Vers l'intégration du contexte dans une mémoire de traduction sous-phrastique : détection du domaine de traduction

Fabrizio Gotti[1], Philippe Langlais[1], Claude Coulombe[2]

[1]Université de Montréal, RALI/DIRO
{gottif ; felipe}@iro.umontreal.ca

[2]Lingua Technologies Inc.

Résumé

Nous présentons dans cet article une mémoire de traduction sous-phrastique sensible au domaine de traduction, une première étape vers l'intégration du contexte. Ce système est en mesure de recycler les traductions déjà « vues » par la mémoire, non seulement pour des phrases complètes, mais également pour des sous-séquences contiguës de ces phrases, *via* un aligneur de mots. Les séquences jugées intéressantes sont proposées au traducteur. Nous expliquons également la création d'un utilisateur artificiel, indispensable pour tester les performances du système en l'absence d'intervention humaine. Nous le testons lors de la traduction d'un ensemble disparate de corpus. Ces performances sont exprimées par un ensemble de métriques que nous définissons. Enfin, nous démontrons que la détection automatique du contexte de traduction peut s'avérer bénéfique et prometteuse pour améliorer le fonctionnement d'une telle mémoire, en agissant comme un filtre sur le matériel cible suggéré.

Mots-clés : traduction assistée par ordinateur, mémoire de traduction sous-phrastique, récupération sensible au contexte, détection du domaine de traduction.

Abstract

In this article, we present a sub-sentential translation memory sensitive to the translation topic, a first step towards a full-fledged context-sensitive memory. This system is able to recycle previous translations indexed into the memory, not only for full sentences, but also for contiguous subsegments of these sentences, through word alignment information. Interesting segments are proposed to the translator. We also describe the creation of an artificial user (a simulator), necessary to test the system performances when no human intervention is possible, as is the case for these experiments. We test it when translating a set of disparate bilingual corpora. These performances are reflected in different metrics which we define. Finally, we show that a first attempt to automatically detect the translation context can be beneficial and promises to improve such a memory, by acting as a filter on the target material proposed to the user.

Keywords: computer assisted machine translation, sub-sentential translation memory, context-sensitive retrieval, translation topic detection.

1. Introduction

Parmi l'ensemble des outils d'aide à la traduction, le système de mémoire de traduction (SMT) est certainement l'outil le plus populaire auprès des traducteurs professionnels. Comme l'explique (Planas, 2000), ce succès est dû à deux types de redondances que le traducteur rencontre fréquemment dans son activité et qui sont prises en compte de manière naturelle (du

moins en théorie) par une mémoire de traduction. Primo, il est fréquent que le traducteur ait à traduire un texte proche d'un autre texte déjà traduit par le passé (c'est par exemple le cas lorsqu'il traduit une nouvelle version d'un manuel technique). L'auteur parle alors de redondance *inter-document*. Secundo, un même texte peut contenir de nombreux passages répétitifs, un phénomène que (Planas, 2000) qualifie alors de redondance *intra-document*.

Une des rares études où un mécanisme est présenté pour exploiter la redondance intra-document est celle de (Brown, 2005), dans le cadre d'un système de traduction basé sur l'exemple que l'on peut voir comme une extension d'un SMT (Planas et Furuse, 2000). Brown montre comment, en tenant compte des traductions précédentes, il est possible de favoriser certains segments suggérés par le SMT, au détriment d'autres fragments, moins pertinents. Ce cadre de traduction dans lequel l'usager se trouve au moment d'interroger la mémoire est le *contexte*. Cette information contextuelle est donc à même de réduire la quantité de matériel proposé à l'utilisateur. Ceci répond à une faiblesse potentielle des SMT : sans filtre, ils sont en mesure d'inonder le traducteur de fragments, les rendant ainsi inutilisables.

Désireux d'exploiter l'approche contextuelle, le RALI et Lingua Technologies Inc. collaborent actuellement pour développer une mémoire de traduction de troisième génération (MT3G) sensible au contexte, c'est-à-dire un système capable de recycler des traductions au niveau sous-phrastique tout en tenant compte du cadre de traduction où se trouve l'utilisateur. Dans cette étude, nous présentons une tentative d'intégration de cette information contextuelle qui revient à détecter le domaine de traduction (*translation topic*) dans lequel évolue le traducteur, pour augmenter le rendement et l'ergonomie de notre système. Nous considérons cette approche comme un premier pas vers la prise en compte du contexte au sens où l'entend (Brown, 2005).

Nous présentons en section 2 l'architecture de notre SMT et le cadre dans lequel nous le développons. Nous nous intéressons ensuite (section 3) au problème non trivial de l'évaluation d'un tel système ; problème abordé par plusieurs auteurs, notamment par (Simard et Langlais, 2001) et par (Planas, 2000). Nous présentons ensuite en section 4 le mécanisme que nous avons mis en place pour tenir compte du contexte de traduction lors de l'interrogation de la mémoire. Nous montrons que cette approche perfectible donne déjà des améliorations de nature à augmenter la productivité d'un traducteur professionnel. Nous concluons nos travaux en section 5 et dressons la liste des extensions de notre approche sur lesquelles nous travaillons actuellement.

2. La mémoire de traduction sous-phrastique

Une mémoire de traduction sous-phrastique bilingue est une suite de n entrées $\langle S_i, T_i \rangle, i \in [1, n]$, où un segment S_i (typiquement une phrase) est en relation de traduction avec T_i. Un traducteur interroge la mémoire avec une phrase S quelconque et espère une traduction T.

Ce qui en fait une mémoire sous-phrastique ici est que nous conservons également un alignement de mots A_i pour toute paire de phrases $\langle S_i, T_i \rangle$. L'alignement A_i met en relation les mots de la phrase source S_i avec les mots de T_i. Ceci permet au système de faire correspondre à des sous-séquences d'une phrase S_i une partie de T_i, augmentant ainsi l'utilité de la mémoire pour le traducteur, comme l'ont montré (Planas et Furuse, 2000) ainsi que (Langlais et Simard, 2003).

2.1. Mécanisme d'interrogation de la mémoire sous-phrastique

Dans notre système, si on ne peut repérer une phrase complète S_i dans la mémoire, on interroge la mémoire avec des sous-séquences de la phrase. Ceci se déroule en quatre étapes.

1. S_i est d'abord tronçonnée en plusieurs fragments f. Même si une étude précédente (Gotti *et al.*, 2005) montrait que la segmentation de S_i en sous-unités linguistiquement motivées était avantageuse, nous avons préféré, pour des raisons purement techniques, considérer ici toutes les sous-séquence de S_i de deux mots ou plus. Ce compte de mots exclut les mots fonctionnels (« de », « la », etc.).
2. On interroge ensuite la mémoire avec ces fragments. Tous les fragments qui sont introuvables dans la mémoire sont alors éliminés ; les autres sont considérés *valides*.
3. On calcule une *couverture optimale* de S_i avec tous les fragments source valides, c'est-à-dire une couverture qui exclut les chevauchements des fragments. On évite ainsi de proposer plusieurs fois à l'utilisateur du matériel cible en provenance d'une même région de S_i, ce qui restreint le nombre de réponses proposées, en limitant le nombre de fragments. On cherche pour cela à maximiser la couverture de S_i tout en minimisant le nombre de fragments couvrants. Les fragments qui ne font pas partie de cette couverture optimale sont éliminés.
4. On récupère pour chaque fragment f_m les phrases sources où f_m est une sous-séquence, de même que les phrases cibles correspondantes et leurs alignements de mots respectifs. On peut dès lors récupérer à l'aide des alignements de mots la liste des k fragments correspondants $\mathcal{G}_m = \{g_{m_1}, \ldots, g_{m_k}\}$ dans le matériel cible. Cette liste est ordonnée en ordre décroissant de la fréquence des fragments g_{m_j}. Ce tri permet de proposer d'abord à l'usager les fragments les plus intéressants, parce qu'ils sont plus souvent en cooccurrence avec le fragment source f_m.

L'identification de la traduction d'un fragment dans la phrase cible T_i (*translation spotting*) se fait à partir des alignements de mots *via* une technique similaire à la stratégie *expansion* décrite dans (Simard, 2003b). La traduction ainsi repérée est une séquence contiguë de T_i.

2.2. Mise en œuvre du système

Corpus utilisés Typiquement, l'utilisateur d'une mémoire de traduction est appelé à traduire des textes de sources et de sujets (domaines de traduction) disparates. Par conséquent, il est selon nous discutable de peupler un SMT avec un seul corpus, comme c'est par exemple le cas dans (Langlais et Simard, 2003). Nous avons donc peuplé notre mémoire avec neuf corpus distincts, présentés au tableau 1. Chaque corpus est en fait un bitexte, où chaque phrase française a son pendant anglais. Avant leur indexation, ces corpus sont mis en minuscules, une stratégie typique des moteurs de recherche, utilisée pour augmenter le rappel. Le lecteur constatera que ces corpus indexés ont des tailles et des domaines disparates, même si certains d'entre eux ont des thèmes similaires, par exemple ASPC et SANTÉ. Chacun de ces bitextes a un jeu de tests, constitué de 256 phrases prises aléatoirement de la même source. Ces jeux de tests sont bien sûr disjoints des corpus indexés.

L'indexeur et le moteur de recherche Nous avons indexé les corpus présentés au tableau 1 à l'aide de LUCENE, un indexeur et moteur de recherche de texte en JAVA disponible gratuitement dans le cadre du projet Apache (lucene.apache.org). La mémoire se présente donc comme un

Corpus	Description	$\|S\|$	$\|e\|$	$\|f\|$
BIBLE	La Bible (http://www.ibs.org)	28 884	13 906	24 483
CRTC-B	Décisions de radiodiffusion du CRTC (http://www.crtc.gc.ca)	434 162	109 599	113 747
CRTC-T	Décision de télécom du CRTC	188 042	49 010	52 790
CRDP	Décisions juridiques canadiennes	2 060 604	259 800	270 142
EUROPARL	Corpus Europarl (Koehn, 2005)	899 676	92 774	106 433
HANSARD	Débats de la Chambre des communes du Canada 1986-1994	1 751 443	85 778	106 951
SANTÉ	Site web de Santé Canada (http://www.hc-sc.gc.ca)	562 050	105 990	114 179
ASPC	Site web de l'Agence de santé publique du Canada (http://www.phac-aspc.gc.ca)	243 242	69 800	74 783
XEROX	Modes d'emploi bilingues divers de Xerox	51 573	11 475	13 472

Tableau 1. Caractéristiques des bitextes utilisés pour peupler la mémoire de traduction et extraire des corpus de test. $|S|$ désigne le nombre de phrases (paires de phrases) dans le bitexte, $|e|$ est le nombre de types anglais (formes anglaises distinctes) et $|f|$ est le nombre de types français.

grand index contenant neuf sous-index (ou sous-mémoires), un par corpus. L'indexation prend environ 2 h sur un bon PC et occupe 1,4 Go d'espace disque.

Aligneurs de phrases et de mots On effectue deux opérations sur chaque corpus avant son indexation. Le corpus est d'abord aligné au niveau des phrases, de façon à peupler la mémoire avec des paires de phrases en relation de traduction. Cette étape a été réalisée avec le programme JAPA disponible à l'adresse http://rali.iro.umontreal.ca/Japa. Chaque paire de phrases des bitextes est ensuite alignée au niveau des mots à l'aide de modèles statistiques entraînés par la boîte à outils GIZA++ (Och et Ney, 2000). L'alignement produit est stocké dans la mémoire, aux côtés des phrases indexées. Ces deux opérations de prétraitement sont les plus coûteuses en termes de ressources et sont donc effectuées hors-ligne.

3. Évaluation des performances de la mémoire

C'est une chose de bâtir un système de mémoire de traducction, c'en est une autre d'en évaluer les performances. En effet, contrairement aux systèmes de traduction qui, sans intervention humaine, *produisent* une traduction qui peut être comparée à une ou plusieurs références, une mémoire de traduction *assiste* un traducteur humain. Il est donc délicat d'évaluer une traduction sans demander à un traducteur d'agencer, corriger et enrichir les unités proposées par un SMT.

3.1. Simulation de l'usager

Pour évaluer notre SMT, nous tentons de simuler le comportement d'un utilisateur qui serait appelé à traduire les corpus de test, à l'instar de (Langlais et Simard, 2003). Pour chaque phrase source S_i, le simulateur essaie de couvrir les mots d'une traduction de référence T_i en copiant/collant les fragments cibles récupérés par la mémoire (ou des sous-fragments de ceux-ci). Le fonctionnement du simulateur dépend de certains paramètres présentés ici et dont nous déterminons les valeurs optimales à la section 3.3.

Le comportement de cet « utilisateur artificiel » ou simulateur comporte trois étapes, qui suivent celles de la section 2.1, dont nous poursuivons la numérotation. Rappelons que, à ce stade, chacune des sous-séquences sources f_m est associée à un ensemble $\mathcal{G}_m$ de fragments cibles.

5. On filtre les fragments de $\mathcal{G}_m$ en ne conservant que ceux dont la fréquence est supérieure ou égale à un paramètre min_{freq}. Les éléments de $\mathcal{G}_m$ sont triés en ordre décroissant de fréquence.
6. Le simulateur explore, pour chaque fragment f_m, tous les fragments cibles (ensemble $\mathcal{G}_m$) qui lui sont associés. Pour chacun de ces fragments cibles, le simulateur tente de trouver la plus grande sous-séquence commune entre lui et la traduction de référence T_i. Nous appelons $\mathcal{H}_m$ l'ensemble de ces sous-séquences chevauchantes extraites des fragments dans $\mathcal{G}_m$. Ainsi, si un élément de $\mathcal{G}_m$ est `ils défendent la loi 6 maintenant` et que la phrase de référence est `Voilà que les ministrent défendent la loi 9 maintenant.`, la sous-séquence retenue et ajoutée dans $\mathcal{H}_m$ est `défendent la loi`.

   Cette étape est contrôlée par deux paramètres : min_{unit} et max_{nbu}. Le paramètre min_{unit} est la taille minimale autorisée des fragments dans $\mathcal{H}_m$. Pour l'exemple vu plus haut, si $min_{unit} > 3$, la séquence `défendent la loi` ne sera pas ajoutée à $\mathcal{H}_m$ (elle compte 3 mots seulement). Ce paramètre correspond donc au nombre minimal d'unités cibles que le simulateur va « copier-coller » pour construire une traduction. Le paramètre max_{nbu} désigne la cardinalité maximale de $\mathcal{H}_m$. Lorsque l'ensemble $\mathcal{H}_m$ a atteint cette taille maximale, le simulateur cesse de considérer de nouveaux fragments dans $\mathcal{G}_m$, et passe au fragment suivant f_{m+1}.
7. On utilise enfin tous les fragments colligés dans les ensembles $\mathcal{H}$ pour couvrir T_i en évitant tout chevauchement entre les fragments. On cherche pour cela à maximiser le nombre de mots couverts dans T_i tout en minimisant le nombre de fragments couvrants. En cas d'*ex æquo*, on tranche en faveur des fragments les plus fréquents. Nous appelons $\mathcal{O} = \{o_1, \ldots, o_l\}$ l'ensemble des l fragments cibles membres de cette couverture optimale. $\mathcal{O}$ est donc l'ensemble des fragments qui sont sélectionnés automatiquement par le simulateur pour « produire » une traduction.

3.2. Métriques d'évaluation

Nous proposons ici plusieurs métriques pour évaluer la qualité de la mémoire de traduction dans différents environnements lorsque pilotée par le simulateur décrit à la section 3.1. Ceci permet d'ajuster le fonctionnement de la mémoire. Une bonne mémoire de traduction est notamment un compromis entre trois contraintes : le rappel, la précision et le temps d'exécution. En effet, pour être viable, un SMT doit atteindre le délicat point d'équilibre où il sera en mesure de suggérer suffisamment de matériel cible au traducteur pour l'aider, sans pour autant l'inonder de réponses. Les métriques qui suivent nous guident vers un compromis.

Rappel En présence d'une traduction de référence T, le rappel R correspond au ratio des mots de T qui sont couverts par les fragments de $\mathcal{O}$. Cette couverture cible représente la richesse et la justesse de cette mémoire pour l'utilisateur. On s'intéressera également plus brièvement à la couverture source, soit la proportion des mots de la phrase source S qui sont couverts. Elle indique l'adéquation de la mémoire pour traduire un texte donné, sans pour autant exprimer la qualité du matériel retourné.

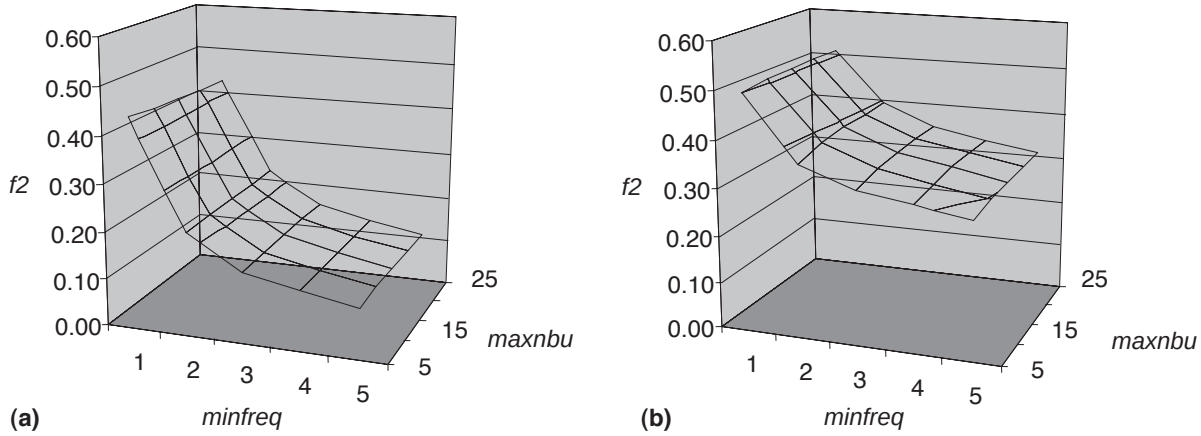

Figure 1. Valeur de la F-mesure $f2$ en fonction des paramètres de filtrage des fragments cibles min_{freq} et max_{nbu}, pour différents corpus de test. ***(a)*** *Sous-mémoire* BIBLE *interrogée avec le corpus* BIBLE. ***(b)*** *Sous-mémoire* HANSARD *interrogée avec le corpus* HANSARD.

Précision La précision traduit l'ergonomie de la mémoire. Il ne suffit pas que la mémoire soit en mesure de fournir des fragments intéressants : encore faut-il que l'usager puisse les repérer sans avoir à considérer trop de matériel cible. Définir une mesure de précision revient à définir quel est ce matériel cible considéré par l'usager et est sujet à diverses interprétations. Nous utilisons deux mesures de précision : $P1$ et $P2$. $P1$ est le nombre de mots de T couverts par les fragments de $\mathcal{O}$ divisé par le nombre total de mots cibles considérés par le simulateur dans les ensembles $\mathcal{G}_m$, lorsqu'il cherchait des sous-séquences communes entre les fragments de $\mathcal{G}_m$ et T (voir section précédente).

$P2$ est le nombre de mots de T couverts par les fragments de $\mathcal{O}$ divisé par le nombre de mots considérés par le simulateur pour construire les fragments présents dans $\mathcal{O}$. Pour l'exemple introduit à la section précédente, si l'unité `défendent la loi` fait partie de $\mathcal{O}$, les mots considérés pour construire cette unité sont `ils défendent la loi 6 maintenant`. Naturellement, $P1 \leq P2$. La métrique $P1$ suppose que le traducteur continue à rechercher des fragments pour des parties déjà traduites de T, tandis que $P2$ présume au contraire que le traducteur fait d'un coup d'oeil la sélection de fragments qui ne se chevauchent pas.

On combine le rappel R et la précision pour obtenir une F-mesure. Puisque l'on a deux précisions, on aura deux F-mesures, $f1$ et $f2$, pour $P1$ et $P2$ respectivement.

Temps Pour mesurer ce dernier, nous avons chronométré le temps que chaque phrase source prend pour être traitée par le simulateur.

3.3. Le système d'évaluation à l'œuvre

Pour tester le simulateur ainsi que la réponse des métriques aux paramètres min_{freq}, max_{nbu}, on a soumis les 256 phrases des corpus de test de BIBLE et HANSARD au simulateur et nous avons interrogé respectivement les sous-mémoires BIBLE et HANSARD. On traduit de l'anglais vers le français. On a mesuré notamment l'impact conjoint de min_{freq} et max_{nbu} sur $f2$. Les résultats sont présentés à la figure 1.

Pour les deux corpus, $f2$ diminue de façon significative lorsque la fréquence minimale tolérée d'un segment cible min_{freq} augmente, ce qui est attendu. Cette tendance est due à une diminution importante du rappel R et à une augmentation plus lente de la précision $P2$. BIBLE est plus sensible à min_{freq} que HANSARD, parce que BIBLE est un corpus plus petit, où les fréquences sont plus faibles. Cette différence explique aussi que $f2$ est toujours plus grande pour HANSARD, qui est plus riche. Aussi, $f2$ est peu sensible à l'augmentation du nombre maximal d'unités considérées max_{nbu}, parce que R augmente au même rythme que $P2$ diminue.

Nos tests montrent que $f1$ réagit de façon très similaire à $f2$. La couverture source n'est naturellement pas affectée par ces paramètres et est de 0,78 % pour BIBLE et de 0,93 % pour HANSARD, ce qui dénote une bonne adéquation du système à la tâche. La plus petite couverture source pour BIBLE est là encore le reflet de sa taille réduite. Cette différence explique le temps moyen d'exécution de 0,20 s pour BIBLE comparé à 2,7 s pour HANSARD.

À la lumière de ces résultats et pour simuler au mieux un traducteur humain, le simulateur utilise $min_{freq} = 1$, $max_{nbu} = 10$, et $min_{unit} = 2$ pour les expériences de contexte.

4. Intégration du contexte

Le traitement indépendant de chacune des phrases n'est pas une solution optimale. (Brown, 2005) montre, dans le cadre d'un système de traduction, la pertinence de favoriser le choix d'exemples aux positions proches (dans le texte desquels ils sont extraits). La traduction d'une phrase dans son système favorise les régions de la mémoire d'où sont extraits les fragments qui ont servi lors des traductions précédentes. Il appelle cet historique le contexte de traduction.

Nous faisons le pari que des unités textuelles supérieures à la phrase (paragraphes, sections, etc.) sont également à même de cibler une fenêtre prometteuse de la mémoire. Dans cette première tentative d'intégration du contexte, la fenêtre identifiée correspond à un sous-index complet (voir section 2.2) et l'unité textuelle est le corpus de test au complet. Cette façon de procéder revient, en fin de compte, à détecter automatiquement le *domaine de traduction* dans lequel se trouve le traducteur, et à adapter la mémoire en conséquence. Cette première approche pourrait néanmoins être aisément modifiée, en réduisant la taille de la fenêtre contextuelle et la taille de l'unité textuelle employée, pour détecter le contexte au sens de l'historique de traduction.

Cette stratégie permet la désambiguïsation de certains mots à traduire. Par exemple, le mot « chambre » peut aussi bien désigner la chambre des communes, dans les débats parlementaires canadiens, qu'une chambre d'hôpital, dans les textes médicaux. Si l'utilisateur traduit un texte sur le milieu hospitalier, notre système ciblera alors un sous-index qui emploie le mot dans son acception médicale.

4.1. Approche « recherche d'information »

Pour tenir compte du domaine de traduction, notre système commence par sélectionner la sous-mémoire la plus similaire au texte à traduire et ne fait ensuite ses requêtes que dans ce sous-index, plutôt que dans l'entièreté de l'index. Le score de similarité entre texte et sous-mémoire est calculé à l'aide de LUCENE et correspond à un score $tf \cdot idf$ classique en recherche d'information. La figure 2 présente ce score pour toutes les combinaisons de corpus et sous-index.

Pour tous les corpus de test, on constate que le score de similarité est maximal lorsque le sous-index et le test ont la même origine, ce qui valide l'approche. On constate également que des

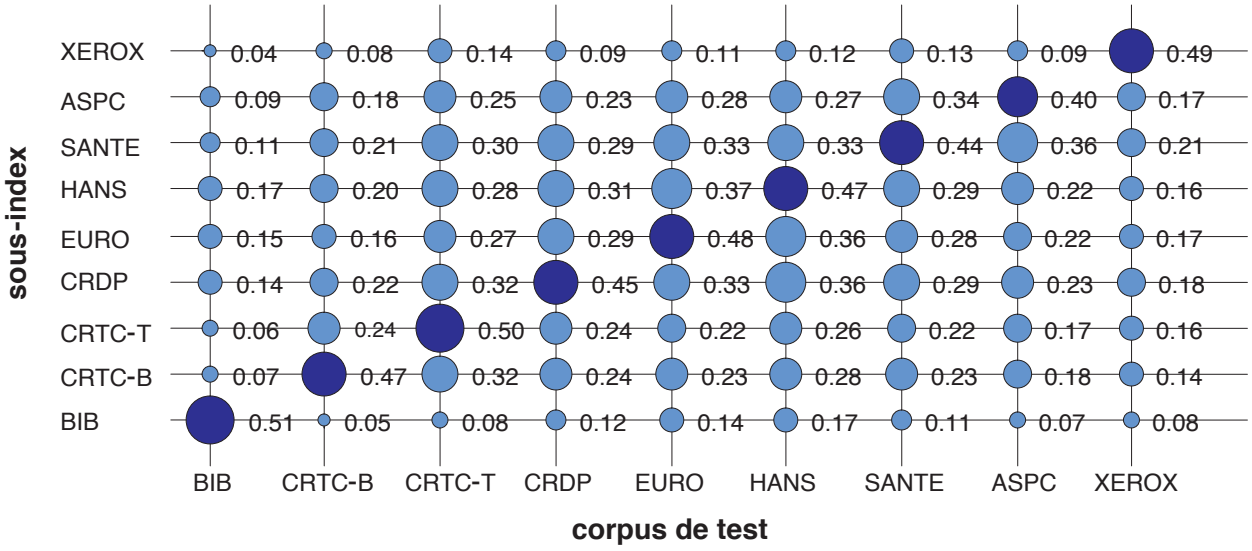

Figure 2. Score de similarité $tf \cdot idf$ entre les neuf corpus de test et chacun des neuf sous-index de la mémoire. La taille des pastilles est proportionnelle à la similarité ; pour chaque corpus, la pastille foncée indique le sous-index le plus similaire

sous-index apparentés (par exemple SANTÉ et ASPC) ont des scores similaires pour tous les corpus. À l'inverse, des corpus peu apparentés comme BIBLE et XEROX ont un score de similarité beaucoup plus faible. Notons également que, plus un sous-index est petit (comme BIBLE), plus sa similarité avec des corpus à traduire non apparentés est faible.

4.2. Résultats du simulateur sur la mémoire sensible au contexte

Pour mesurer l'intérêt de l'intégration du contexte dans la mémoire de traduction, on a demandé au simulateur de traduire chaque corpus de test dans deux conditions : une première fois avec accès à toute la mémoire de traduction, une seconde fois avec accès à la sous-mémoire la plus similaire au texte à traduire, c'est-à-dire, comme on l'a constaté à la section 4.1, la sous-mémoire qui a la même origine que le texte à traduire. On rapporte au tableau 2 des métriques de performance pour chaque test.

Généralement, on constate que la métrique $f2$ est sensiblement plus élevée (à l'exception de ASPC) lorsque l'on considère le contexte pour limiter la taille de la mémoire à explorer. Cette augmentation de $f2$ est due à une augmentation rapide de $P2$ parallèlement à une plus lente diminution du rappel R. Le simulateur a donc moins de matériel à explorer, sans pour autant que la couverture cible en soit affectée. On note les mêmes gains pour $f1$ (non présenté). Le temps moyen de traitement d'une phrase diminue également de façon très importante, ce qui est tout naturel puisque l'on restreint la mémoire à explorer et est d'un intérêt pratique. Notons néanmoins que, si le temps de réponse moyen est de l'ordre de 1 s pour la plupart des phrases sources, ce temps passe à 3 ou 4 s pour l'interrogation de certaines sous-mémoires avec certains corpus de test. Ces délais trop grands pourraient demander quelques réglages.

	sans contexte		avec contexte		variation	
corpus	$f2$	temps (s)	$f2$	temps (s)	$\Delta f2$ (%)	Δ temps (%)
BIBLE	0,41	7,33	0,44	0,18	+7,32	-97,54
CRTC-B	0,68	3,20	0,68	1,24	0,00	-61,25
CRTC-T	0,45	1,46	0,46	1,28	+2,22	-12,33
CRDP	0,45	3,28	0,51	4,60	+13,33	+40,24
EUROPARL	0,41	10,64	0,41	2,63	0,00	-75,28
HANSARD	0,47	9,11	0,52	3,98	+10,64	-56,31
SANTÉ	0,41	2,05	0,47	0,54	+14,63	-73,66
ASPC	0,48	1,32	0,47	0,20	-2,08	-84,85
XEROX	0,11	0,81	0,11	0,69	0,00	-14,81
moyenne	0,43	4,36	0,45	1,70	+4,65	-61,01

Tableau 2. Moyennes des F-mesures $f2$ et des temps d'exécution en secondes pour chaque corpus de test de 256 phrases soumis au simulateur lorsque celui-ci a accès à toute la mémoire de traduction (colonne « sans contexte ») et lorsqu'il utilise le contexte pour sélectionner la partie de la mémoire la plus pertinente pour faire ses requêtes (colonnne « avec contexte »)G

5. Discussion

La mémoire de traduction sous-phrastique décrite ici de même que la stratégie que nous utilisons pour retrouver et suggérer du matériel cible réagissent sainement aux tests que nous décrivons. De plus, une première tentative d'intégration du contexte se montre clairement encourageante : le simple fait que cette stratégie permette de réduire significativement le temps moyen d'interrogation de la mémoire sans nuire à la qualité du résultat est très intéressant. De plus, nous observons une augmentation de la précision et de la pertinence du matériel cible retourné, ce qui suggère qu'il est possible de restreindre la quantité de matériel retourné au traducteur sans pour autant omettre les fragments intéressants.

Il n'en reste pas moins que l'intégration du contexte est encore à l'état préliminaire ici ; cette étude est avant tout une exploration du potentiel de l'approche. La taille des fenêtres que le contexte sélectionne dans la mémoire (qui correspondent à des sous-index complets) est discutable : on aurait pu la réduire, par exemple en tenant compte du découpage des textes en paragraphes. On pourrait également ajouter de l'information positionnelle au contexte, de manière à favoriser favoriser les segments venant d'une région de la mémoire qui a déjà permis de traduire certaines phrases.

Ces informations contextuelles pourraient être intégrées dans un score composite de pertinence qui servirait à trier et à filtrer les fragments proposés à l'utilisateur. Ce score pourrait combiner la fréquence des fragments et le score d'alignement de mots qui a servi à les créer. En outre, notre SMT pourrait tenir compte de la taille des segments (en mots) pour respecter la préférence d'un utilisateur pour les unités plus longues. Ces étapes de filtrage, dont l'importance a été soulignée dans l'introduction, ont fait l'objet, entre autres sujets, des travaux de (Simard, 2003a).

Enfin, notons que le simulateur d'usager fonctionne comme on pourrait s'y attendre, si on en juge par les scores de couverture du matériel cible qu'il est capable d'obtenir. Ses réactions aux paramètres de réglage du système sont conformes à l'intuition. Naturellement, ce genre de stratégie ne remplacera jamais l'évaluation du système par le traducteur humain (d'ailleurs à

l'agenda de notre projet). Le simulateur se veut avant tout un auxiliaire lors du développement de celle-ci, et est une approximation du travail du traducteur.

Références

BROWN R. (2005). « Context-sensitive Retrieval for Example-based Translation ». In *2nd Workshop on EBMT of MT-Summit X*. Phuket, Thailand, p. 9–15.

GOTTI F., LANGLAIS P., MACKLOVITCH E., BOURIGAULT D., ROBICHAUD B. et COULOMBE C. (2005). « 3GTM : A Third-Generation Translation Memory ». In *3rd Computational Linguistics in the North-East (CLiNE) Workshop*. Gatineau, Québec.

KOEHN P. (2005). « Europarl : A Parallel Corpus for Statistical Machine Translation ». In *2nd Workshop on EBMT of MT-Summit X*. Phuket, Thailand, p. 79–86.

LANGLAIS P. et SIMARD M. (2003). « De la traduction probabiliste aux mémoires de traduction (ou l'inverse) ». In *TALN*. Batz-sur-Mer, p. 195–204.

OCH F. et NEY H. (2000). « Improved Statistical Alignment Models ». In *Proceedings of ACL*. Hongkong, China, p. 440–447.

PLANAS E. (2000). « Extending Translation Memories ». In *EAMT Workshop, "Harvesting existing resources"*. Ljubljana, Slovenia.

PLANAS E. et FURUSE O. (2000). « Multi-level similar segment matching algorithm for translation memories and Example-based Machine Translation ». In *Proceedings of the 18th conference on Computational linguistics*. Saarbrücken, Germany, p. 621–627.

SIMARD M. (2003a). *Mémoires de traduction sous-phrastiques*. PhD thesis, Université de Montréal.

SIMARD M. (2003b). « Translation Spotting for Translation Memories ». In R. Mihalcea et T. Pedersen (éds.), *HLT-NAACL 2003 Workshop : Building and Using Parallel Texts : Data Driven Machine Translation and Beyond* : Association for Computational Linguistics. Edmonton, Alberta, Canada, p. 65–72.

SIMARD M. et LANGLAIS P. (2001). « Sub-sentential Exploitation of Translation Memories ». In *MT Summit VIII*. Santiago de Compostella, Spain.