TALN 2010, Montréal, 19–23 juillet 2010

Alignement de traductions rares à l'aide de paires de phrases non alignées

Julien Bourdaillet Stéphane Huet Philippe Langlais
RALI - DIRO - Université de Montréal
C.P. 6128, succursale centre-ville
H3C 3J7, Montréal, Québec, Canada
{bourdaij,huetstep,felipe}@iro.umontreal.ca

Résumé. Bien souvent, le sens d'un mot ou d'une expression peut être rendu dans une autre langue par plusieurs traductions. Parmi celles-ci, certaines se révèlent très fréquentes alors que d'autres le sont beaucoup moins, conformément à une loi zipfienne. La *googlisation* de notre monde n'échappe pas aux mémoires de traduction, qui mettent souvent à mal ou simplement ignorent ces traductions rares qui sont souvent de bonne qualité. Dans cet article, nous nous intéressons à ces traductions rares sous l'angle du repérage de traductions. Nous argumentons qu'elles sont plus difficiles à identifier que les traductions plus fréquentes. Nous décrivons une approche originale qui permet de mieux les identifier en tirant profit de l'alignement au niveau des mots de paires de phrases qui ne sont pas alignées. Nous montrons que cette approche permet d'améliorer l'identification de ces traductions rares.

Abstract. There generally exist numerous ways to translate a word or a phrase in another language. Among these translations, some are very common while others are far less so, according to a zipfian law. As with the rest of the world, translation memories are *googlized*, leading to poorly handled or even simply ignored rare translations, while they are often of good quality. In this paper, we tackle this problem in a transpotting framework. We show that these rare translations are harder to identify than common translations. We describe an original approach based on the word alignment of sentences which are not aligned. We show that this approach significantly improves the identification of those rare translations.

Mots-clés : Traduction automatique statistique, alignement de mots, traduction rares, contrôle de pertinence.

Keywords: Statistical machine translation, word alignment, rare translations, relevance feedback.

1 Introduction

Le repérage de traduction, ou *transpotting* (diminutif de *translation spotting*), consiste à identifier dans du texte cible la traduction d'une requête en langue source (Simard, 2003b; Véronis & Langlais, 2000). Nous appelons *transpot* l'ensemble des mots cibles automatiquement associés à une requête dans une paire de phrases grâce à un algorithme de transpotting.

Dans (Huet *et al.*, 2009a; Bourdaillet *et al.*, 2010), nous avons présenté la nouvelle version du concordancier bilingue `TransSearch`. Grâce à des algorithmes de transpotting à l'état de l'art, celui-ci permet

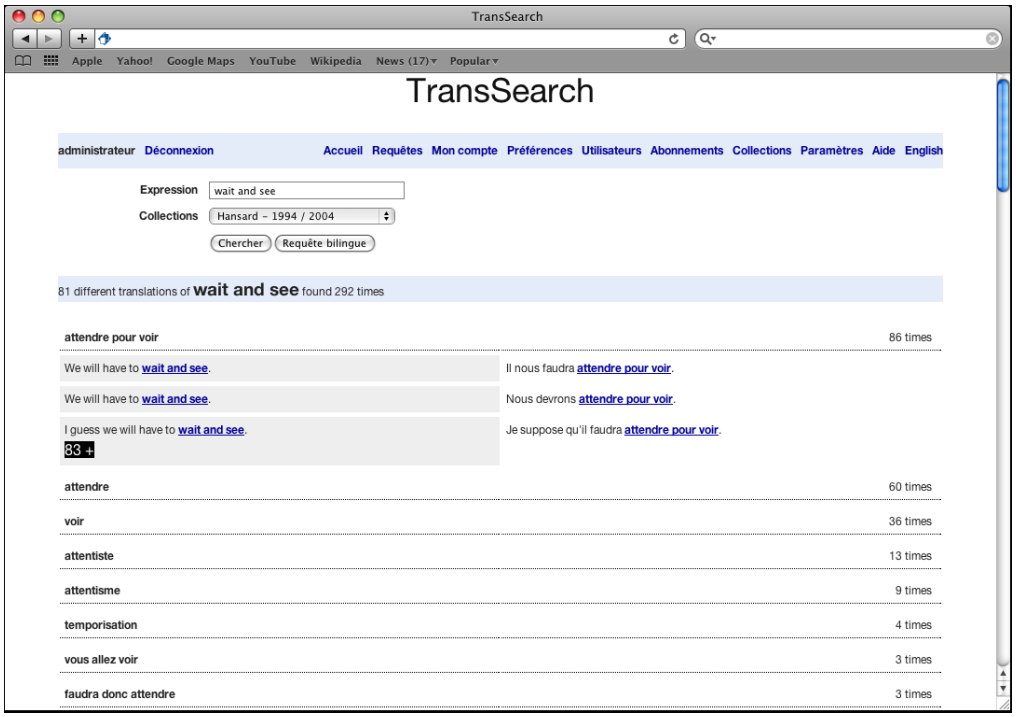

FIG. 1 – Interface utilisateur du concordancier bilingue `TransSearch` suite à la soumission de la requête `wait and see`.

maintenant de répondre à une requête utilisateur libre par un ensemble de traductions. La figure 1 présente l'interface de l'application.

Bien souvent, le sens d'un mot ou d'une expression peut être rendu dans une autre langue par plusieurs traductions. Parmi celles-ci, certaines se révèlent très fréquentes alors que d'autres le sont beaucoup moins, conformément à une loi zipfienne. Par exemple, dans nos travaux nous utilisons le corpus bilingue du Hansard constitué des actes des débats du parlement canadien, soit plus de 8 millions de paires de phrases. On y trouve 1558 paires contenant `meanwhile`, dont 964 contiennent la traduction `pendant ce temps`, 91 la traduction `en attendant`, mais une seule a la traduction `sur ces entrefaites`. De même, la requête `wait and see` apparaît dans 292 paires, soit 86 traductions par `attendre pour voir`, 60 par `attendre`, mais seulement une par `qui vivra verra`. Or dans la version courante de l'application `TransSearch`, ces deux traductions idiomatiques ne sont pas identifiées correctement par le système, privant ainsi l'utilisateur de résultats très intéressants.

Dans cet article, nous nous intéressons à un cas particulier de la tâche de transpotting, à savoir la recherche de traductions rares. On peut considérer une traduction comme rare quand on ne la trouve pas dans un dictionnaire bilingue, ou bien quand son nombre d'occurrences dans un grand corpus parallèle comme le Hansard est très faible, soit — et c'est notre définition ici — égal à une seule occurrence.

Ces traductions rares posent problème aux modèles statistiques d'alignement de mots sur lesquels sont basés les algorithmes de transpotting. En effet, les distributions de transfert sont mal estimées du fait du nombre très restreint de cooccurrences requête/traduction (une seule) dans le bitexte d'entraînement. Dans cet article, nous cherchons à affiner ces distributions pour les traductions rares. Pour cela, nous faisons l'hypothèse que des paires de phrases non alignées, c'est-à-dire qui ne sont pas en relation de traduction,

mais qui néanmoins contiennent la requête et sa traduction rare, peuvent aider à affiner ces distributions.

Dans la suite de cet article, nous présentons en section 2 les différents algorithmes de transpotting que nous utilisons. Nous adaptons en section 3 le concept de contrôle de pertinence (ou *relevance feedback*) populaire en recherche d'information à la tâche de transpotting. Ceci nous permet d'adapter les algorithmes de transpotting à la requête à transpotter. Nous présentons en section 4 l'idée originale sur laquelle notre approche s'appuie pour améliorer la recherche de traductions rares. Nous décrivons en section 5 les gains atteints par notre approche et dressons des perspectives de cette étude en section 6.

2 Transpotting

Le *transpotting* est un problème d'alignement au niveau des mots. Il est donc tout à fait approprié de se baser sur les modèles d'alignement de mots IBM largement utilisés en traduction automatique statistique (Brown *et al.*, 1993).

Formellement, à partir d'une phrase $S = s_1...s_n$ exprimée dans une langue dite source et de sa traduction $T = t_1...t_m$, un alignement $a = a_1...a_m$ de type IBM consiste à connecter chaque mot de T à un mot de S ($a_j \in \{1, ..., n\}$) ou au mot vide ($a_j = 0$), ce dernier rendant compte des mots cibles non traduits. Pour le modèle IBM 2, la probabilité jointe d'une phrase cible et de son alignement, étant donnée la phrase source, est exprimée par :

$$p(t_1^m, a_1^m | s_1^n) = p(m|n) \prod_{j=1}^{m} p(t_j | s_{a_j}) \times p(a_j | j, m, n) \tag{1}$$

où $p(m|n)$ est la distribution de longueur des phrases, le premier terme du produit est la probabilité de transfert et le second la probabilité d'alignement. Avec cette décomposition, il est facile et efficace de calculer l'alignement le plus probable entre deux phrases, $\operatorname{argmax}_{a_1^m} p(a_1^m | t_1^m, s_1^n)$, appelé (par abus de langage) l'alignement de Viterbi, en temps $O(mn)$.

Le modèle HMM est une généralisation du modèle IBM 2 (Vogel *et al.*, 1996). Dans ce cas, la probabilité d'alignement dans l'équation (1) est exprimée par $p(a_j | a_{j-1}, n)$, où une dépendance du premier ordre est introduite en modélisant la probabilité d'alignement du mot cible courant en fonction de l'alignement du mot cible précédent. L'alignement de Viterbi est obtenu par programmation dynamique en temps $O(mn^2)$.

Un algorithme simple de transpotting consiste à calculer l'alignement de Viterbi d'une paire de phrases avec un modèle IBM 2 ou HMM, puis à extraire les mots cibles alignés à la requête source pour sélectionner le transpot recherché. On appelle ces algorithmes `IBM2` et `HMM` respectivement. Cette approche a tendance à produire — le plus souvent à tort — des transpots discontinus. Afin de remédier à cela, Simard (2003b) a proposé l'algorithme suivant ne générant que des transpots continus.

Pour chaque paire $\langle j_1, j_2 \rangle \in [1, m] \times [1, m], j_1 < j_2$, deux alignements de Viterbi sont calculés : l'un entre la suite de mots $t_{j_1}^{j_2}$ et la requête $s_{i_1}^{i_2}$, et l'autre entre les mots restants des phrases source $\bar{s}_{i_1}^{i_2} \equiv s_1^{i_1-1} s_{i_2+1}^n$ et cible $\bar{t}_{j_1}^{j_2} \equiv t_1^{j_1-1} t_{j_2+1}^m$. Le transpotting revient alors à trouver le transpot $\hat{t}_{\hat{j}_1}^{\hat{j}_2}$ maximisant :

$$\operatorname*{argmax}_{(j_1, j_2)} \left\{ \max_{a_{j_1}^{j_2}} p(a_{j_1}^{j_2} | s_{i_1}^{i_2}, t_{j_1}^{j_2}) \times \max_{\bar{a}_{j_1}^{j_2}} p(\bar{a}_{j_1}^{j_2} | \bar{s}_{i_1}^{i_2}, \bar{t}_{j_1}^{j_2}) \right\} \tag{2}$$

Cet algorithme a une complexité en $O(m^3n)$ en utilisant un modèle IBM2 pour calculer les alignements de Viterbi, et en $O(m^3n^2)$ avec un modèle HMM. Toutefois, certains calculs sont factorisables par programmation dynamique, comme détaillé dans (Bourdaillet *et al.*, 2010), ce qui donne une complexité en $O(mn)$ avec IBM 2 et $O(mn^2)$ avec HMM. On appelle ces algorithmes `C-IBM2` et `C-HMM` respectivement.

Finalement, cet algorithme (avec IBM 2 ou HMM) peut profiter des distributions de transfert obtenues en entraînant les modèles dans les deux directions de traduction (les modèles IBM ne sont pas symétriques). Pour cela, les probabilités de transfert d'un mot cible t étant donné un mot source s sont reformulées en :

$$p_{bi}(t|s) = c(p_{S\to T}(t|s), p_{T\to S}(s|t)) \tag{3}$$

où $c(\cdot)$ est une fonction à spécifier combinant les probabilités de transfert des deux directions de traduction. Cette bidirectionnalisation de l'algorithme ne modifie pas la complexité temporelle. Pour HMM, on appelle cet algorithme `C-HMM-bi`.

Enfin, Callison-Burch *et al.* (2005) ont proposé d'utiliser les modèles à base de segments pour transpotter. La trousse à outils MOSES permet d'entraîner un tel modèle à l'état de l'art. Pour transpotter, on extrait de la table de segments l'ensemble des paires candidates dont le segment source est égal à la requête et dont le segment cible apparaît dans la phrase à transpotter. Afin de garder le meilleur candidat, ces paires sont valuées en combinant leurs scores associés dans la table de segments. On appelle cette méthode `PBM`.

3 Pseudo contrôle de pertinence

Dans sa thèse, Simard (2003a) a été le premier à suggérer que la tâche de transpotting pouvait être assimilée à une tâche de recherche d'information, le transpot étant l'information à extraire de la phrase cible. La technique décrite ci-dessous s'inscrit dans cette continuité et renforce l'analogie entre transpotting et recherche d'information.

Principe. En recherche d'information, le pseudo contrôle de pertinence (en anglais, *pseudo relevance feedback*) est une technique permettant d'affiner les résultats d'une recherche (Croft & Harper, 1979). Dans un premier temps, une requête est soumise au système qui retourne une liste de documents, et dans un second temps, on extrait de cette liste un ensemble de caractéristiques pertinentes permettant d'affiner une seconde recherche.

Dans (Huet *et al.*, 2009b), nous avons proposé une adaptation de cette idée pour améliorer nos techniques de transpotting de la façon suivante. Dans un premier temps, on recherche dans le bitexte indexé la requête req à traduire, ce qui donne une liste de paires de phrases dont la phrase source contient req. Puis l'algorithme de transpotting transpotte chacune de ces paires de phrases. Des phrases cibles transpottées, on extrait la liste $\mathcal{T}_{req}$ des traductions obtenues (dans cette liste, une traduction apparaît autant de fois qu'elle a été transpottée dans la liste de paires de phrases). Ces traductions permettent d'adapter le système à la requête lors de la seconde passe.

Pour cela, on crée un bitexte artificiel $\mathcal{B}_1 = \{\langle req, trad\rangle | trad \in \mathcal{T}_{req}\}$ où chaque paire de pseudo-phrases est composée : de req pour la partie source, et de l'une des ses traductions pour la partie cible. En dénombrant les alignements de mots, on peut alors estimer à partir de $\mathcal{B}_1$ une distribution de transfert à posteriori *locale* à la requête. Cette distribution locale peut ensuite être combinée à la distribution *globale* (résultant de l'entraînement d'un modèle IBM sur tout le bitexte, soit la distribution utilisée lors de la première

passe) de la façon suivante :

$$p_{rf}(t|s) = \begin{cases} \lambda p_{glob}(t|s) + (1-\lambda) p_{loc}(t|s) & \text{si } s \in req \\ p_{glob}(t|s) & \text{sinon} \end{cases} \quad (4)$$

où λ est un paramètre contrôlant la combinaison linéaire. L'interpolation n'a lieu que pour les mots sources de la requête ; en effet, par construction de $\mathcal{B}_1$, la distribution locale est estimée uniquement pour ces mots sources. La distribution de l'équation (4) est utilisée lors de la deuxième phase de transpotting.

Résultats expérimentaux. Dans (Bourdaillet *et al.*, 2010), nous avons testé l'utilisation du contrôle de pertinence décrit ci-dessus afin d'améliorer l'application `TransSearch`. La méthode permet de corriger certains transpots erronés. Par exemple pour la requête `way of life`, les transpots `qualité de vie` et `manière d'être` gagnent quelques rangs dans la distribution des traductions de la requête. Malheureusement, les gains numériques sont peu significatifs et les résultats décevants. Néanmoins, nous pensons que cette idée a un potentiel intéressant et nous avons cherché à l'utiliser pour la tâche qui nous intéresse dans cet article, le repérage de traductions rares.

4 Alignements de paires de phrases non alignées

La technique présentée dans la section 3 repose sur le fait qu'une paire requête/traduction apparaît plusieurs fois dans le bitexte indexé. On espère que les occurrences correctement transpottées vont aider à retranspotter correctement celles qui le furent mal. En effet, si les transpots corrects sont les plus fréquents dans $\mathcal{T}_{req}$, on peut supposer que la masse de probabilité de traduction soit correctement répartie dans la distribution à posteriori locale. Or dans le cas de la recherche de traductions rares, par définition ces traductions n'apparaissent qu'une seule fois dans le bitexte indexé (bien que la requête traduite puisse avoir plusieurs autres traductions). Appliquer la technique de la section 3 à ce cas reviendrait à construire un bitexte $\mathcal{B}_1$ composé d'une seule paire de pseudo-phrases, ce qui serait sans intérêt. Il convient donc de modifier cette méthode pour adapter les algorithmes de transpotting au cas des traductions rares.

Pour cela, nous modifions la façon de générer $\mathcal{T}_{req}$ l'ensemble des transpots associés à une requête req. Soit $\langle S_{rare}, T_{rare} \rangle$ une paire de phrases contenant la traduction rare $trad_{rare}$, avec $req \in S_{rare}$ et $trad_{rare} \in T_{rare}$. Par une simple recherche, on extrait du bitexte indexé l'ensemble $\mathcal{S}_{req}$ des phrases sources contenant req. S_{rare} est nécessairement l'une d'elles, et c'est l'unique traduction de T_{rare}. On crée ensuite un bitexte artificiel $\mathcal{B}_2 = \{\langle S, T_{rare} \rangle | S \in \mathcal{S}_{req}\}$. Aucune de ces paires (sauf une) n'a de sens linguistique, puisqu'elles sont artificielles et que les phrases ne sont pas en relation de traduction ; elles peuvent être considérées comme du bruit. En revanche, il existe une régularité dans ce bruit, à savoir le fait que chacune des paires contient la requête et sa traduction rare, qui elles par définition sont en relation de traduction.

On peut alors transpotter chacune des paires de phrases de $\mathcal{B}_2$ avec un des algorithmes présentés dans la section 2. Les transpots obtenus sont tous des sous-séquences de T_{rare}. Étant donné la façon dont sont construites les paires de $\mathcal{B}_2$, les aligner avec un algorithme basé sur les modèles IBM constitue une utilisation discutable de ceux-ci ne correspondant que partiellement à la fonction optimisée par l'algorithme Espérance-Maximisation durant leur entraînement. On peut donc s'attendre à ce que le transpot identifié dans chaque paire ne soit pas forcément très bon. Toutefois, on peut également espérer qu'en accroissant la taille de $\mathcal{B}_2$ et donc en accroissant les « avis » donnés sur l'identité du transpot rare, on arrive à tirer profit de la régularité au sein de $\mathcal{B}_2$ pour mieux discriminer le transpot rare. L'intuition sous-jacente est

que, dans la mesure où la paire requête/traduction rare est la seule régularité dans le bruit, on peut espérer qu'elle « ressorte » ou émerge de ce bruit.

Finalement, on peut appliquer la technique du contrôle de pertinence de la section 3. Une fois chaque paire de phrases de $\mathcal{B}_2$ transpottée, on extrait l'ensemble $\mathcal{T}_{req}$ des transpots de la requête. On crée le bitexte $\mathcal{B}_1$ afin d'estimer la distribution de traduction à posteriori locale qui est réinjectée dans l'algorithme de transpotting conformément à l'équation (4). Puis, la paire $\langle S_{rare}, T_{rare} \rangle$ est transpottée à nouveau durant la seconde passe, où on espère que le transpot trouvé sera amélioré par rapport à la première passe.

5 Expériences

5.1 Corpus

Nos expériences utilisent le corpus parallèle français-anglais du Hansard, les actes officiels des débats parlementaires canadiens, des années 1986 à 2007. On dispose au total de plus de 8,3 millions de paires de phrases. L'indexation de ce corpus avec le moteur Lucene[1] permet de soumettre des requêtes auxquelles le moteur répond en retournant l'ensemble des paires de phrases contenant la requête.

Pour constituer le corpus de test de requêtes/traductions rares, nous avons procédé de manière semi-automatique, en extrayant automatiquement un ensemble de paires de phrases puis en les validant manuellement. Pour ce faire, à partir des logs des requêtes soumises par les utilisateurs à `TransSearch`, nous avons extrait un sous-ensemble de ces requêtes réelles ayant une entrée dans un dictionnaire de mots et d'expressions bilingues disponible au RALI. Nous avons ensuite interrogé avec ces couples requête/traduction le bitexte indexé afin de compter leur nombre d'occurrences. Finalement, nous avons gardé les couples apparaissant dans une seule paire de phrases dans tout le corpus. Par exemple, la requête `meanwhile` apparaît dans 1558 paires de phrases, mais le couple `meanwhile`/`sur ces entrefaites` n'apparaît que dans une seule paire. Nous avons ainsi retenu environ un millier de paires de phrases.

Nous avons ensuite examiné manuellement chacune de ces paires afin de vérifier que la traduction de référence de la requête était effectivement exacte en contexte. Cette étape permet de filtrer différents cas problématiques : traduction en contexte incorrecte à cause de la polysémie de la requête ou de la traduction ; phrase contenant plusieurs mots sémantiquement proches de la requête, la traduction de référence traduisant un de ces mots et non la requête ; bornes exactes du transpot indécidables même pour un humain.

Au final, nous avons conservé 591 paires de phrases dont nous connaissons la requête à transpotter et la traduction rare associée. De celles-ci, nous avons extrait 115 paires pour constituer un corpus de développement, appelé DEV. Nous gardons le reste des 476 paires pour former le corpus de test, appelé TEST.

5.2 Métriques

Les métriques que nous utilisons pour évaluer la tâche de transpotting sont très simples. Dans le corpus TEST, chaque paire de phrases contient la requête et sa traduction. On souhaite qu'un algorithme de transpotting identifie cette traduction afin de la présenter à un utilisateur qui aurait soumis la requête. Il convient donc de calculer le ratio du nombre de fois où la traduction de référence a été identifiée correctement.

[1] http ://lucene.apache.org

Un second indicateur intéressant donne crédit à un transpot partiellement identifié. En effet, plutôt que d'identifier l'ensemble des mots de la traduction de référence, un algorithme de transpotting peut ne réussir à en identifier qu'un sous-ensemble. Bien qu'un tel résultat soit insatisfaisant, cela indique que l'algorithme a pointé dans la phrase cible la zone de la traduction de référence, mais qu'il subsiste un problème d'identification précise de ses bornes. On peut donc également calculer le ratio du nombre de fois où au moins un mot de la traduction de référence a été identifié.

5.3 Entraînement des modèles statistiques et optimisation des paramètres

Les distributions d'alignement et de transfert des modèles IBM 2 et HMM décrits dans la section 2 sont obtenues grâce à GIZA++. Les modèles sont entraînés sur le bitexte de 8,3 millions de paires de phrases décrit dans la section 5.1 comprenant entre autre DEV et TEST.

Afin d'améliorer très significativement les performances des algorithmes de transpotting, les distributions de transfert des modèles IBM 2 et HMM sont remplacées par celles d'un modèle IBM 4. Pour cela, GIZA++ utilise la séquence de modèles suivante avec 5 itérations pour chaque : 1, 2, HMM, 3 et 4.

La table de segments nécessaires à l'algorithme `PBM` est obtenue en utilisant MOSES sur le même bitexte et avec la même séquence de modèles (mais jusqu'à HMM seulement).

L'équation (3) de combinaison des distributions de transfert des deux directions de traduction nécessite un opérateur de combinaison. Les meilleurs résultats sur DEV ont été obtenus avec la moyenne géométrique.

L'équation (4) combinant les distributions de transfert globale et locale à chaque requête nécessite d'optimiser le paramètre λ contrôlant la combinaison linéaire. Nous l'avons optimisé sur DEV séparément pour chaque algorithme de transpotting. Les valeurs obtenues sont discutées dans la section suivante.

5.4 Résultats

Capacité des algorithmes de transpotting à identifier les traductions rares. La première expérience que nous présentons compare les performances des différents algorithmes de transpotting présentés dans la section 2. Chacun de ces algorithmes doit transpotter les requêtes du corpus TEST ; les résultats sont présentés dans les colonnes 2 et 3 du tableau 1.

Sans surprise, on constate que les algorithmes à base de modèle HMM sont meilleurs que leur alter-ego à base de modèle IBM2, que ceux avec contrainte de contiguïté sont meilleurs que sans, et que le modèle HMM bidirectionnel est meilleur que le monodirectionnel. Dans (Bourdaillet *et al.*, 2010), nous avons constaté des résultats similaires sur un corpus de test beaucoup plus grand d'environ 1,4 million de paires de phrases et non limité à des couples requête/traduction rare.

Au premier abord, le mauvais score de `PBM` peut sembler plus étonnant. En fait il faut garder à l'esprit, d'une part que tous les autres transpotteurs utilisent une table de transfert au niveau des mots IBM 4 (cf. section 5.3), ce qui améliore significativement leurs performances. D'autre part, les couples requête/traduction rare n'ont été vus qu'une seule fois à l'entraînement du modèle à base de segments, et n'ont pas forcément été conservés dans la table de segments par l'heuristique `grow-diag-final` par défaut de MOSES (Koehn *et al.*, 2007). Ceci met en évidence le fait que les modèles à base de segments sont clairement insuffisants pour une recherche de traductions rares et justifie pleinement l'usage des différents

Algorithme	Sans contrôle de pertinence		Avec contrôle de pertinence		
	% réf. trouvée	% 1 mot trouvé	% réf. trouvée	% 1 mot trouvé	λ
PBM	32,5	47,9	-	-	-
IBM2	43,8	58,5	-	-	-
HMM	47,4	63,3	-	-	-
C-IBM2	51,3	70,4	58,4	79,0	0,99
C-HMM	54,3	73,6	60,9	83,2	0,98
C-HMM-bi	**65,6**	**83,9**	**69,7**	**86,5**	0,50

TAB. 1 – Scores des algorithmes de transpotting sans contrôle de pertinence (colonnes 2 et 3) et avec (colonnes 4 et 5). Les colonnes 2 et 4 indiquent le pourcentage de traduction de référence trouvées exactement, et les colonnes 3 et 5 le pourcentage de fois où au moins un mot de la traduction de référence est identifié. La colonne 6 indique le meilleur paramètre λ utilisé pour combiner les distributions de transfert globale et locale, conformément à l'équation (4) du contrôle de pertinence.

algorithmes de transpotting présentés dans la section 2.

Enfin, on remarquera que les pourcentages de fois où l'on trouve au moins un mot de la traduction de référence sont environ 20 points supérieurs à ceux où la traduction est identifiée exactement. Cela indique que les algorithmes identifient souvent la zone de la traduction de référence dans la phrase cible, mais qu'il reste un problème d'identification précise des bornes. C'est un problème de précision où les distributions de transfert permettent généralement d'aligner à la requête les mots qui lui sont associés, mais le rang élevé des mots grammaticaux dans ces distributions pose problème comme le soulignait Moore (2004).

Contrôle de pertinence avec paires de phrases non alignées. La seconde expérience consiste à tester la validité de la méthode de transpotting en deux passes décrite dans la section 4, à savoir le contrôle de pertinence grâce à des paires de phrases non alignées.

Pour ce faire, nous avons considéré les trois meilleurs algorithmes de transpotting relativement aux résultats présentés dans le tableau 1, et appliqué la méthode décrite en section 4. Les colonnes 4 et 5 du tableau 1 donnent les résultats avec 200 paires artificielles. La figure 2 présente les résultats obtenus en fonction du nombre de paires artificielles utilisées.

Au vu des résultats, on constate les gains significatifs obtenus par la méthode, et ce pour les trois algorithmes testés. En pourcentage de références trouvées, on a un gain d'environ 6,5 points pour C-IBM2 et C-HMM, et d'environ 4 points pour C-HMM-bi. Un tel gain de 4 points est du même ordre de grandeur que ceux entre IBM2 et HMM d'une part, ou C-IBM2 et C-HMM d'autre part.

Un point très intéressant est le fait que le gain le plus important est obtenu avec les 10 premières paires de phrases (gain de 3,1 points pour C-HMM-bi). Cela implique qu'un petit nombre de paires de phrases artificielles suffit à adapter la distribution de transfert à la requête. En outre, cela signifie que le coût de la méthode de contrôle de pertinence est raisonnable et qu'elle peut être déployée dans le système commercial TransSearch.

Enfin, la dernière colonne du tableau 1 donne la valeur optimale du paramètre λ utilisé pour combiner les distributions de transfert globale et locale dans l'équation (4) du contrôle de pertinence. Pour les deux modèles monodirectionnels C-IBM2 et C-HMM, on observe que la confiance accordée au modèle local est

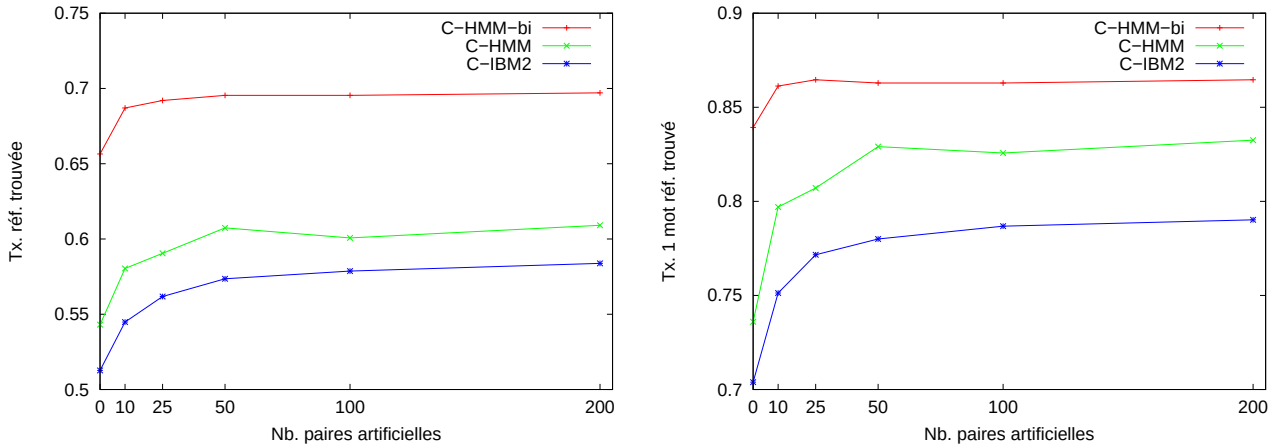

FIG. 2 – Pourcentage de fois où la traduction de référence est trouvée (à gauche) et où au moins un mot de la traduction de référence est trouvé (à droite) en fonction du nombre de paires de phrases non alignées utilisées pour le contrôle de pertinence.

très faible. En effet, nous avons constaté que dès que la valeur de λ descend en-dessous de 0,95 les scores baissent jusqu'à annuler les gains du contrôle de pertinence, avant de dégrader les scores en dessous de $\lambda = 0{,}7$. Au contraire, pour le modèle bidirectionnel `C-HMM-bi` le poids accordé au modèle local est très important. En fait, pour toutes les valeurs de λ que nous avons testées, le contrôle de pertinence améliore significativement les résultats.

Finalement, pour la requête `messy` le contrôle de pertinence avec paires artificielles et `C-HMM-bi` permet ainsi d'obtenir la traduction `fouillis` au lieu de `qui` sans contrôle de pertinence. De même, pour la requête `lead the way`, on obtient `être en tête` au lieu de `tête au`. Enfin, pour la requête `wait and see`, on obtient `qui vivra verra` au lieu de `vivra verra`.

6 Conclusion

Dans cet article, nous avons présenté une méthode de transpotting dédiée aux traductions rares. Celle-ci repose sur le principe du contrôle de pertinence en utilisant des paires de phrases non alignées. Les expériences réalisées montrent des gains significatifs pour le repérage de traductions rares.

Ces gains sont observés en utilisant seulement un petit nombre de paires de phrases non alignées pour chaque requête. Ceci suggère que la méthode présentée ici est industrialisable dans l'application commerciale `TransSearch`. En effet, après la première passe de transpotting, on a, très approximativement, 1/3 des paires de phrases qui ont donné des transpots uniques dans la distribution des traductions de la requête. Ceux-ci sont soit des erreurs des algorithmes de transpotting, soit des traductions rares. On pourrait appliquer le contrôle de pertinence uniquement sur ces paires de phrases afin d'améliorer les résultats. Le coût est fonction de l'algorithme de transpotting utilisé, à un facteur constant près : le nombre de paires non alignées (par requête) utilisées pour le contrôle de pertinence.

Dans l'avenir, nous souhaitons évaluer la méthode décrite dans ce papier pour le transpotting de traductions non rares. En effet, la méthode est suffisamment générale pour qu'on puisse espérer obtenir des gains dans

la plupart des cas. En outre, on peut envisager d'appliquer cette méthode pour une tâche de traduction complète. En effet, le contrôle de pertinence par alignement de paires de phrases non alignées pourrait être utilisé pour valider ou corriger les paires de segments rares extraites par un modèle à base de segments.

Remerciements

Cette étude est financée par le Conseil National de Recherches du Canada, en collaboration avec l'entreprise canadienne Terminotix.[2]

Références

BOURDAILLET J., HUET S., LANGLAIS P. & LAPALME G. (2010). TransSearch : from a bilingual concordancer to a translation finder. *À paraître dans Machine Translation*.

BROWN P., DELLA PIETRA V., DELLA PIETRA S. & MERCER R. (1993). The mathematics of statistical machine translation : parameter estimation. *Computational Linguistics*, **19**(2), 263–311.

CALLISON-BURCH C., BANNARD C. & SCHROEDER J. (2005). A compact data structure for searchable translation memories. In *10th European Conference of the Association for Machine Translation (EAMT)*, p. 59–65, Budapest, Hongrie.

CROFT W. & HARPER D. (1979). Using probabilistic models of information retrieval without relevance information. *Journal of Documentation*, **35**(4), 285–295.

HUET S., BOURDAILLET J. & LANGLAIS P. (2009a). Intégration de l'alignement de mots dans le concordancier bilingue TransSearch. In *16e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, Senlis, France.

HUET S., BOURDAILLET J., LANGLAIS P. & LAPALME G. (2009b). Harnessing the redundant results of translation spotting. In *12th Machine Translation Summit*, Ottawa, Canada.

KOEHN P., HOANG H., BIRCH A., CALLISON-BURCH C., FEDERICO M., BERTOLDI N., COWAN B., SHEN W., MORAN C., ZENS R., DYER C., BOJAR O., CONSTANTIN A. & HERBST E. (2007). Moses : open source toolkit for statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, p. 177–180, Prague, République tchèque.

MOORE R. C. (2004). Improving IBM word alignment model 1. In *42th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 518–525, Barcelone, Espagne.

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

SIMARD M. (2003b). Translation spotting for translation memories. In *HLT-NAACL Workshop on Building and Using Parallel Texts : Data Driven Machine Translation and Beyond*, p. 65–72, Canada.

VÉRONIS J. & LANGLAIS P. (2000). *Evaluation of parallel text alignment systems — The Arcade project.*, chapter 19, p. 369–388. Kluwer Academic Publisher, Dordrecht, Pays-Bas.

VOGEL S., NEY H. & TILLMANN C. (1996). HMM-based word alignment in statistical translation. In *16th Conference on Computational Linguistics*, p. 836–841, Copenhague, Danemark.

[2]http ://www.terminotix.com