TALN 2001, Tours, 2-5 juillet 2001

Récupération de segments sous-phrastiques dans une mémoire de traduction

Philippe Langlais et Michel Simard
RALI/DIRO
Université de Montréal
{Felipe, Simardm}@IRO.UMontreal.CA

Résumé - Abstract

L'utilité des outils d'aide à la traduction reposant sur les *mémoires de traduction* est souvent limitée par la nature des segments que celles-ci mettent en correspondance, le plus souvent des phrases entières. Cet article examine le potentiel d'un type de système qui serait en mesure de récupérer la traduction de séquences de mots de longueur arbitraire.

The usefullness of translation support tools based on *translation memories* is often limited by the nature of the text segments that they connect, generally whole sentences. This article examines the potential of a type of system that would be able to recuperate the translation of arbitrary sequences of words.

Mots clés : mémoire de traduction sous-phrastique, traduction assistée par ordinateur, traduction automatique à base d'exemples.

1 Introduction

> L'ensemble des nombres réels contient plus de solutions à plus de problèmes d'arithmétique que toute autre ressource...

Le lecteur averti aura reconnu, sous un déguisement grossier, la célèbre citation de Isabelle et al. (1993): en remplaçant l'arithmétique par la traduction et l'ensemble des nombres réels par celui des traductions existantes, on reconnaît bien l'affirmation qui a servi de leitmotiv à quantité de travaux sur les outils d'aide à la traduction.

La première question que cette boutade soulève est celle que se pose tout écolier aux prises avec un devoir de mathématiques: comment aller chercher ces solutions ? Dans le cas de la traduction, toute une gamme d'approches ont été proposées, allant de la simple consultation interactive (par exemple, avec le système *TransSearch* (Macklovitch et al.2000) [1]) à la traduction automatique statistique et/ou à base d'exemples, en passant par l'extraction automatique de dictionnaires.

Bien sûr, si les solutions sont effectivement présentes dans les traductions existantes, l'approche

[1]TransSearch est disponible via le Web du RALI: `www-rali.iro.umontreal.ca`

la plus attirante est certainement celle qui consiste à aller les y chercher directement. C'est précisément cette approche qui est préconisée dans la plupart des systèmes dits de "mémoire de traduction" (ci-dessous "SMT"). Dans des systèmes comme ceux commercialisés par les firmes Trados (*Translator's Workbench*), IBM (*Translation Manager/2*), Atril (*Déjà-Vu*) et Star AG (*Transit*), on fait le pari qu'il est possible de retrouver dans un nouveau texte à traduire des phrases complètes qui ont déjà été traduites. En organisant systématiquement la production des traducteurs en paires de phrases traduction l'une de l'autre (la "mémoire de traduction"), il devient alors possible de récupérer ces traductions et de les proposer au traducteur au moment opportun.

Si cette stratégie peut s'avérer productive dans certains contextes bien définis (documents apparentés, révisions, etc.), il faut toutefois reconnaître que dans le cas général, il est bien rare de se retrouver devant de telles répétitions de phrases entières. C'est pourquoi les promoteurs de cette solution tentent de mettre en oeuvre différents mécanismes d'appariement approximatif des phrases (*fuzzy matching*), dans l'espoir de retrouver dans la mémoire de traduction des phrases qui "ressemblent" à celles du nouveau texte-source. Malgré tout, la portée de ces mécanismes est encore très limitée, et par conséquent bride l'applicabilité même de ces systèmes. C'est du moins la conclusion à laquelle semblent mener directement des études récentes comme celle de Planas et Furuse (1999) et celle de Macklovitch et Russell (2000).

Dans ce dernier article, les auteurs proposent différents moyens de pallier les lacunes des systèmes actuels de gestion des traductions antérieures. Il est notamment question d'analyse morphologique, de l'identification et du traitement modulaire des entités nommées et expressions à caractère numérique et d'analyse de surface (*chunking*). Toutes ces mesures visent à assouplir le mécanisme d'appariement entre le texte à traduire et le contenu de la mémoire de traduction. Avec un peu de recul, il est clair que ces suggestions sont en fait autant de pas dans la direction de la *traduction automatique basée sur les exemples* (TABE). Dans la mesure où l'on voit les SMT comme une forme simplifiée de TABE, il s'agit en fait d'une évolution tout-à-fait naturelle.

Bien que la plupart des travaux en TABE proposent des mécanismes complexes d'appariement entre le texte à traduire et les exemples de la base (voir (Somers1999) pour un survol), certains travaux récents dans ce domaine tendent à démontrer que des procédures d'appariement simples sur des séquences de mots arbitraires du texte-source permettent de récupérer une quantité appréciable de segments de traduction pertinents. On pense notamment aux travaux effectués dans le cadre du projet *PanGloss* (Brown1996). L'idée à la base de ces travaux est intuitivement simple : s'il est peu probable de retrouver dans une mémoire de traduction des phrases complètes d'un texte à traduire, il doit néanmoins être possible de retrouver des sous-segments de ces phrases, à la limite, des mots isolés. Si on arrive d'abord à couvrir la totalité du nouveau texte avec de tels segments, puis à identifier correctement la traduction de ces segments dans la mémoire de traduction, et finalement à agencer ces traductions en un tout cohérent, on obtient une traduction du texte source.

Dans un contexte de traduction entièrement automatique, le succès de chacune de ces étapes est crucial à la réussite de toute l'entreprise. Dans un contexte d'aide à la traduction humaine, où le traducteur à le loisir de sélectionner, de modifier ou encore même de fournir des morceaux de traductions manquants, le succès d'une telle approche se mesure plutôt en terme d'*utilité*: dans quelle mesure les traductions partielles extraites d'une mémoire de traduction sous-phrastique sont-elles susceptibles d'être utiles à un traducteur ? C'est précisément ce que nous tentons d'évaluer dans cette étude.

2 Cadre applicatif

Nous nous intéressons à une gamme d'outils d'aide à la traduction qui, dans l'espoir de faciliter la tâche du traducteur, tentent de récupérer automatiquement dans une mémoire de traduction des segments de texte potentiellement réutilisables.

À l'instar des systèmes de mémoire de traduction (SMT) existants, l'objectif n'est pas de traduire la totalité du texte sans le concours du traducteur. Il ne s'agit donc pas de traduction automatique, et c'est toujours au traducteur humain que revient l'initiative du processus. C'est également lui qui est ultimement responsable du produit fini. Ce mode d'interaction s'avère pertinent dans toutes les situations où une traduction de qualité est requise (Macklovitch and Russell2000).

La tâche du système consiste essentiellement à proposer des éléments de solution ; la manière avec laquelle ces briques traductionnelles sont proposées à l'utilisateur ne fait pas l'objet de cette étude. Nous nous concentrons au contraire sur la description d'un cadre directeur visant à l'exploitation sous-phrastique d'une mémoire de traduction. Nous commençons pour cela par préciser ce terme, puis nous décrivons les trois opérations élémentaires nécessaires à la réalisation d'un SMT, à savoir, la soumission d'une requête, la sélection des couples traductionnels et la formulation d'une réponse.

2.1 Formalisation d'une mémoire de traduction

Une mémoire de traduction M est un ensemble de N couples $\{\langle S_i, T_i\rangle,\ i \in [1, N]\}$ où S_i est un segment source en relation de traduction avec le segment cible T_i.

Aucune contrainte théorique ne régit *a priori* la nature des segments présents dans la mémoire ; mais en pratique dans les SMT existants, ces segments sont des phrases complètes[2]. Le mécanisme privilégié de constitution d'une mémoire dans les systèmes actuels consiste en effet à colliger des paires de phrases à mesure que le traducteur les produit. Alternativement, on peut recourir à l'alignement automatique des phrases d'une collection de traductions déjà disponible (voir Véronis et Langlais (2000) pour une évaluation comparative des méthodes existantes.)

2.2 Soumission d'une requête à une mémoire de traduction

Soit un segment e à traduire. Nous définissons une **requête** comme une sous-séquence de e. Cette définition limite certainement le potentiel applicatif d'un SMT particulier et tout autre mécanisme plus évolué d'interrogation de la base de traduction (requête à trou, mise sous forme canonique, etc.) moyennant des techniques adéquates s'avérerait plus productif.

Nous notons R_e l'ensemble des requêtes sources formulées à partir de la phrase à traduire e, et qui apparaissent au moins une fois dans la mémoire de traduction M. La figure 1 montre sur un exemple repris tout au long de ce texte les requêtes d'au moins deux mots qui sont présentes dans M. Comme l'illustre cette figure, les requêtes soumises sont par nature fortement intersectantes.

Intuitivement, il n'est pas nécessairement fructueux de s'intéresser à toutes les sous-séquences

[2] La notion de phrase étant sujette à variation d'un système à l'autre.

will he / will he table / will he table the / he table / he table the / table the / both firms / the recommendations / the recommendations made / the recommendation made by / recommendations made / recommendations made by / recommendations made by both / recommendations made by both firms / made by / made by both / by both / by both firms /	
Source:	*Will he table the recommendations made by both firms ?*
Cible:	Déposera-t-il les recommandations faites par les deux agences ?

FIG. 1: Les 18 requêtes d'au moins deux mots faites avec succès à la mémoire de traduction pour une phrase source donnée (en anglais ici).

possibles de e. En particulier, il est peu probable que la recherche de simples mots dans la base offre une information plus pertinente que la consultation d'un lexique bilingue. De la même manière, on peut penser qu'il est préférable de ne considérer que des séquences "linguistiquement motivées", par exemple des syntagmes. Cette supposition repose sur l'intuition que des séquences plus ou moins arbitraires de mots risquent de mener à des traductions tout aussi arbitraires, qui ont peu de chances d'être récupérables. À l'opposé, un groupe ayant un statut syntaxique clair (par exemple, un syntagme nominal simple) a de fortes chances d'être traduit par un groupe tout aussi cohérent, facilement réutilisable.

C'est dans ce but que nous avons implanté un filtre (appelé `chunk`) dont le rôle est de sélectionner les requêtes potentiellement intéressantes. Ce filtre requiert un parenthéseur capable de segmenter un texte source en unités sous-phrastiques que par la suite nous appelons **chunks**. Tous les chunks identifiés sont des requêtes sources valides, de même que tout groupement de suites de chunks l'est. En d'autres termes, lorsque ce filtre est activé, nous ne cherchons pas de séquence de mots sources dont le début (resp. la fin) ne coïncide pas avec le début (resp. la fin) d'un chunk (voir la figure 2 pour l'application de ce filtre sur notre exemple courant).

Segmentation:	will [$_{NP}$ he table] [$_{NP}$ the recommendations] [$_{VP}$ made by] [$_{NP}$ both firms] ?
Requêtes:	will he table / he table / the recommendations / the recommendations made by / made by / both firms

FIG. 2: Segmentation du texte source de notre exemple et les six requêtes retenues lorsque le filtre `chunk` est activé.

Description du parenthéseur

Nous avons utilisé les ressources mises à disposition dans le cadre de l'action de recherche partagée organisée lors de la quatrième conférence CoNLL (Computational Natural Language Learning) sur le thème de la segmentation (Sang and Buchholz2000). La tâche consistait à étiqueter des phrases d'un corpus de test à l'aide d'un jeu restreint d'une vingtaine d'étiquettes permettant de représenter des découpages simples de nature pseudo-syntaxiques.

Ces étiquettes sont de trois types: les étiquettes identifiant un début de groupe (nominal, B-NP, verbal, B-VP, adjectival, B-ADJ, adverbial, B-ADV, ou autre), de non début de groupe (I-NP, I-VP, etc.) ; une dernière étiquette (O) étant généralement associée à des symboles de ponctuation n'appartenant à aucun groupe précis.

Les participants disposaient pour mettre au point leur système d'un corpus d'entraînement d'environ 9000 phrases extraites du Wall Street Journal et étiquetées avec ce jeu d'étiquettes. La figure 3 reporte un exemple d'étiquetage d'une de ces phrases ainsi que la segmentation induite.

Étiquetage: He$_{B-NP}$ reckons$_{B-VP}$ the$_{B-NP}$ current$_{I-NP}$ account$_{I-NP}$ deficit$_{I-NP}$ will$_{B-VP}$ narrow$_{I-VP}$ to$_{B-PP}$ only$_{B-NP}$ \$ $_{I-NP}$ 1.8$_{I-NP}$ billion$_{I-NP}$ in$_{B-PP}$ September$_{B-NP}$

Segmentation induite: [$_{NP}$ He] [$_{VP}$ reckons] [$_{NP}$ the current account deficit] [$_{VP}$ will narrow] [$_{PP}$ to] [$_{NP}$ only \$ 1.8 billion] [$_{PP}$ in] [$_{NP}$ September]

FIG. 3: Illustration du processus de segmentation par étiquetage. L'exemple est tiré de Sang et Buchholz (2000).

Le parenthéseur que nous avons développé dans cette étude repose sur un étiqueteur markovien simple où w_1^n désigne les mots de la phrase à étiqueter et t_1^n désigne une séquence d'étiquettes associée:

$$p(t_1^n|w_1^n) = \prod_{i=1}^{n} p(t_i|t_{i-1})p(w_i|t_i)$$

Les paramètres de chacune des distributions $p(t_i|t_{i-1})$ et $p(w_i|t_i)$ ont été estimés par l'algorithme itératif EM sur le corpus d'entraînement disponible. L'algorithme de Viterbi a été utilisé pour trouver — pour un segment à étiqueter donné w_1^n — la séquence d'étiquettes qui maximise la quantité $p(t_1^n|w_1^n)$. Les détails de cet étiqueteur sont décrits dans (Langlais2001).

2.3 Sélection de couples traductionnels

Certaines requêtes apparaissent fréquemment dans une mémoire de traduction. Par exemple la requête "*this basis*" apparaît 3576 fois dans la mémoire exploitée par *TransSearch*. Il n'est pas envisageable — pour le type d'application visé — de submerger l'utilisateur avec la totalité des segments cibles appariés dans la mémoire avec une requête source donnée (à fortiori dans le cas d'un ensemble R_e de requêtes). Au contraire, la viabilité d'un SMT repose grandement sur la précision avec laquelle on fournit à l'utilisateur l'information utile et juste celle-ci (nous reviendrons sur ce point dans la section 4). Un premier filtre simple, nommément `select-N`, consiste à ne conserver qu'un nombre restreint N de réponses faites à une requête donnée.

Nous appuyant sur l'hypothèse qu'il est peu probable qu'un utilisateur fasse un usage intensif de blocs traductionnels correspondants à des requêtes qui se chevauchent, nous avons également implanté un second filtre qui restreint le nombre de requêtes de R_e aux seules requêtes qui maximisent — sans chevauchement — la couverture (en terme de mots) de la phrase à traduire. Plus précisément, nous appliquons une méthode de programmation dynamique pour sélectionner parmi les requêtes de R_e celles dont la juxtaposition maximise la couverture du segment source à traduire e tout en favorisant des recouvrements constitués d'un minimum de séquences. Ce filtre (appelé `cover`) s'avère en pratique une bonne manière de filtrer les couples traductionnels. Dans notre exemple, parmi les 18 séquences sources, seules les 3 suivantes sont retenues: [will he table] [the recommendations made] [by both firms].

2.4 Formulation d'une réponse traductionnelle

Un autre type de filtre qu'il est souhaitable d'appliquer afin de réduire la quantité de matériel cible proposé à l'utilisateur consiste à apparier les couples traductionnels à un niveau sous-phrastique (mots ou autres). Plusieurs méthodes ont été proposées pour localiser à l'intérieur de traductions la sous-séquence cible en relation de traduction avec une séquence source donnée. Certaines ont fait l'objet d'une évaluation comparative, dans le cadre de l'ARC-A2 (Véronis and Langlais2000).

Dans ce travail nous avons implanté une méthode d'alignement de mots basée sur les transducteurs grammaticaux inversés (ITG) proposés par Dekai Wu (1997). En bref, cette procédure prend en entrée une paire de segments et une grammaire spécialisée (spécifiée sous formes de règles probabilistes). Elle tente alors de segmenter de manière récursive et parallèle la paire de segments en identifiant à chaque étape l'alignement le plus probable entre deux sous-séquences. La figure 4 illustre le résultat d'un alignement produit par cette méthode pour un couple traductionnel et une requête donnés.

Séquence source: the recommendations made by	
Couple traductionnel:	
source:	What we find in this bill are things that are directly from **the recommendations made by** these groups.
cible:	Ce qu'on trouve dans ce projet de loi, ce sont des choses qui émanent directement des **recommandations faites par** ces groupes
Séquence cible: recommandations faites par	

FIG. 4: Segment cible identifié par notre procédure d'alignement de mots pour un couple traductionnel et une requête donnés.

3 Mesure de la viabilité d'un SMT sous-phrastique

Nous souhaitons évaluer le potentiel d'un SMT répondant aux contraintes imposées par le cadre décrit dans la section précédente. Afin de rendre cette évaluation indépendante — autant que faire se peut — des différents choix d'implantation ainsi que de la nature des ressources disponibles (engins de traduction, modèles de langues, dictionnaires, etc.), nous avons élaboré un protocole d'évaluation utilisant un bitexte de test B. Un bitexte est défini simplement comme un ensemble de paires de phrases en relation de traduction $B_i = \langle s_i, t_i \rangle$.

L'idée principale consiste à simuler un scénario dans lequel un utilisateur traduit une à une les phrases s_i de B. On mesure alors comment les propositions d'un SMT particulier contribuent à l'aider à produire la traduction oracle t_i. La mesure de la portion de t_i couverte par le matériel cible proposé par un SMT est intuitivement un bon indicateur de la "récupérabilité" du contenu de la mémoire de traduction. Toutefois, cette mesure seule ne suffit pas pour évaluer la viabilité d'un SMT. Bien sûr, plus le système propose de texte à l'utilisateur, plus ce dernier est susceptible d'y trouver matière à produire sa traduction. Mais au delà d'une certaine quantité, l'examen des propositions est fastidieux, rendant le système inutilisable. Ceci suggère naturellement le recours aux notions de **précision** et de **rappel** pour mesurer la pertinence d'un SMT particulier.

Dans le contexte de notre étude, la **précision** mesure la proportion pertinente (comptée en mots) de matériel cible proposé, tandis que le **rappel** indique la proportion (comptée en mots) de la partie cible de B couverte par les propositions du SMT.

Puisque nous ne visons pas dans cette étude une application spécifique, nous sommes contraints à faire des hypothèses sur la stratégie développée par l'utilisateur que nous simulons. Nous admettons tout d'abord que l'utilisateur construit mentalement une traduction (la traduction oracle fournie par B) qu'il ne modifie pas lors de la saisie. Nous admettons ensuite, que lorsque cela est pertinent[3], il fera usage de "copiés/collés" pour exploiter les briques traductionnelles qui lui sont proposées. Lorsque cela n'est pas pertinent, l'utilisateur tape simplement sa traduction. Enfin, lorsque l'utilisateur sélectionne un texte dans une brique traductionelle, on suppose qu'il conserve la partie la plus grande qui correspond à la traduction oracle. La figure 5 montre un exemple de simulation de traduction où la précision du SMT associé est de 6/21 (6 mots seulement parmi les 21 proposés ont été utilisés) et le rappel est de 6/11 (6 mots seulement ont servi à produire la traduction oracle qui en contient 11).

segments sources	briques traductionnelles proposées
will he table	va-t-il déposer / *déposera-t-il* à
the recommendations made by	*recommandations faites par* le / des recommandations faites par le
both firms	les entreprises / deux sociétés

Oracle: Déposera-t-il les recommandations faites par les deux agences ?
Couverture cible: [Déposera-t-il] les [recommendations faites par] les deux agences ?

FIG. 5: Simulation d'une session de traduction pour la paire de phrase de l'exemple 1, lorsque les filtres `select-2` et `cover` sont appliqués. Les portions en italique identifient les parties des propositions que l'utilisateur sélectionne.

4 Expériences

4.1 Mémoire de traduction

La mémoire de traduction que nous avons utilisée ici est constituée de textes des débats parlementaires canadiens ; textes communément regroupés sous le nom Hansard. Notre corpus couvre environ 15 ans de débats (de 1986 à 2000 inclusivement) et totalise près de 100 millions de mots de chaque langue. Le tout a été automatiquement segmenté en phrases, lesquelles ont ensuite été alignées avec le programme *SFIAL* (une version améliorée de la méthode de Simard et al. (1992)), produisant ainsi plus de 5 millions de paires de segments.

4.2 Corpus de test

Nous avons isolé deux bitextes assez différents de nature pour mesurer les taux de précision et de rappel obtenus. Chaque bitexte est constitué de 100 paires de phrases extraites aléatoirement

[3]La pertinence d'un copié/collé est dans cette étude définie par: une brique traductionnelle contient au moins deux mots de la traduction oracle.

de l'un des corpus suivants: *Hansard*, constitué de textes du Hansard non rencontrés dans la mémoire de traduction et *Verne*, le roman "De la terre à la lune" de Jules Verne.

Nos expériences ont été effectuées en prenant respectivement pour langue source et cible l'anglais et le français. Ce choix relève de contraintes pratiques et rien ne s'oppose à priori à l'utilisation d'une autre paire de langues.

4.3 Résultats

Nous avons effectué plusieurs expériences afin d'évaluer l'impact des différents paramètres affectant le comportement du système. Nous ne rapportons ici que les résultats nous semblant les plus intéressants.

4.3.1 Couvertures source et cible

Nous avons d'abord voulu évaluer dans quelle mesure le contenu de notre mémoire de traduction était réutilisable. Nous avons donc mesuré la couverture source et cible de différentes configurations de SMT sur nos bitextes de test. Nous rapportons ici les résultats obtenus avec deux de ces systèmes. Le premier système interroge la mémoire de traduction avec toute sous-séquence source d'au moins deux mots, calcule sur les requêtes ayant obtenue au moins une réponse une couverture source optimale (filtre `cover`) ; au plus les 10 premières réponses de chaque requête de la couverture sont ensuite conservées (filtre `select-10`). Le second système applique de plus le filtre `chunk` pour limiter le nombre initial de requêtes considérées.

La couverture source indique la proportion des mots de la partie source du bitexte de test couverte par les requêtes présentes dans la mémoire de traduction ; tandis que la couverture cible indique la proportion des mots de la traduction oracle que nous pouvons reconstruire par juxtaposition de briques traductionelles fournies par le système testé. Ces deux taux sont reportés en table 1.

Il est intéressant de noter tout d'abord les taux de couverture source qui sont entre 71% et 96% selon le type de bitexte utilisé. Ces taux sont étonnement bons si l'on considère que seules les séquences d'au moins deux mots ont été considérées. De manière non surprenante, cette couverture est d'autant plus élevée que le type de bitexte est proche de ceux disponibles dans la mémoire. À titre de comparaison, Brown (1996) reporte un taux de couverture source de l'ordre de 70%.

Les taux de couverture cible sont eux plus faibles. Le fait d'utiliser une unique traduction oracle pour calculer cette couverture explique partiellement ce résultat. Une évaluation plus réaliste de ce taux pourrait être effectuée à partir de différentes traductions d'un même texte, en suivant par exemple le protocole d'évaluation décrit par Niessen et al. (Niessen et al.2000). Dans le cas de textes proches de ceux constituant la mémoire, nous observons une couverture dépassant les 50%, ce qui en soit est un résultat assez encourageant.

Enfin, notons que si les couvertures (source ou cible) sont toujours plus élevées dans le cas du premier système, le ratio de la couverture cible sur la couverture source, qui constitue une sorte de mesure du rendement moyen d'une requête, est toujours en faveur du filtre `chunk` ; ce qui confirme au moins partiellement le bien fondé des hypothèses formulées dans la section 2.2. Ce résultat est d'autant plus encourageant que les temps de traitement liés à une requête sont non négligeables.

	select-10 + cover			select-10 + cover + chunk		
bitexte	source	cible	ratio	source	cible	ratio
Hansard	95.8%	58.2%	0.6	74.3%	53.7%	0.7
Verne	71.9%	28.4%	0.4	45.9%	22.6%	0.5

TAB. 1: Couvertures source et cible mesurées sous les deux scénarios blind et chunk pour les deux bitextes de test. $ratio$ désigne le rapport de la couverture cible sur la couverture source.

4.3.2 Précision et rappel

Comme nous l'avons précisé dans la section 3, les taux de couverture (source ou cible) ne nous offrent aucun moyen d'apprécier la charge de travail incombant à l'utilisateur qui utiliserait les propositions faites par un SMT particulier. Nous avons à cet effet mesuré les taux de précision et de rappel tels que définis plus haut. Nous reportons en table 2 ces taux sur nos bitextes pour le SMT obtenant les meilleurs résultats.

Il s'agit d'un système qui considère toute suite d'au moins deux mots comme requête, et qui ne propose à l'utilisateur que la meilleure association à chaque requête source de la couverture calculée par le filtre cover. La meilleure association à une requête est celle qui obtient le meilleur score d'alignement par notre procédure d'alignement.

bitexte	précision	rappel
Hansard	37.14	28.09
Verne	22.27	11.27

TAB. 2: Précision et rappel pour les bitextes Hansard et Verne

Cette table indique que, dans le cas du bitexte Hansard, plus d'un tiers des briques traductionnelles proposées sont utilisables pour reconstruire 28% de la traduction oracle. La performance sur le bitexte Verne est moindre, mais montre toutefois que le SMT est serviable pour l'aide à la traduction de textes complètement différents de ceux indexés dans la mémoire de traduction.

Dans une application réelle, il est raisonnable de penser qu'un utilisateur ne doit pas être submergé par des propositions dont la taille excède de beaucoup celle de la traduction. En terme de nos expériences, cela correspond à des taux de précision supérieurs ou égaux aux taux de rappel mesurés ; ce que nous observons en table 2.

5 Conclusion

Notre objectif initial était d'évaluer le potentiel applicatif d'un système de mémoire de traductions capable de proposer à son utilisateur des unités sous-phrastiques. Nous avons pour cela proposé un cadre applicatif ainsi qu'une méthodologie d'évaluation d'un tel système.

Les résultats reportés sur des bitextes de nature variée sont encourageants. Ils démontrent principalement que les SMT commerciaux actuels sous-utilisent grandement le contenu de leur mémoire en se concentrant sur des phrases complètes. Aucune des phrases-sources de notre corpus de test ne se retrouvait telle quelle dans la mémoire de traduction.

Nous sommes cependant encore loin d'un SMT sous-phrastique pleinement opérationnel. Il est tout d'abord certain que notre modèle d'alignement de mots est perfectible. Ensuite, il importe de souligner les restrictions engendrées par le choix que nous avons fait de n'aller chercher dans la mémoire que des sous-séquences apparaissant *verbatim* dans le texte source. Les mécanismes à mettre en œuvre pour assouplir cette définition d'une requête dépasse cependant largement l'étude présentée. Enfin, nous ne faisons aucune mention aux problèmes d'interface usager inhérents à la réalisation d'un véritable système ; aspect qui peut grandement cautionner la viabilité d'un système.

Références

Ralf D. Brown. 1996. Example-Based Machine Translation in the Pangloss System. In *Proceedings of the International Conference on Computational Linguistics (COLING) 1996*, pages 169–174, Copenhagen, Denmark, August.

Pierre Isabelle, Marc Dymetman, George Foster, Jean-Marc Jutras, Elliott Macklovitch, François Perrault, Xiaobo Ren, and Michel Simard. 1993. Translation Analysis and Translation Automation. In *Proceedings of the 5th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Kyoto, Japan.

Philippe Langlais. 2001. Combinaison de modèles markoviens pour la segmentation sous-phrastique de textes. Technical report, RALI.

Elliott Macklovitch and Graham Russell. 2000. What's been Forgotten in Translation Memory. In *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 137–146, Cuernavaca, Mexico.

Elliott Macklovitch, Michel Simard, and Philippe Langlais. 2000. TransSearch: A Free Translation Memory on the World Wide Web. In *Proceedings of the Second International Conference on Language Resources & Evaluation (LREC)*, pages 1201–1208, Athens, Greece.

Sonja Niessen, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for mt research. In *Proceedings of the Second International Conference on Language Resources & Evaluation (LREC)*, pages 39–45, Athens, Greece.

Emmanuel Planas and O. Furuse. 1999. Formalizing Translation Memories. In *Proceedings of Machine Translation Summit VII*, pages 331–339.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL 2000*, pages 127–132, Lisbon, Portugal, september.

Michel Simard, George Foster, and Pierre Isabelle. 1992. Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the 4th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–82, Montréal, Québec.

Harold Somers. 1999. Review Article: Example-based Machine Translation. *Machine Translation*, 14:113–157.

Jean Véronis and Philippe Langlais. 2000. Evaluation of Parallel Text Alignment Systems – The ARCADE Project. In Jean Véronis, editor, *Parallel Text Processing*, Text, Speech and Language Technology. Kluwer Academic Publishers, Dordrecht.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, September.