLOSS (section 2.3) d'une phrase française (F) de traduction de référence (E). Une segmentation (S_0) est premièrement calculée à partir des 49 segments sources de $\mathcal{M}$ qui couvrent partiellement F. T_0 est la traduction (GLOSS) associée. Les segments en gras sont impliqués dans l'hypothèse recevant le meilleur score à chaque itération. Plus de 4,100 hypothèses ont été évaluées en un tiers de seconde.

2.3 L'opérateur d'initialisation

Initialisation GLOSS Dans `ReWrite` (Germann *et al.*, 2001), l'hypothèse courante est initialisée en collectant pour chaque mot sa traduction privilégiée selon le modèle lexical (paires de mots source/cible). Nous avons adapté cette idée aux modèles de segments (paires de séquences de mots source/cible). Une complication survient dans notre cas, puisque la phrase à traduire S n'est pas pré-découpée en segments. Plusieurs segmentations étant possibles, nous avons décidé de retenir celle qui minimise le nombre de segments sources du modèle de segment $\mathcal{M}$, tout en couvrant complètement S. Notre espoir est ici que des segments longs captureront plus d'information pertinente à leur traduction hors-contexte. Cette segmentation peut être implémentée efficacement par programmation dynamique (Langlais *et al.*, 2007).

Une fois la segmentation source effectuée, nous prenons simplement la traduction privilégiée (selon $\mathcal{M}$) de chaque segment que nous concaténons pour former une traduction.

Initialisation par `Pharaoh` Nous avons testé une autre manière d'initialiser la recherche. Elle consiste à partir de la meilleure traduction produite par `Pharaoh` [8]. Cela revient à dire que nous faisons le pari que la recherche locale permet de corriger certaines erreurs faites par le premier décodeur. Nous appelons cette variante CASCADE dans la suite.

La figure 3 montre un exemple d'application de la recherche locale dans le cas de l'initialisation GLOSS.

3 Protocole expérimental

3.1 Corpus

Nous avons réalisé nos expériences en utilisant les ressources de la tâche partagée du workshop sur la traduction statistique qui s'est tenu en 2006, en marge de l'ACL (Koehn & Monz, 2006). Cette année-là, les systèmes participants avaient à traduire des textes en espagnol, en allemand et en français vers et depuis l'anglais. Les textes disponibles pour l'entraînement proviennent du corpus Europarl. Une portion d'environ 700 000 paires de phrases dans chaque langue, `train`, constituait le matériel d'entraînement ; deux corpus de développement de 2 000 phrases chacun, `dev` et `devtest`, étaient destinés respectivement à ajuster les systèmes (les λ dans l'équation 1) et à réaliser des tests à blanc. Nous avons utilisé pour nos tests les 2 000 phrases du jeu de test officiel de la tâche partagée extraites du corpus Europarl [9].

3.2 Système de référence

Le système de base que nous utilisons dans cette étude est le système état-de-l'art mis à disposition par les organisateurs. Il s'agit d'un système maintenant classique où le modèle de langue est un modèle trigramme entraîné à l'aide de SRILM (Stolcke, 2002), les tables de traductions (avec des segments d'au plus 7 mots) sont entraînées par les scripts fournis par les organisateurs. Chaque paire de segments dans cette table est notée par quatre scores recevant chacun leur coefficient de pondération (λ) ainsi qu'un score permettant de contrôler (de manière passive) la longueur des traductions produites (*phrase penalty*). Le modèle de distorsion natif à `Pharaoh` ainsi qu'un second modèle de contrôle de la longueur des traductions (*word penalty*) reçoivent à leur tour un coefficient. Au total, ce sont huit coefficients qui sont ajustés sur `dev` en appliquant l'algorithme de minimisation d'erreur `minimum-error-rate-training.perl`.

4 Expériences

Nous comparons [10] dans un premier temps `feGreedy` et `Pharaoh` [11] en leur demandant de maximiser la même fonction (équation 1). Les deux variantes du premier moteur (GLOSS et CASCADE) sont testées. Les résultats sont indiqués dans le tableau 1.

8. `Pharaoh` donne accès à l'alignement ayant produit la meilleure traduction grâce à l'option `-trace`.
9. Des résultats similaires sont observés sur la partie hors-domaine du jeu de test.
10. Par simplicité nous mesurons dans la suite la qualité des traductions produites à l'aide des mesures automatiques WER (pour *word error rate*) et BLEU (Papineni *et al.*, 2002) couramment employées dans la communauté.
11. Les seuils contrôlant l'espace de recherche que `Pharaoh` explore ont été laissés à leurs valeurs par défaut.

		L=fr		L=es		L=de	
		WER	BLEU	WER	BLEU	WER	BLEU
	Pharaoh	54.85	30.90	54.23	29.64	62.32	17.68
	GLOSS	54.27	29.83	**53.22**	28.99	62.53	17.03
L → en	CASCADE	**53.38**	**31.42**	**52.77**	**30.14**	**61.73**	17.88
	Pharaoh	51.69	29.96	51.04	30.54	60.54	24.45
	GLOSS	50.93	29.13	50.77	29.67	**57.55**	23.84
en → L	CASCADE	**50.46**	**30.27**	**50.02**	**30.87**	**58.85**	24.66

TABLE 1 – Performances de différents algorithmes de recherche. Les données en gras sont significativement meilleures (à 99%) que celles associées à Pharaoh.

La variante GLOSS enregistre des valeurs de BLEU inférieures à celles mesurées pour Pharaoh, les différences sont cependant assez faibles. Les taux d'erreurs au niveau des mots sont en fait le plus souvent en faveur de GLOSS. Ceci est d'autant plus remarquable que notre implémentation n'encode qu'un nombre restreint d'opérations de voisinage. Nous observons avec intérêt que CASCADE permet d'améliorer les traductions produites par Pharaoh, ce qui constitue un résultat très satisfaisant et valide l'idée que la recherche locale offre une façon simple et efficace de corriger les traductions produites par un système natif. Pour toutes les directions de traduction, les améliorations apportées par CASCADE sont significatives [12].

Une analyse plus fine des traces de cette session de traduction permet d'observer que 40% des traductions produites par Pharaoh ont un meilleur score (équation 1) après application de la recherche locale (CASCADE). C'est donc en terme d'algorithme de recherche un succès. De manière moins surprenante, 90% des traductions initiales produite par GLOSS sont améliorées par la recherche locale.

Pas moins de 40% des opérations remportant une itération dans l'algorithme local sont des opérations de remplacement (replace) d'une traduction par une autre. L'opération move est également productive dans la variante CASCADE et illustre bien le pouvoir de post-correction qu'offre la recherche locale. Une fois un problème identifié dans les traductions produites par un système natif, il "suffit" d'encoder une opération spécifique visant à sa correction ; ce que nous avons fait pour l'opération move.

Certaines opérations sont marginalement utiles. C'est par exemple le cas de l'opération swap ce qui s'explique par le fait que la table de transfert capture déjà de nombreux réordonnancements locaux. En dernière observation, soulignons que CASCADE requiert beaucoup moins d'itérations pour converger que GLOSS, ce qui semble normal. 70% des traductions effectuées par CASCADE nécessitent au plus 2 itérations, alors que seulement un peu plus de la moitié des traductions effectuées par GLOSS requièrent un maximum de 4 itérations. Dans les deux cas, les deux variantes requièrent habituellement moins de 10 itérations avant stabilisation.

Nous tenons à souligner que bien que n'ayant pas pris la peine d'implémenter une version efficace de notre moteur de traduction, feGreedy requiert de l'ordre de 4 minutes de calculs pour traduire 1 000 phrases [13], contre plus d'une heure pour Pharaoh. Réduire les temps de

12. Selon le test d'échantillonnage multiple avec replacement (*boostrap resampling*) décrit dans (Zhang & Vogel, 2004), en évaluant 1 000 échantillons de 700 phrases chacun. Niveau de confiance fixé à 99%.

13. Ce temps ne tient pas compte du calcul de la première hypothèse courante. Temps approximatifs mesurés sur un ordinateur Pentium à 3 GHz disposant de 4Go de mémoire vive.

	Pharaoh			CASCADE		
pile	WER	BLEU	temps	WER	BLEU	temps
50	51.82	29.24	40min.	50.26	29.65	<5 min.
100	51.46	29.23	1h. 20min.	50.32	29.62	<5 min.
200	51.15	29.44	2h. 40min.	50.18	29.69	<5 min.
500	50.86	29.51	6h. 15min.	50.11	29.74	<5 min.
1000	50.64	29.54	12h. 15min.	50.04	29.74	<5 min.

TABLE 2 – Performance de Pharaoh et de CASCADE sur les 1 000 premières phrases du corpus de test en fonction du nombre maximum d'hypothèses stockées par Pharaoh dans une pile.

calcul de notre implémentation serait facile puisqu'à chaque opération de voisinage, une nouvelle hypothèse est temporairement construite puis évaluée au complet, alors qu'une opération n'introduit en général que peu de modifications dans le calcul de l'équation 1.

Nous concluons cette exploration de la recherche locale par une expérience où nous augmentons le nombre d'hypothèses que Pharaoh est autorisé à manipuler par pile. Les résultats de cette expérience sont consignés en tableau 2 pour des systèmes traduisant du français vers l'anglais.

Nous observons d'une part qu'augmenter l'espace de recherche est payant, puisque plus d'un point en WER peut être gagné de cette façon. Ce gain ne doit pas nous faire oublier cependant que le temps mis pour obtenir les traductions passe de 40 minutes à plus de 12 heures lorsqu'on passe d'une limite de 100 hypothèses par pile à 1 000. De manière plus intéressante, nous constatons surtout que CASCADE permet systématiquement d'améliorer la meilleure traduction produite par Pharaoh, que l'on mesure cette amélioration par WER ou par BLEU. L'amélioration en BLEU apportée par CASCADE à la meilleure traduction produite par la première version de Pharaoh (la plus rapide) est supérieure à celle enregistrée par la version la plus longue de Pharaoh (+.4 *versus* +.3). Moins de 5 minutes ont été nécessaires pour obtenir cette amélioration, contre presque 12 heures dans le second cas.

5 Travaux reliés

L'idée de composer des moteurs de traduction en cascade a été proposée initialement par (Berger *et al.*, 1994) dans le cadre du système Candide ; système à base de modèles de traduction mot à mot (Brown *et al.*, 1993). Malheureusement, les auteurs ne décrivent ni leur algorithme de recherche locale, ni n'en fournissent une évaluation. D'autres travaux ont été menés sur cette idée. Notamment (Marcu, 2001) et (Watanabe & Sumita, 2003) où un algorithme de recherche locale basé sur les mots tente d'améliorer une traduction produite par un système de mémoires de traductions (sous-phrastique dans le premier cas, phrastique dans le second).

Plus récemment, (Simard *et al.*, 2007) et (Chen *et al.*, 2007) présentaient simultanément la même idée qui consiste à entraîner un modèle de traduction statistique à partir d'un bitexte dont la partie source est produite par un système natif (le système Systran dans ces études) et la partie cible est une traduction de référence (manuelle) ; l'espoir étant que le modèle de traduction résultant saura corriger des erreurs commises par le système natif. Il est important de souligner que notre approche, bien qu'elle puisse être utilisée comme une étape de post-traitement, ne requiert aucun entraînement d'un modèle de traduction supplémentaire.

6 Conclusions et perspectives

Dans cette étude, nous avons développé un algorithme de recherche locale pour un système de traduction statistique basé sur les segments. Nous avons discuté les avantages de notre approche et avons réalisé des expériences la validant. Nous avons montré qu'une variante de cet algorithme permet d'améliorer les traductions produites par le système à l'état de l'art `Pharaoh`.

Cette étude ouvre plusieurs perspectives. Nous souhaitons en particulier comparer différents algorithmes de recherche locale, et étudier l'influence du processus de segmentation permettant d'initialiser l'hypothèse courante. Notre motivation initiale était d'explorer des approches souples à la post-édition de traductions qui peuvent identifier des erreurs systématiques dans les traductions produites par un système donné. Un pas dans cette direction consiste à augmenter le nombre de modèles utilisés dans la fonction de score et d'en ajuster les contributions via les coefficients qui leur sont associés.

Remerciements

Nous remercions les relecteurs pour leur commentaires pertinents.

Références

BERGER A. L., BROWN P. F., PIETRA S. A. D., PIETRA V. J. D., GILLETT J. R., LAFFERTY J. D., MERCER R. L., PRINTZ H. & UREŠ L. (1994). The Candide system for machine translation. In *HLT*, p. 157–162, Morristown, NJ, USA.

BESACIER L., MAHDHAOUI A. & LE V.-B. (2007). The LIG Arabic/English speech translation system at IWSLT 07. In *4th IWSLT*, p. 137–140, Trento, Italy.

BROWN P. F., PIETRA S. A. D., PIETRA V. J. D. & MERCER R. L. (1993). The mathematics of statistical machine translation : Parameter estimation. *Computational Linguistics*, **19**(2), 263–311.

CHEN Y., EISELE A., FEDERMANN C., HASLER E., JELLINGHAUS M. & THEISON S. (2007). Multi-engine machine translation with an open-source SMT decoder. In *2nd Workshop on SMT*, p. 193–196, Prague, Czech Republic.

CREGO J. M. & MARINO J. B. (2007). Extending MARIE : a N-gram-based smt decoder. In *ACL, Demo and Poster Sessions*, p. 213–216, Prague.

DÉCHELOTTE D., SCHWENK H., BONNEAU-MAYNARD H., ALLAUZEN A. & ADDA G. (2007). A state-of-the-art statistical machine translation system based on Moses. In *XIth MT Summit*, p. 127–133, Copenhagen, Denmark.

FOSTER G., GANDRABUR S., LANGLAIS P., PLAMONDON P., RUSSEL G. & SIMARD M. (2003). Statistical machine translation : Rapid development with limited resources. In *MT Summit IX*, p. 110–117, New Orleans.

GARCÍA I. & CASACUBERTA F. (2001). Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *8th MT Summit*, p. 115–120, Santiago de Compostela, Spain.

GERMANN U. (2003). Greedy decoding for statistical machine translation in almost linear time. In *HLT-NAACL*, p. 72–79, Edmonton, Canada.

GERMANN U., JAHR M., KNIGHT K., MARCU D. & YAMADA K. (2001). Fast decoding and optimal decoding for machine translation. In *39th ACL*, p. 228–235, Toulouse, France.

KOEHN P. (2004). Pharaoh : a Beam Search Decoder for Phrase-Based SMT. In *6th AMTA*, p. 115–124, Washington, DC.

KOEHN P., FEDERICO M., SHEN W., BETOLDI N., HOANG H., CALLISON-BURCH C., COWAN B., ZENS R., DYER C., BOJAR O., C.MORAN, CONSTANTIN A. & HERBST E. (2006). *Open Source Toolkit for Statistical Machine Translation : Factored translation models and confusion network decoding*. University summer worskhop, Johns Hopkins University.

KOEHN P. & MONZ C. (2006). Manual and automatic evaluation of machine translation between European languages. In *1st Workshop on SMT*, p. 102–121, New York City.

KOEHN P., OCH F. J. & MARCU D. (2003). Statistical Phrase-Based Translation. In *HLT*, p. 127–133.

LANGLAIS P., PATRY A. & GOTTI F. (2007). A greedy decoder for statistical phrase-based machine translation. In *11th TMI*, p. 104–113, Skövde, Sweden.

MARCU D. (2001). Towards a unified approach to memory- and statistical-based machine translation. In *39th ACL*, p. 378–385, Toulouse, France.

NIESSEN S., VOGEL S., NEY H. & TILLMANN C. (1998). A DP-based search algorithm for statistical machine translation. In *36th ACL and 17th COLING*, p. 960–966, Montréal, Canada.

OLTEANU M., DAVIS C., VOLOSEN I. & MOLDOVAN D. (2006). Phramer – an open source statistical phrased-based translator. In *1st Workshop on SMT*, p. 150–153, New York, USA.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*, p. 311–318, Philadelphia, Pennsylvania.

PATRY A., GOTTI F. & LANGLAIS P. (2006). Mood : A modular object-oriented decoder for statistical machine translation. In *5th LREC*, p. 709–714, Genoa, Italy.

RUSSELL S. & NORVIG P. (1995). *Artificial Intelligence. A Modern Approach*. Prentice Hall.

SIMARD M., UEFFING N., ISABELLE P. & KUHN R. (2007). Rule-based translation with statistical phrase-based post-editing. In *2nd Workshop on SMT*, p. 203–206, Prague, Czech Republic.

STOLCKE A. (2002). SRILM - an extensible language modeling toolkit. In *ICSLP*, Denver, Colorado.

TILLMANN C., VOGEL S., NEY H. & ZUBIAGA A. (1997). A DP-based search using monotone alignments in statistical translation. In *35th ACL*, p. 289–296, Madrid, Spain.

WANG Y.-Y. & WAIBEL A. (1997). Decoding algorithm in statistical machine translation. In *35th ACL*, p. 366–372, Madrid, Spain.

WATANABE T. & SUMITA E. (2003). Example-based decoding for statistical machine translation. In *MT Summit IX*, p. 410–417, New Orleans, Louisiana.

ZHANG Y. & VOGEL S. (2004). Measuring confidence intervals for the machine translation evaluation metrics. In *10th TMI*, p. 85–94, Baltimore, Maryland, USA.