TALN 2005, Dourdan, 6–10 juin 2005

Traduction automatique statistique avec des segments discontinus

Michel Simard∗, Nicola Cancedda∗, Bruno Cavestro∗,
Marc Dymetman∗, Eric Gaussier∗, Cyril Goutte∗,
Philippe Langlais†, Arne Mauser‡, Kenji Yamada⋆

(∗) Xerox Research Centre Europe (XRCE)
prenom.nom@xrce.xerox.com

(†) Laboratoire RALI, Université de Montréal
felipe@iro.umontreal.ca

(‡) Lehrstuhl für Informatik VI, RWTH Aachen
arne.mauser@kullen.rwth-aachen.de

(⋆) USC Information Science Institute
kyamada@isi.edu

Mots-clefs : traduction automatique statistique, segments discontinus, modèles log-linéaires

Keywords: statistical machine translation, discontinuous phrases, log-linear models

Résumé Cet article présente une méthode de traduction automatique statistique basée sur des segments non-continus, c'est-à-dire des segments formés de mots qui ne se présentent pas nécéssairement de façon contiguë dans le texte. On propose une méthode pour produire de tels segments à partir de corpus alignés au niveau des mots. On présente également un modèle de traduction statistique capable de tenir compte de tels segments, de même qu'une méthode d'apprentissage des paramètres du modèle visant à maximiser l'exactitude des traductions produites, telle que mesurée avec la métrique NIST. Les traductions optimales sont produites par le biais d'une recherche en faisceau. On présente finalement des résultats expérimentaux, qui démontrent comment la méthode proposée permet une meilleure généralisation à partir des données d'entraînement.

Abstract This paper presents a phrase-based statistical machine translation method, based on non-contiguous phrases, i.e. phrases with *gaps*. A method for producing such phrases from a word-aligned corpora is proposed. A statistical translation model is also presented that deals with such phrases, as well as a training method based on the maximization of translation accuracy, as measured with the NIST evaluation metric. Translations are produced by means of a beam-search decoder. Experimental results are presented, that demonstrate how the proposed method allows to better generalize from the training data.

1 Introduction

L'évolution des modèles et des méthodes et la prolifération des corpus parallèles ont, depuis peu, poussé les approches statistiques à l'avant-plan de la recherche en traduction automatique. Bien qu'on retrouve toujours au coeur de ces approches le cadre général qui a motivé les propositions initiales de l'équipe IBM (Brown et al.1993), on a pu observer des transformations importantes au cours des dernières années. La plus remarquable est sans doute le passage du niveau des mots à celui de *segments* de longueur variable[1] (Och et al.1999; Marcu and Wong2002; Tillmann and Xia2003). Alors que les modèles traditionnels prenaient pour unité de base le mot, les modèles "segmentaires" reconnaissent le rôle primordial que jouent dans la langue les expressions combinant plusieurs mots, et l'importance de les traduire en bloc. C'est bien sûr le cas des multitermes, qu'on rencontre plus fréquemment dans les domaines techniques et spécialisés, mais aussi des expressions idiomatiques, des locutions, et de tout un ensemble de phénomènes de la langue générale.

Mais le succès des approches segmentaires ne s'explique pas uniquement par l'importance et la fréquence de ces phénomènes linguistiques. En fait, l'utilisation de segments de plus d'un mot améliore la qualité des traductions, même lorsque ces segments n'ont pas de réel statut linguistique. Face à la rareté des événements sur lesquels se fonde l'estimation des nombreux paramètres d'un modèle de traduction, le concepteur se retrouve souvent devant un choix difficile, entre des estimations peu fiables et un lissage plus ou moins arbitraire. À défaut de résoudre ce dilemme, l'emploi d'unités plus longues représente l'application d'un principe intuitif : lorsqu'on a vu un long segment de texte en langue-source souvent traduit d'une certaine façon, il y a tout lieu de croire que cette traduction est préférable à toute autre qu'on pourrait obtenir de façon compositionnelle. En somme, les modèles segmentaires incorporent dans un cadre statistique l'intuition derrière la traduction automatique basée sur les exemples et, à la limite, les mémoires de traduction. Finalement, les segments de plusieurs mots contribuent à résoudre le problème du choix lexical face aux ambiguïtés de la langue-source. Alors que le mot anglais *bank* se traduit presque systématiquement par *banque* en français, il suffit d'avoir observé que *river bank* a été traduit par *rive*, ne fût-ce que quelques fois, pour produire la bonne traduction.

Les modèles segmentaires existants ne traitent que des segments constitués de mots contigus. Nous proposons ici un modèle capable de gérer des *segments discontinus*, c'est-à-dire des expressions formés de mots qui ne sont pas nécéssairement contigus, tant dans la langue-source que dans la langue-cible. La suite de cet article est ainsi structuré : en section 2, nous discutons des motivations pour traiter les segments discontinus, et présentons une méthode pour obtenir de telles unités, à partir d'un corpus d'entraînement; le modèle de traduction log-linéaire conditionnel que nous avons adopté fait l'objet de la section 3; nous décrivons brièvement le décodeur à la section 4; enfin, nous présentons en section 5 les résultats d'expériences que nous avons menées dans le but d'évaluer le potentiel de notre approche.

2 Les segments discontinus

Notre objectif, avec des segments constitués de mots non-contigus est d'améliorer la qualité des traductions produites, d'abord en élargissant la portée des effets mentionnés plus haut de

[1] On utilise couramment le terme *phrase* en anglais, de façon un peu abusive, faut-il souligner.

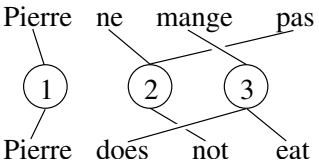

Figure 1: Alignement d'une négation, entre le français et l'anglais.

désambiguïsation lexicale et de traduction basée sur les exemples, mais aussi en prenant compte de nouveaux phénomènes linguistiques. Les verbes à particules, en anglais, constituent un exemple d'un tel phénomène. Dans une phrase comme "Mary *switches* her bedside lamp *off*" ("Marie éteint sa lampe de chevet") les modèles de traductions basés sur les mots sont généralement incapables de rendre compte de l'effet combiné de *switch* et de *off*. Alors qu'ils traitent correctement les locutions inséparables comme *to run out*, les modèles segmentaires existants sont tout aussi impuissants dans ce cas. Notons que ce phénomène ne se limite pas à l'anglais, puisqu'on l'observe également en allemand et dans bien d'autres langues.

Les unités linguistiques non-contiguës ne se limitent pas aux seuls verbes : la négation se forme de façon différente en français et en anglais, et les modèles existants sont incapables de représenter correctement l'alignement de mots complexe qui en résulte (figure 1). D'une façon générale, un modèle autorisant des relations de type *plusieurs-à-plusieurs* permet de rendre compte du fait qu'un même concept peut se voir réalisé par des unités de granularité différente dans différentes langues, sans égard pour la contiguïté.

Au sein d'une bi-phrase, nous appelons *bi-segment* une paire constituée d'un *segment-source* et d'un *segment-cible* : $b = \langle \tilde{f}, \tilde{e} \rangle$. Le segment-source est une suite de mots de la langue-source et de *jokers* (représentés par le symbole $\diamond$); on définit le segment-cible de manière analogue. Par exemple, $\tilde{f} = f_1 \diamond \diamond f_2 f_3$ est un segment-source de longueur 5, constitué d'un mot source, suivi de deux jokers, puis de deux mots-source contigus.

Avec de tels bi-segments, la traduction d'une phrase en langue-source f est produite en combinant les bi-segments $b = \langle \tilde{f}, \tilde{e} \rangle$ d'un ensemble choisi de façon d'une part à recouvrir entièrement la phrase f, et d'autre part à produire une phrase e bien formée dans la langue-cible. La production d'une traduction complète peut être décrite par une suite ordonnée de bi-segments $b_1 ... b_K$: on dépose d'abord le segment-cible $\tilde{e}_1$ du bi-segment b_1, puis chacun des segments subséquents $\tilde{e}_k$ sur la première position "libre", c'est-à-dire soit le joker le plus à gauche, soit l'extrémité droite de la séquence (figure 2) .

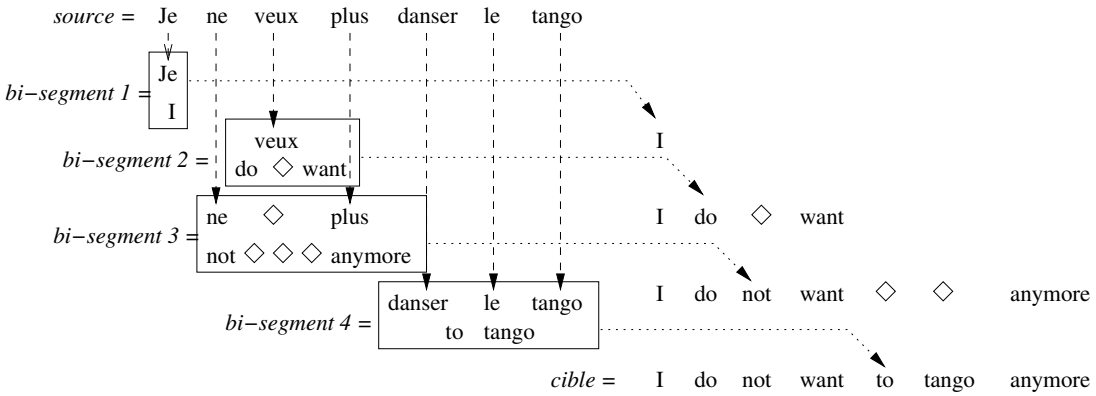

Figure 2: Production d'une traduction par combinaison de bi-segments.

Notre approche nécessite une *banque de bi-segments*, contenant les "briques" qui seront utilisées pour construire les traductions. La constitution d'une telle banque s'effectue en deux étapes : on aligne d'abord les mots d'un corpus bilingue, de façon à obtenir des bi-segments de base; on combine ensuite ces bi-segments, de manière à obtenir des briques de taille et de complexité croissante.

La première étape repose sur l'utilisation de la méthode d'alignement de mots proposée par (Goutte et al.2004). Cette méthode produit des alignements de type *plusieurs-à-plusieurs* entre les mots de la source et de la cible, par le biais d'une partition parallèle des deux textes, vus comme des ensembles de mots. Chaque mot appartient ainsi à un et un seul sous-ensemble dans cette partition, les sous-ensembles correspondants dans la source et la cible constituent ce qu'on appelle des *cepts*, et l'ensemble de ces cepts constitue l'alignement. Chaque cept réunit donc des mots de la source et de la cible, sans aucune contrainte de contiguïté. Dans la figure 1, ces cepts sont représentés par les cercles numérotés 1, 2 et 3.

L'ensemble des cepts observés dans un corpus bilingue constitue naturellement une banque de bi-segments élémentaires, que nous appelons L_1. Partant de là, on peut construire des banques de segments complexes : en combinant deux-à-deux les cepts provenant d'une même paire de phrases, on génère l'ensemble que nous appelons L_2. Par exemple, dans la figure 1, en combinant les cepts 1 et 2, on obtient le bi-segment $\langle$*Pierre ne* $\diamond$ *pas*, *Pierre* $\diamond$ *not*$\rangle$. Les combinaisons de 3 cepts produisent l'ensemble L_3, et ainsi de suite. La taille de ces ensembles croît théoriquement de façon exponentielle avec le nombre de cepts combinés. Comme nous le verrons plus loin, le nombre de bi-segments disponibles affecte directement le temps requis pour produire une nouvelle traduction. C'est pourquoi on aura recours à différentes méthodes de filtrage, visant à ne conserver que les bi-segments les plus susceptibles d'être utiles, en se basant par exemple sur la fréquence des observations dans un corpus de référence.

3 Le modèle de traduction

En traduction automatique statistique, étant donnée une phrase-source $f_1^J = f_1 ... f_J$, on recherche la phrase-cible $e_1^I = e_1 ... e_I$ qui en constitue la traduction la plus probable :

$$\hat{e}_1^I = \text{argmax}_{e_1^I}\{P(e_1^I|f_1^J)\}$$

Notre approche repose sur une modélisation directe de la probabilité a posteriori $P(e_1^I|f_1^J)$ au moyen d'un modèle log-linéaire :

$$P_\lambda(e_1^I|f_1^J) = \frac{1}{Z_{f_1^J}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right)$$

Dans cette équation, la contribution de chacune des *fonctions-attributs* h_m est pondérée par un facteur λ_m, lesquels constituent les paramètres du modèle; $Z_{f_1^J}$ représente un facteur de normalisation propre à la phrase source f_1^J. Il est possible d'introduire des variables additionnelles dans le modèle, de façon à tenir compte de phénomène cachés; on modifie alors les fonctions-attributs pour y incorporer ces variables. Par exemple, notre modèle doit prendre en compte l'ensemble des bi-segments qui est à l'origine d'une traduction; les fonctions-attributs auront donc la forme générale $h_m(e_1^I, f_1^J, b_1^K)$. Le recours à ce genre de modèle est maintenant monnaie courante en traduction automatique (Tillmann and Xia2003; Zens and Ney2003; Och and Ney2004).

Notre modèle repose présentement sur sept fonctions-attributs. h_{bp} est la *fonction-attribut des bi-segments*. Elle représente la probabilité de produire la phrase en langue-cible, étant donné le découpage de la source, tel que prescrit par l'ensemble de bi-segments utilisé, sous l'hypothèse que chaque segment-source génère un segment-cible de façon indépendante du reste de la phrase-source :

$$h_{bp}(e_1^I, f_1^J, b_1^K) = \sum_{k=1}^{K} \log P(\tilde{e}_k|\tilde{f}_k) \quad (1)$$

Les probabilités des segments-cible sont estimées sur la base de décomptes dans un corpus de référence aligné au niveau des mots. Cette fonction-attribut démontre une forte tendance à surestimer la probabilité des bi-segments peu fréquents. C'est pourquoi on introduit également une *fonction-attribut compositionnelle* h_{comp}, qui se calcule de la même façon que h_{bp} dans l'équation (1), sauf que les probabilités des segments-source sont estimées sur la base de probabilités de traduction des mots qui composent le bi-segment, à la manière du modèle IBM-1 (Brown et al.1993) :

$$P(\tilde{e}|\tilde{f}) = \frac{1}{|\tilde{f}|^{|\tilde{e}|}} \prod_{e \in \tilde{e}} \sum_{f \in \tilde{f}} P(e|f)$$

Ici encore, l'estimation des probabilités de traduction lexicales $P(e|f)$ se fonde sur des décomptes dans le corpus d'entraînement.

h_{tl} est la *fonction attribut langue-cible*. Elle repose sur un modèle N-gramme de la langue-cible. Elle ne tient donc compte que de la suite de mots e_1^I résultant de la combinaison des bi-segments.

Deux fonctions-attributs, h_{wc} et h_{bc}, contrôlent respectivement la longueur de la phrase-cible et le nombre de bi-segments ayant servi à produire celle-ci : $h_{wc}(e_1^I, f_1^J, b_1^K) = I$ et $h_{bc}(e_1^I, f_1^J, b_1^K) = K$. Une sixième fonction $h_{reord}(e_1^I, f_1^J, b_1^K)$ mesure le degré de divergence dans l'ordre des mots de la source et de la cible.

Toutes les fonctions ci-dessus font plus ou moins partie de l'arsenal habituel des fonctions-attributs employées en traduction automatique. Une seule fonction, h_{gc} concerne spécifiquement les segments discontinus, et permet au modèle de contrôler dans une certaine mesure la nature des segments qu'il utilise. Cette fonction prend pour valeur le nombre total de jokers apparaissant dans les segments (source ou cible) de b_1^K.

Nous choisissons les valeurs des paramètres λ_m de façon à maximiser la qualité des traductions produites sur un corpus d'entraînement, tel que proposé par (Och2003). À la différence de ce dernier, toutefois, nous avons développé une version de la métrique d'évaluation de traduction NIST (Doddington2002) qui est dérivable par rapport aux λ_m, ce qui ouvre la voie à l'utilisation de méthodes d'optimisation par descente de gradient (Newton, quasi-Newton, etc.). Pour chacune des phrases sources $f_1 ... f_S$ du corpus d'entraînement, notre système de traduction peut produire plusieurs phrases cibles $e_{s,k}$, ordonnées suivant les valeurs de $P_\lambda(e_{s,k}|f_s)$. Nous calculons alors une version de la métrique d'évaluation NIST, dans laquelle la contribution de chaque phrase est pondérée par :

$$w_{s,k}^{\alpha}(\lambda) = \frac{P_\lambda(e_{s,k}|f_s)^\alpha}{\sum_{k'} P_\lambda(e_{s,k'}|f_s)^\alpha},$$

où α est un paramètre de lissage qu'on fixe de manière expérimentale.

À la différence d'une approche par maximum de vraisemblance dans un modèle log-linéaire, qui correspond à un problème convexe et conduit à un minimum global unique, ce genre

d'apprentissage est assez sensible à l'initialisation des paramètres λ. Notre approche consiste alors à utiliser un ensemble d'initialisations aléatoires pour les paramètres, à effectuer l'optimisation pour chaque initialisation, et à choisir le modèle qui donne la meilleure performance.

Finalement, rappelons que cette procédure d'entraînement requiert des traductions multiples pour chaque phrase-source du corpus d'entraînement. En pratique, notre décodeur peut générer une liste des N-meilleures traductions de chaque phrase-source. Toutefois, différentes valeurs initiales des paramètres λ peuvent conduire à des listes différentes. Il est donc judicieux de répéter le processus : décodage des N-meilleures traduction, optimisation de la valeur de λ, re-décodage des N-meilleures traduction avec ces nouveaux paramètres, ré-optimisation de ceux-ci, etc. Afin d'assurer la convergence du processus d'optimisation, il convient de combiner à chaque itération les nouvelles N-meilleures traductions avec celles obtenues lors des itérations précédentes.

4 Le décodage

Notre méthode de décodage repose sur une recherche en faisceau par piles (*beam-search stack decoding*), tel que proposée dans (Koehn2003), que nous avons adaptée aux segments discontinus. La traduction d'une phrase en langue-source est le résultat d'une suite de *décisions*; chacune de celles-ci implique le choix d'un ensemble de positions à couvrir dans la phrase-source et d'un bi-segment adéquat. La traduction finale s'obtient en combinant ces décisions dans l'ordre, comme à la figure 2. Au cours du processus de décodage, les traductions partielles (que nous appelons des *hypothèses*) sont accumulées dans des listes (les *piles*), chacune desquelles regroupe des hypothèses qui recouvrent le même nombre de mots dans la phrase-source. On *étend* une hypothèse en y comblant la première position libre dans la cible (voir la section 3); chaque hypothèse ainsi étendue est stockée dans la pile correspondant au nouveau nombre de mots traduits dans la source.

On associe un score à chaque hypothèse. Ce score est la combinaison d'une composante exacte et d'une composante heuristique : la composante exacte est obtenue en combinant la contribution des valeurs de fonctions-attributs des décisions qui constituent l'hypothèse; la composante heuristique se veut un estimé optimiste du coût nécessaire pour compléter la traduction, tenant compte notamment de la présence de segments discontinus. Chaque pile fait l'objet d'un filtrage, visant à y éliminer les hypothèses les moins prometteuses. Ce filtrage se fonde à la fois sur la valeur du score et sur le nombre d'hypothèses dans la pile.

On trouve la traduction finale dans la "dernière" pile, c'est-à-dire celle correspondant à une couverture totale de la phrase-source. On récupère alors la traduction ayant le meilleur score, et qui constitue une phrase bien formée, c'est-à-dire sans jokers.

5 Évaluation

Nous avons effectué certaines expériences, visant à évaluer le potentiel de notre approche, et en particulier l'apport des bi-segments discontinus. Toutes nos expériences ont porté sur la traduction du français vers l'anglais. Nous avons utilisé des textes provenant du corpus *Aligned*

Corpus	phrases	mots-source	mot-cible
construction des bi-segments	931 000	17,2M	15,2M
entraînement no. 1	250	3646	3295
entraînement no. 2	250	3793	3441
test no. 1	250	3007	2745
test no. 2	250	3238	2949

Table 1: Caractéristiques des corpus utilisés.

nombre max. de jokers	source	cible	source et cible
0	1 047 101	1 224 910	831 034
1	2 232 226	2 448 223	1 959 154
2	3 403 827	3 403 827	3 403 827

Table 2: Distribution cumulative des bi-segments de B_2, en fonction du nombre maximum de jokers dans la source, la cible et les deux.

Hansards of the 36th Parliament of Canada[2]. De cet ensemble de données, nous avons extrait cinq sous-corpus : un corpus de *construction des bi-segments*, deux corpus d'*entraînement* et deux corpus de *test*. Ces sous-corpus ont été extraits des portions dites *training*, *test-1* et *test-2* des Hansard alignés. Pour des raisons d'efficacité, nous nous sommes limités aux phrases de 30 mots et moins, et à des corpus d'entraînement et de test de petite taille. Le tableau 1 résume les principales caractéristiques des corpus.

Nous avons construit des banques de bi-segments, suivant la méthode présentée à la section 2. Cette méthode génère potentiellement un très grand nombre de bi-segments. Or le temps requis pour le décodage croît de façon essentiellement linéaire avec le nombre de bi-segments disponibles. C'est pourquoi il importe de limiter la taille des banques. Pour ces expériences, nous nous sommes donc limités à la combinaison des ensembles L_1 à L_5, c'est-à-dire obtenu de toutes les combinaisons de 1, 2, 3, 4 ou 5 cepts du corpus de construction. Partant de là, nous avons construit deux banques, qui se différencient par le nombre maximal de jokers admis dans les segments source ou cible : les bi-segments de la banque B_0 ne comportent aucun joker (ce sont donc des bi-segments continus), alors que ceux de la banque B_2 comportent au plus 2 jokers dans la source ou la cible. Dans chacune de ces banques, nous avons exclus les bi-segments n'apparaissant qu'une fois dans le corpus, et pour tout segment-source, nous n'avons retenu que les 20 segments-cible les plus fréquents. La distribution cumulative des bi-segments dans la banque B_2 en fonction du nombre de jokers qu'ils comportent est donnée au Tableau 2.

Nous avons ensuite procédé à l'estimation des paramètres du modèle, suivant la méthode de la section 3 : partant de paramètres aléatoires, nous avons produit les 1000 meilleures traductions pour chacune des phrase des corpus d'entraînement. Nous avons effectués ce processus 3 fois, chaque fois partant de paramètres aléatoires différents, pour chacun des 2 corpus d'entraînement, afin de contrôler la stabilité du processus. Pour chacun des ensembles de données d'entraînement résultants, nous avons alors cherché les valeurs de λ_m maximisant le score NIST lissé, à partir de 100 initialisations aléatoires. Pour chacune des banques de bi-segments B_0 et B_2, nous avons effectué 2 itérations de ce processus; comme on peut le voir à la figure 3, le processus converge rapidement.

Les phrases des corpus de test ont ensuite été traduites avec les paramètres optimisés. Nous

[2]Corpus compilé par Ulrich Germann et distribué par le *USC Information Sciences Institute*

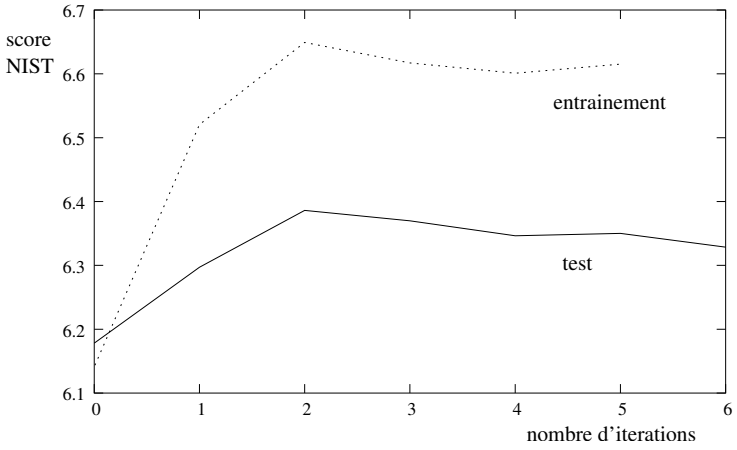

Figure 3: Variation du score NIST en fonction du nombre d'iterations

avons mesuré la qualité des traductions en termes des métriques NIST et BLEU (Papineni et al.2002). À titre de comparaison, nous avons également produit un modèle IBM-4 à partir des données de construction des bi-segments et d'entraînement, à l'aide du système *GIZA++* (Och and Ney2000). Nous avons alors traduit les données de test à l'aide du décodeur *ReWrite* (Germann et al.2001). Les résultats de ces expériences sont rapportés au tableau 3.

Des valeurs supérieures des métriques NIST et BLEU indiquent de meilleures performances; globalement, notre système se comporte donc sensiblement mieux avec la banque B_2 qu'avec B_0, qui produit elle-même des résultats légèrement supérieurs à ceux obtenus avec un modèle IBM-4. Les banques B_0 et B_2 ne diffèrent que par la présence de segments discontinus dans B_2 : c'est donc en allant "piocher" parmi ceux-ci que le modèle arrive à améliorer ses résultats. Ceci semblerait donc supporter notre thèse, que l'utilisation de bi-segments discontinus est bénéfique.

En examinant plus attentivement les traductions produites avec la banque B_2, on constate que les bi-segments discontinus, bien que 3 fois plus nombreux dans la banque que leurs homologues continus, n'ont pas nécessairement la faveur du modèle de traduction. Par exemple, notre système a produit les traductions les plus probables pour les 250 phrases du corpus de test en utilisant 1479 bi-segments, soit 5,92 bi-segments par phrase en moyenne. De ce nombre, seulement 242 sont discontinus, soit moins de 17%, ou 0,96 bi-segment discontinu par phrase. C'est donc dire que dans nombre de situations, notre système préfère encore utiliser des bi-segments continus.

En pratique, les bi-segments discontinus sont utilisés dans des circonstances qui coïncident par-

Corpus	Expérience	*ReWrite*		B_0		B_2	
		NIST	BLEU	NIST	BLEU	NIST	BLEU
test no. 1	1	6,59	0,36	6,63	0,38	6,82	0,39
	2			6,65	0,38	6,83	0,38
	3			6,72	0,38	6,70	0,37
test no. 2	1	6,12	0,31	6,16	0,32	6,20	0,32
	2			6,20	0,32	6,34	0,34
	3			6,14	0,31	6,24	0,32

Table 3: Résultats expérimentaux

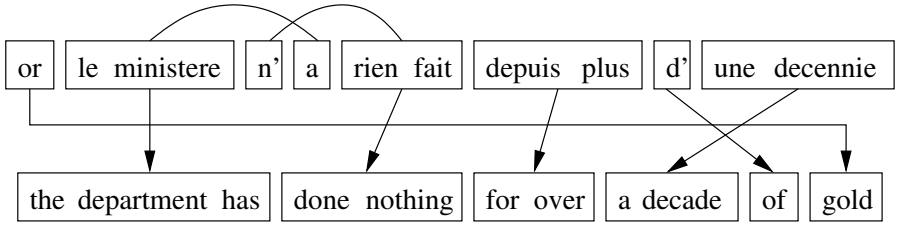

Figure 4: Exemple de traduction avec des bi-segments discontinus

fois avec certains des phénomènes que nous souhaitions voir ainsi traités, mais pas toujours. Et si l'apport des bi-segments discontinus est globalement positif, il reste que ceux-ci introduisent également des problèmes. La figure 4, qui montre un exemple de traduction provenant du corpus de test, tel qu'effectué avec la banque B_2, illustre assez bien la situation. D'une part, on voit comment les bi-segments discontinus permettent de traiter le cas de la négation en français : La combinaison de deux bi-segments $\langle$*Le ministère* $\diamond$ *a*, *the department has*$\rangle$ et $\langle$*n'* $\diamond$ *rien fait*, *done nothing*$\rangle$ permet d'arriver à une traduction assez judicieuse. Par ailleurs, comment expliquer cette mystérieuse apparition en fin de phrase du segment "*of gold*" (en français "*en or*" ou "*d'or*")? D'abord, le système a pris la conjonction de coordination française *or* pour un substantif, qu'il a traduite par *gold*. Il a alors récupéré la préposition *d'*, laissée pour compte dans le bi-segment $\langle$*une décennie*, *a decade*$\rangle$, et s'est servie de sa traduction la plus fréquente (*of*) pour introduire ce nouveau substantif.

De telles erreurs sont assez typiques du comportement de notre système dans son état actuel. Deux facteurs en sont vraisemblablement à l'origine. D'abord, nous n'admettons pas dans notre modèle la possibilité de bi-segments dont l'une ou l'autre partie serait vide, qui permettraient, par exemple, de rendre compte de la "disparition" de la préposition *d'* dans le passage à l'anglais. Mais la méthode d'alignement utilisée pour constituer les banques de bi-segments est également en cause ici. En pratique, on constate que les mots-outils qui ne sont pas explicitement traduits sont souvent mal alignés, entraînant la présence de bi-segments "faussement discontinus" dans la banque, par exemple $\langle$*devons essayer*, *need* $\diamond$ *try*$\rangle$ dans laquelle la préposition anglaise *to* est escamotée, ou encore $\langle$*soins* $\diamond$ *santé*, *health care*$\rangle$, dans laquelle c'est le *de* français qui a disparu. De tels bi-segments, combinés à une absence de traitement des insertions et suppressions, entraînent forcément des erreurs de traduction.

6 Conclusions

Nous avons présenté une approche de la traduction automatique statistique par segments de texte discontinus. Une première implantation de cette approche nous a permis de valider le bien-fondé de notre hypothèse de départ, suivant laquelle ces segments discontinus permettraient de mieux représenter certains phénomènes linguistiques, et ainsi de faire meilleur usage des donnés d'apprentissage.

Dans l'implantation actuelle de notre système, le temps requis pour le décodage est encore souvent prohibitif, ce qui ralentit notamment le cycle d'apprentissage des paramètres. Ceci est d'autant plus critique que certaines expériences semblent indiquer que la qualité des traductions produites par notre système aurait beaucoup à gagner d'un volume plus important de données d'entraînement. Nous examinons présentement différentes stratégies d'optimisation du processus de décodage. Mais le nombre de bi-segments disponibles au moment de la traduction d'une

phrase demeure un facteur dominant de complexité. Le rôle relativement mineur que jouent finalement les bi-segments discontinus dans les traductions optimales suggère qu'on pourrait effectuer une sélection plus judicieuse des bi-segments dès l'étape de construction des banques. Une hypothèse qui nous apparaît prometteuse est celle suivant laquelle les bi-segments qui sont réellement utiles sont ceux qui représentent des traductions de nature non-compositionnelles. La construction des banques pourrait donc incorporer une mesure de compositionnalité, par exemple une variante de l'information mutuelle (Lin1999). Par ailleurs, les bi-segments de nos banques sont relativement petits (en moyenne, moins de 4 mots), lorsqu'on les compare à ceux utilisés dans des systèmes comparables (par exemple, jusqu'à 7 mots dans (Och and Ney2004)). Nous envisageons d'incorporer des segments discontinus beaucoup plus grands qui, plutôt que d'être calculés a priori, proviendraient d'une recherche directe dans le corpus d'entraînement. De tels segments, comparables à des repérages approximatifs ("*fuzzy matches*") dans une mémoire de traduction, joueraient alors le rôle de "phrases à trous" dans le processus de décodage.

Références

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the ARPA Workshop on Human Language Technology*.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast Decoding and Optimal Decoding for Machine Translation. In *Proceedings of ACL'01*, Toulouse, France.

Cyril Goutte, Kenji Yamada, and Eric Gaussier. 2004. Aligning Words Using Matrix Factorisation. In *Proceedings of ACL'04*, pages 503–510.

Philipp Koehn. 2003. *Noun Phrase Translation*. Ph.D. thesis, University of Southern California.

Dekang Lin. 1999. Automatic Identification of Non-compositional Phrases. In *Proceedings of ACL'99*, pages 317–324, College Park, USA, June.

Daniel Marcu and William Wong. 2002. A Phrase-based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of EMNLP'02*, Philadelphia, USA.

F. J. Och and H. Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL'00*, pages 440–447, Hongkong, China, October.

Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proceedings of EMNLP/VLC'99*, College Park, USA.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL'03*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evalution of Machine Translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, USA.

Christoph Tillmann and Fei Xia. 2003. A Phrase-Based Unigram Model for Statistical Machine Translation. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada.

Richard Zens and Hermann Ney. 2003. Improvements in Phrase-Based Statistical Machine Translation. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada.