L'alignement pour les pauvres : Adapter la bonne métrique pour un algorithme dynamique de dilatation temporelle pour l'alignement sans ressources de corpus bilingues

Kim Gerdes[1]

[1]ILPGA, Sorbonne Nouvelle, Paris 3, LPP (CNRS), Signes (Inria) – Paris 5e – France

Abstract

Bilingual corpora are essential for the construction of bilingual resources just as for any other work in translation studies, but the alignment itself needs bilingual resources or important interventions of bilingual speakers. This article describes work in progress on bilingual text alignment with a dynamic time warping algorithm. These algorithms are the only ones that work without bilingual resources and without operating on the assumption that there are similarities between source and target language (lexical or punctuation cognates): only the signal of the corpus to be aligned is compared with the signals of the words in the target text. We show that the choice of the correct metric used in this comparison is essential for the usefulness of the results. The results can be further enhanced in two ways: by widening the segment to include compound words and other linear collocations, and by including similar words (intra-language cognates à la Levenshtein) in the bilingual word couples, thus including word inflection. The word couples are anchor points for the alignment of the two texts. This approach gives reasonably good results even for language couples like French (rich inflection) and Chinese (isolating language with spaceless writing). A preliminary version, written in Python, C, and Javascript, runs on a web server for high accessibility.

Résumé

Les corpus bilingues alignés sont essentiels dans l'élaboration de ressources bilingues comme dans tout travail traductologique, mais l'alignement nécessite en lui-même des ressources bilingues ou d'importantes interventions de locuteurs bilingues. Cet article décrit un travail en cours sur l'alignement de textes bilingues à l'aide d'un algorithme dynamique de dilatation temporelle. De tels algorithmes sont les seuls à fonctionner sans aucune ressource bilingue et sans se baser sur des similarités entre langue source et cible (cognats lexicaux ou de ponctuation) : seul le signal de chaque mot dans le corpus à aligner est comparé aux signaux des mots de la langue cible. Nous montrons que le choix de la bonne métrique utilisée dans cette comparaison est primordial pour l'utilité des résultats. Il est possible d'améliorer les résultats de deux manières : en élargissant les segments pour inclure des mots composés et d'autres collocations linéaires, et en incluant des mots similaires (cognats intra-langues dégagés à la Levenshtein) dans les couples de mots bilingues pour capter les flexions des mots. Les points d'ancrage ainsi dégagés, en fonction de la similarité des signaux, servent à aligner les textes des deux langues. L'approche donne des résultats satisfaisants même pour des couples de langues comme le français (flexion riche) et le chinois (langue isolante, écriture sans espacement). Pour une accessibilité maximale, l'implémentation, écrite en Python, C et Javascript, tourne, en version préliminaire, sur un serveur web.

Mots-clés : alignement non-assisté, corpus bilingues, algorithme, programmation dynamique, dilatation temporelle (time warping), mesures de distance

1. Introduction

Qu'ont en commun la forme écrite d'un texte et celle de sa traduction ? Le fait que les deux textes aient approximativement le même sens n'est pas visible directement et on ne peut faire que très peu d'affirmations sur un quelconque couple de langues, la syntaxe et le système

d'écriture pouvant varier trop d'une langue à l'autre. La nature discrète et linéaire de toute langue a comme effet qu'on peut malgré tout faire les constats suivants :

Les mots existent, c'est-à-dire que la langue a des segments formés de morphèmes indissociables[1] et les mots se représentent à l'écrit par une forme unique ou par un ensemble fini de formes. En plus, les formes correspondant à un seul mot (les allomorphes) sont souvent des formes graphiquement proches.

Pour un texte et sa traduction, on peut affirmer que certains mots ne se traduisent pas ou se traduisent par des constructions complexes, réparties sur plusieurs mots, mais la plupart des mots sont traduits par des mots (ou des suites de mots contigus). Parmi ces mots ayant une traduction linéairement contrainte, beaucoup de mots, même non ambigus, ont des traductions ambiguës[2]. Mais on peut faire l'hypothèse que dans un texte suffisamment long, on trouvera des mots (ou des groupes de mots) ayant une traduction « facile » dans le sens qu'ils correspondent à des mots (ou des groupes de mots) dans le texte traduit. L'hypothèse centrale exploitée dans cet article est que le caractère linéaire d'un texte fait que ces mots apparaîtront à des endroits similaires dans la traduction. On peut donc considérer comme signal l'apparition des formes dans un texte. Et si le mot a une traduction « facile », alors le signal dans le texte traduit ressemblera au signal source.

Cet article présentera la reconnaissance de ces signaux de mots dans les textes source et cible, certaines améliorations à apporter aux algorithmes connus, la combinaison à des algorithmes de distance de mots pour inclure les signaux groupés des allomorphes et finalement l'utilisation de paires de mots (ou groupes de mots) en tant que points d'ancrage pour l'alignement des paragraphes. Cette méthode nous sert à réaliser un système d'alignement de paragraphes sans paramétrage et surtout sans ressources linguistiques ; un système qui, dans sa version finale, sera accessible en ligne, donnant ainsi l'accès direct à l'alignement à des non-informaticiens et utilisateurs finaux de bitextes.

2. D'autres approches

La plupart des systèmes d'alignement se basent sur certaines formes de similitude graphique entre les textes source et cible. L'alignement par cognats (lexicaux ou de ponctuation) constitue l'approche de référence (voir Simard et al. 1992). L'idée de base des cognats est l'exploitation de la similitude graphique entre un mot et sa traduction : les noms propres mais aussi les mots d'origine gréco-latine s'écrivent de manière très similaire dans différentes langues européennes. Les systèmes basés sur les cognats ont une assez grande fiabilité pour les paires de langues les plus étudiées (les langues européennes) même si la plupart des études sont très spécifiques à une application et à une paire de langue, car les distances entre cognats peuvent varier dans différentes paires de langues et la meilleure définition de la métrique reste sujette à débat (voir par exemple Ribeiro et al. 2001).

Il est évidemment plus difficile de trouver des cognats dans des langues avec des systèmes d'écriture différents. Mais même pour le russe ou le japonais, l'approche reste intéressante comme le montrent Knight & Graehl 1998, car les règles de transcription (par exemple des katakana du japonais) sont assez simples, même s'il faut développer des métriques spécifiques

[1] Voir aussi le vieux débat sur l'existence de morphèmes discontinus : Harris 45.

[2] Autrement dit, si *m* est un mot de la langue 0 et $\{t_0,...t_n\}$ les mots traduisant *m* dans la langue 1, alors l'ensemble des mots traduisant $\{t_0,...t_n\}$ est souvent beaucoup plus grand que $\{m\}$.

pour reconnaître la proximité entre les mots et leurs transcription[3]. Le chinois par contre n'a pas de système de transcription phonétique simple : en effet, pour chaque mot d'origine étrangère, il faut choisir entre une multitude de caractères homophones qui transcrivent de manière satisfaisante les sons étrangers. Pour aboutir à une certaine cohérence dans les traductions, les traducteurs chinois ont recours à d'énormes dictionnaires spécialisés dans la transcription. Pour la plupart des paires des langues, il faut donc des ressources linguistiques considérables pour reconnaître des mots « similaires »[4], l'approche de cognats simple restant un privilège des langues européennes avec leur proximité de vocabulaire et l'uniformisation des systèmes d'écriture.

Les premiers essais d'alignement de corpus bilingues sans ressource linguistique préalable ont été effectués par Brown et al. (1991), Gale & Church (1991) et Kay & Röscheisen (1993). Tous les trois aspirent à un alignement au niveau de la phrase et ils travaillent sur des textes techniques ou des traductions particulièrement littérales (Hansard). Les deux premiers se basent sur la proximité de longueur de formes (mots et phrases), le dernier, déjà plus proche de notre approche, décrit un algorithme de programmation dynamique qui fait des hypothèses sur des paires de mots basées sur la fréquence des mots et améliore pas à pas ces hypothèses en prenant en compte les alignements possibles des phrases contenant ces mots.

L'hypothèse de proximité de longueur de mots, confirmée pour la paire anglais-français, est douteuse ne serait ce que pour la paire allemand-français, beaucoup de noms composés de l'allemand étant traduits par des structures *N de N*. La longueur des phrases aussi est influencée par la structure syntaxique de la langue et pour les langues plus éloignées, on s'attend aussi à trouver plus de différences dans la longueur des phrases entre les traductions. De plus, les symboles utilisés pour indiquer la fin des phrases diffèrent de langue à langue et même le point en langues asiatiques est souvent représenté par un rond avec des encodages informatiques variés.

La séparation du texte en paragraphes semble le seul point commun entre pratiquement tous les textes modernes. Le passage à la ligne est alors le cognat « universel ». Cette présente étude est plus ambitieuse que la plupart des études précédentes en ce qu'elle se veut générale et applicable à toute paire de langues sans ressource linguistique préalable et elle est moins ambitieuse en ce qu'elle ne vise que l'alignement des paragraphes. De plus, nous partons de l'hypothèse (souvent simpliste) que tous les paragraphes seront alignés, aucun ne restera orphelin. La limitation à l'alignement des paragraphes, étant donné que la plupart des études récentes alignent des phrases, des syntagmes et même des mots, requiert une justification. En voilà trois :

- Les systèmes d'alignement actuels sont à peu d'exceptions près spécifiques à une paire de langues tirée elle-même d'un nombre très réduit de paires de langues (anglais vers grande langue européenne, japonais, chinois) et n'aspirent pas à une universalité quelconque.
- L'alignement des paragraphes est le plus grossier juste après l'alignement des chapitres ou sections. Il semble raisonnable d'assumer que dans les traductions, les frontières des paragraphes sont plus souvent respectées que les frontières de phrases,

[3] Le défaut plus grave, en ce qui concerne le japonais, est que seuls des textes faisant fréquemment référence à des mots d'origine étrangère pourraient ainsi être alignés. Pour des textes « purement » japonais, le problème est le même que pour le chinois, car il faut connaître la prononciation des caractères chinois.

[4] Voir Meng et al. 2001 pour l'utilisation de cognats de prononciation chinois-anglais dans l'analyse de la parole.

car les premières correspondent à des unités de sens, les dernières à des unités syntaxiques, par nature plus variables entre les langues.

- L'alignement des paragraphes peut être une base pour d'autres approches plus sophistiquées d'alignement par phrase ou par mot. Parfois les algorithmes d'alignement des phrases se basent explicitement sur un alignement préalable au paragraphe (souvent effectué à la main ou semi-automatiquement) : une fois les paragraphes alignés, nous avons à notre disposition un grand nombre de méthodes pour affiner l'alignement, basées par exemple sur la distribution hypergéométrique des mots dans le texte (voir par exemple Lebart et Salem 1994 ou Zimina 2000).

2.1. Alignement de paragraphes par longueur

La première approche d'alignement de paragraphes d'un texte et de sa traduction, qui sera à la base de notre approche, consiste à trouver le meilleur alignement des marques des paragraphes en se basant sur la longueur des paragraphes en caractères ou en mots. Le découpage en mots n'est pas toujours disponible, et la longueur en caractères variant très fortement entre langues, il faut donc passer aux fractions du texte entier, autrement dit à une normalisation de la longueur des deux textes. Au lieu de traiter les positions des passages à la ligne par l'indice absolu de caractère, on les note en tant que pourcentage sur le texte entier.

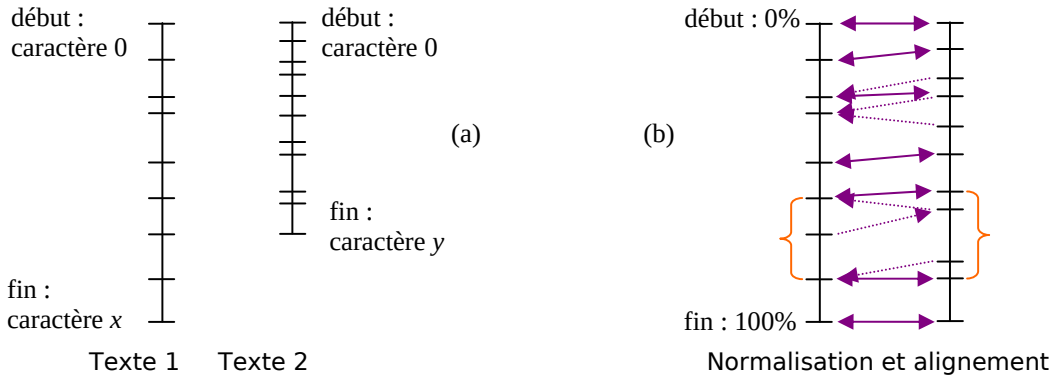


Figure 1 : Marques de paragraphes, normalisation et alignement

La normalisation effectuée, il faut trouver les meilleures paires de marques de paragraphe. Nous indiquons graphiquement le procédé dans la figure 1(b) : Une flèche va de chaque marque de paragraphe à son correspondant le plus proche dans la langue cible. Seules seront prises en compte les flèches bidirectionnelles, autrement dit celles qui correspondent à une paire de marques de paragraphes qui sont mutuellement leurs meilleurs homologues. On peut ainsi aboutir à des multi-correspondances de paragraphes, un exemple d'une correspondance 2-3 étant indiqué avec les accolades dans la figure 1(b).

D'un point de vue computationnel, il s'agit d'un algorithme dynamique standard qui cherche le parcours le plus proche de la diagonale dans un diagramme en treillis. La figure 2 montre à gauche le parcours de l'alignement des marques (0,2,5,6,10)[5] (horizontal) et (0,3,6,10) (vertical). On voit que le parcours aligne les paragraphes du début [0-0][6], ensuite on obtient la correspondance de deux paragraphes avec un : [2,5-3] et finalement [6-6]. Dans le parcours, on ignore les parties horizontales ou verticales, ce qui permet les regroupement multiples

[5] En dixièmes du texte entier pour simplification

[6] Le tiret indique l'association de deux groupes de paragraphes.

comme dans l'exemple (0,1,9,10) – (0,6,10) (dans la figure 2(b)) où on obtient le même type de regroupement de trois paragraphes avec deux paragraphes qu'on a déjà vu dans la figure 1 : [0,1,9-0,6]

Les résultats obtenus avec cet algorithme constituent une amélioration par rapport à un alignement naïf (par exemple la juxtaposition simple de chaque passage à la ligne), mais cette approche est très sensible au bruit et elle marche mieux sur des textes traduits de manière très précise et homogène. Si par exemple la traduction d'un article journalistique contient une entête introductive que l'original n'inclut pas, tous les paragraphes suivants seront décalés et donc mal-alignés. Il semble nécessaire d'ajouter d'autres astuces pour rendre cette approche plus robuste.

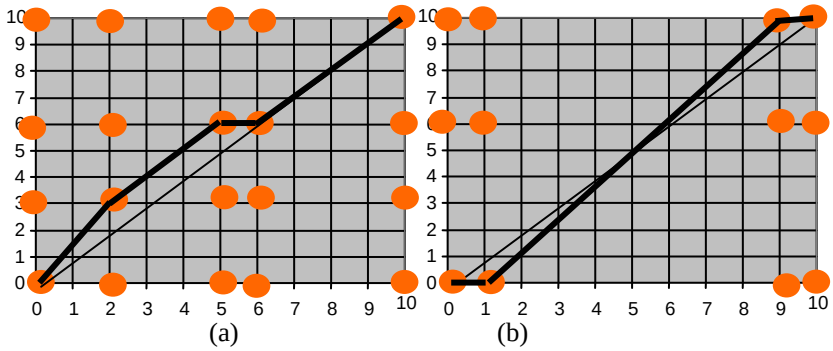


Figure 2 : Treillis pour l'alignement de (0,2,5,6,10) et (0,3,6,10) et de (0,1,9,10) et (0,6,10)

3. Dilatation temporelle

Les algorithmes dynamiques de dilatation temporelle sont aussi basés sur la distribution dans le texte entier d'un mot et de sa traduction, mais contrairement aux marques de paragraphe il faut d'abord déterminer ces paires de mots. L'intuition derrière l'approche de la dilatation temporelle est que le signal du mot ressemble à celui de sa traduction, même si le signal de cette dernière est « déformé » par la traduction : le symbole peut être plus réduit, apparaître plut tôt ou plus tard, manquer certains points, mais il restera « reconnaissable » en tant que traduction du symbole originel.

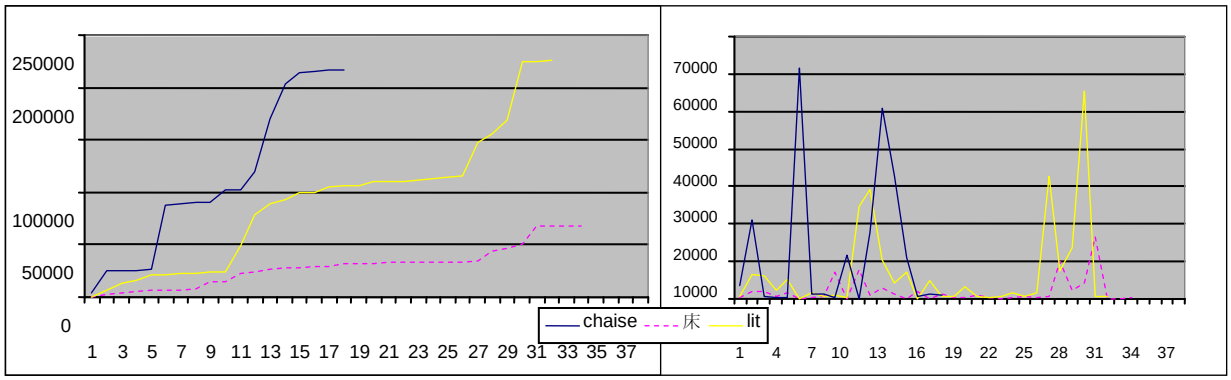


Figure 3 : Les vecteurs d'apparition (gauche) et de récence (droite) pour trois mots

3.1. L'intuition de la dilatation temporelle

Pour illustrer cette intuition, regardons un texte français et sa traduction chinoise : on utilise le premier volume « Aube » du roman *Jean-Christophe* de Romain Rolland (1904-1912, 226981 caractères) et sa traduction par Fu Lei (1957, 68062 caractères)[7]. Si on regarde les points d'apparition (i.e. le numéro de caractère du début) de trois mots : « lit », sa traduction

[7] Gracieusement mis à ma disposition par Jun Miao de l'ESIT, Université Paris 3.

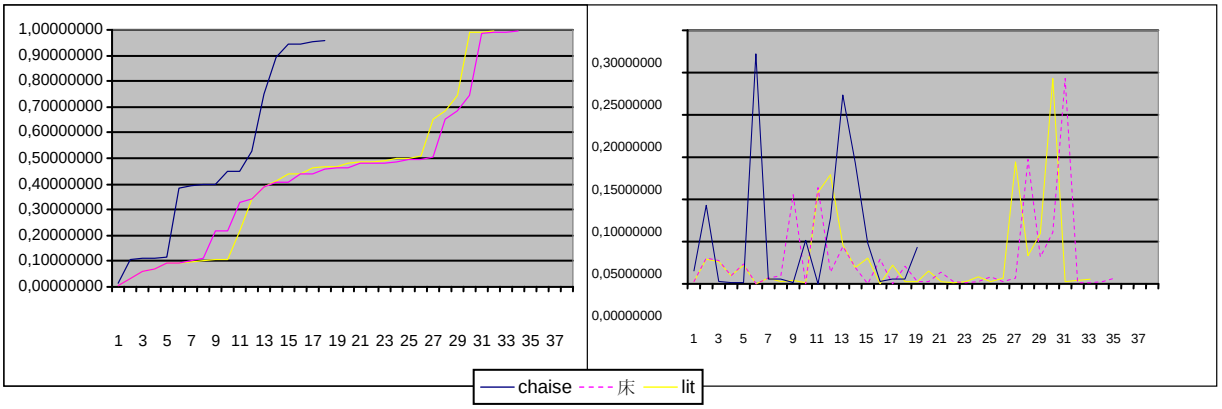


Figure 4 : Apparition et récence de trois mots dans un bitexte normalisé

chinoise « 床 » et « chaise », on obtient trois courbes (figure 3, gauche). Le simple fait que *lit* et *床* apparaissent un nombre de fois similaire (resp. 32 et 34 fois) fait que leurs courbes (claire et pointillée) se ressemblent plus que la courbe sombre de *chaise*. Mais la similitude reste difficile à discerner. Il est préférable de passer à un vecteur de récence : au lieu de représenter les distances de l'apparition du début du texte (les numéros de caractère), on représente la distance (en nombre de caractères) entre chaque apparition. La représentation de ce vecteur fait apparaître beaucoup plus clairement d'une part la similitude entre *lit* et *床*, et d'autre part la différence des deux avec *chaise* (figure 3 droite). Par contre, le fait que le français utilise beaucoup plus de caractères que le chinois apparaît toujours très clairement par la plus grande amplitude des courbes françaises.

Le passage à des fractions du texte entier permet une normalisation de ces courbes. On voit maintenant clairement le lien étroit entre *lit* et *床* par rapport à la *chaise* (diagramme de gauche). Dans le diagramme basé sur les vecteurs de récence (contenant les différences entre les apparitions en fraction de la longueur du texte entier), la similitude des signaux, légèrement décalés par deux apparitions supplémentaires de *床* autour de son 9^{e} et 10^{e} apparition, est apparente (diagramme de droite).

Après un petit résumé de travaux utilisant l'approche de dilatation temporelle, nous déterminerons la métrique utilisée pour mesurer la distance entre ces courbes. Nous expliquerons l'algorithme utilisé pour dégager les couples de mots à l'aide de la similitude de leurs signaux.

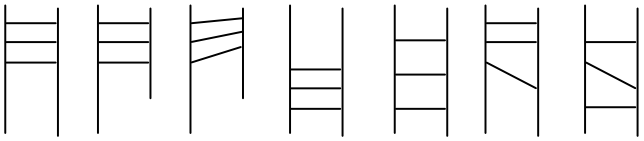
3.2. Utilité de la dilatation temporelle

Les algorithmes dynamiques de dilatation temporelle (*dynamic time warping algorithms, DTW*) cherchent à trouver l'alignement monotone (sans croisement) optimal de deux séquences de longueur variable. L'alignement optimal minimise la distorsion entre les signaux. DTW est utilisé dans des domaines très variés pour la reconnaissance des formes qui peuvent être étirées ou contractées tout en gardant l'information à repérer. On peut citer son utilisation classique dans la reconnaissance vocale (aujourd'hui souvent combinée avec des modèles de Markov cachés) (Jelinek 1997) dans la reconnaissance d'images ou de formes dans des images (où la déformation peut être multidimensionnelle) comme par exemple pour la reconnaissance de signatures ou de visages et encore dans la fouille de données (Ratanamahatana 2004).

En ce qui concerne l'utilisation de DTW pour l'alignement de corpus bilingues, les premiers essais ont été effectués par Fung & McKeown 1994 (voir aussi Somers 1998 pour une comparaison des approches similaires). Fung & McKeown travaillent sur l'alignement

anglais-chinois et montrent que l'algorithme DTW peut trouver des paires de mots qui sont des traductions mutuelles.

Notons que l'algorithme de Fung & McKeown part d'un texte chinois déjà découpé en mots, sans indiquer comment ce découpage a été obtenu, ni sous quelles prémisses linguistiques travaille le découpeur. Cette remarque est importante pour deux raisons : d'une part, leur algorithme n'est donc pas du tout indépendant de ressources linguistiques, bien au contraire : tous les découpeurs du chinois utilisent des lexiques, souvent très étendus pour accomplir cette tâche[8]. D'autre part la notion de « mot » a une influence très importante sur les résultats, parce que leur algorithme utilise directement les mots comme unités alignées. Pour faire l'alignement chinois-allemand par exemple, un découpage du chinois « à l'allemande », c'est-à-dire un système où les noms composés constituent des mots entiers, donnera des résultats bien meilleurs qu'un découpage à l'anglaise, langue germanique également, où les noms composés, structures très similaires aux structures de l'allemand, s'écrivent séparés par des espaces[9]. Sans explication de cette étape de découpage, leurs résultats ne sont donc ni vérifiables ou reproductibles, ni même théoriquement interprétables. Ajoutons que l'alignement de corpus annoncé dans le titre de l'article n'est jamais effectué. En effet, elles trouvent de bonnes paires de mots qui pourraient servir de point d'ancrage pour l'alignement, mais deux points demeurent obscurs : 1. le type d'alignement (aux paragraphes, phrases, mots) pour lequel elles veulent utiliser ces mots, 2. le traitement des paires[10].



3.3. La bonne distance entre des signaux

Le calcul de la distance globale entre deux séquences se base sur la somme des distances locales (entre deux éléments des deux séquences). Il est primordial de trouver la bonne métrique pour la distance locale entre deux valeurs des deux séquences, car les erreurs se multiplient dans le calcul de la somme globale et les séquences longues montreraient à tort une distorsion plus importante (comme c'est le cas dans la métrique de Fung & McKeown 1994 qui se base sur le nombre des mots).

Le graphique A ci-contre montre deux textes de longueur identique (langue 1 à gauche, langue 2 à droite) avec une paire de mots ayant une distribution identique (trois occurrences dans les deux textes à des positions identiques), un très bon candidat donc pour un couple de

[8] Comme l'écriture du chinois ne donne pas d'indices simples sur la fin des mots (contrairement par exemple au japonais où certaines heuristiques simples sur les types de lettre utilisées peuvent nous donner des découpages acceptables), il est naturel de se baser sur des grandes listes de mots. La seule alternative serait de disposer de très grands corpus dans lesquels on chercherait des segments répétés. Mais sans heuristique bien adaptée aux besoins spécifiques de découpage, on risque de trouver des suites de caractères qui ne constituent pas des mots dans le sens souhaité. Une approche similaire sur le français privé d'espaces entre les mots, trouverait par exemple le mot « finde », car « fin » apparaît très souvent suivi de la préposition « de » qui introduit son argument.

[9] Fung & McKeown 1994 donnent elles-mêmes un exemple étonnant : dans leur liste apparaît 一氧化碳 (monoxyde de carbone) deux fois en tant que mot, une fois traduit en tant que « carbone » et une autre fois en tant que « monoxide ». Dans leur corpus, les noms composés ne sont donc pas découpés.

[10] Connaître les paires des mots ne veut pas dire qu'on sache comment aligner les occurrences de ces mots. Comparer avec l'alignement de paragraphes en section 2.1 où on montre comment aligner la « paire » connu de deux passages à la ligne.

mots traduits. Le graphique B montre la même paire de mots dans un alignement où le 2e texte est plus court. Il est clair que le couple est ici un moins bon candidat pour ancrer les textes B que dans la paire A. Le couple de mots montré en C est un meilleur candidat pour les textes de longueurs différentes. Il faut donc encore une fois passer de la position absolue à la fraction du texte. Cette normalisation donnera la même distance aux cas A et C et une plus grande distance au couple présenté en B.

Un deuxième point qui doit être pris en compte dans le design de la métrique est le vecteur de récence utilisé dans le calcul de la distance. A partir du vecteur des positions d'un mot donné (m_1, m_2, m_3, … m_n), Fung & McKeown 1994 calculent le vecteur de récence (m_1, m_2- m_1, m_3- m_2, … m_n- m_{n-1}). Ce vecteur de récence n'est pas symétrique dans le sens qu'il prend en compte la distance entre le début du texte et la première occurrence du mot, mais il ne prend pas en compte la distance entre la dernière occurrence du mot et la fin du texte. Cette asymétrie (du vecteur de récence) ne donne pas dans tous les cas des mauvais résultats. Par exemple, les couples visualisés dans D et E, évidemment d'aussi bons candidats que A, se verront attribuer la même distance que A (égale à 0) bien que les occurrences de leurs mots aient lieu à différents endroits du texte. Par contre, la métrique doit donner à la paire en F la même distance qu'à la paire G. Sans prendre en compte la distance de la dernière occurrence à la fin du texte, la distorsion en F ne sera comptée qu'une fois (en tant que distance entre le 2e et le dernier couple). En G, elle sera comptée deux fois, une fois entre le 1er et le 2e couple, et une fois entre le 2e et le dernier couple de mots. La paire F aura une distance plus petite que G, contrairement à notre intuition sur la structure d'occurrences de traductions dans des textes bilingues.

La métrique utilisée dans Fung & McKeown 1994 ne normalise pas la longueur du texte et leur vecteur de récence ne prend pas en compte la distance entre la dernière paire hypothétique et la fin du texte. Même si notre métrique semble plus intuitive que dans d'autres travaux d'alignement par dilatation temporelle, nous ne pouvons pas directement comparer nos résultats avec ceux de Fung & McKeown 1994. En effet, elles partent d'un texte déjà découpé en mots, tâche exécutée par un algorithme non spécifié. En outre, elles ne montrent qu'un exemple de points d'ancrage trouvés par leur algorithme, basé sur une heuristique non-justifiée (restriction à des fréquences de mots 10 à 300 pour un texte anglais-chinois de 700 ko).

Pour le calcul de notre vecteur de récence, nous utilisons donc un vecteur de positions exprimées par des fractions dans le texte (p_1, p_2, p_3, … p_n). Le vecteur de récence inclut la distance de fin : (p_1, p_2- p_1, p_3- p_2, … p_n- p_{n-1}, **1- p_n**).

3.4. L'algorithme de calcul de distance diluée

Le calcul de la distance de la dilatation temporelle est un algorithme dynamique simple, voici le code en pseudocode :

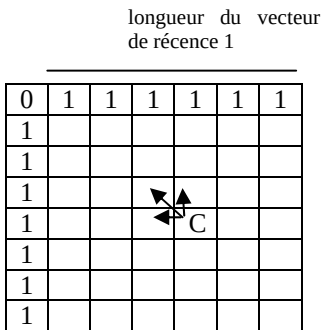
Après avoir calculé les vecteurs de récence pour chaque vecteur de positions des mots, on construit une table croisant les deux vecteurs de récence ainsi qu'une ligne et une

```
timewarp(list1,list2):
        # takes two lists of numbers between 0 and 1
        # and computes a time warp distance
        rec1, rec2 = recency(list1), recency(list2)
        warp[(0,0)] = 0                 # table initiation: corner
        for i=0 to length(rec1) do:
                warp[(i+1,0)] =  1 # table initiation: first line
        for j=0 to length(rec2) do:
                warp[(0,j+1)] =  1 # table initiation: first colon
        for i=0 to length(rec1) do:
                for j=0 to length(rec2) do:
                    warp[(i+1,j+1)] = abs(rec1i-rec2j) +
                    min(warp[(i,j+1)], warp[(i+1,j)], warp[(i,j)])
        return warp[(i+1,j+1)]
```

Figure 5 : Pseudocode de la dilatation temporelle

colonne supplémentaire remplie de 1 (distance maximale) dans la première ligne et dans la première colonne (index 0) à l'exception de la case (0,0) qui contient 0.

Ensuite on remplit la table ligne par ligne : On entre dans chaque case C la distance entre les deux valeurs correspondantes du vecteur de récence auquel on additionne la valeur minimale d'une des trois cases : à gauche, en haut ou en diagonale à gauche en haut de C. Les trois possibilités correspondent à un parcours de la table liant la case C à un de ses voisins, à gauche, en haut ou en diagonal. Le fait qu'on ne puisse chercher ailleurs qu'à gauche et en haut reflète la monotonicité de la dilatation temporelle : on peut tordre mais pas déchirer le signal.



0	1	1	1	1	1	1
1						
1						
1						
1				C		
1						
1						
1						

Une fois les cases remplies, la distance de la paire se trouve en bas à droite dans le tableau : elle correspond au parcours minimal symbolisant l'alignement le moins coûteux des positions des deux mots, comme nous l'avons vu en section 2.1 pour les paragraphes.

Le calcul de la distance présenté ici donne un avantage à des occurrences rares. En effet, dans tous les textes, les mots rares, par exemple les hapax, sont beaucoup plus nombreux que les mots fréquents (loi de Zipf). Les chances qu'il existe des paires d'hapax qui se trouve par hasard dans une position identique (i.e. à la même fraction du texte, par exemple à 47,3% du texte) sont très élevées et ces paires se verront attribuer une distance proche de zéro correspondant à leur distance dans le texte. Par ailleurs, les mots très fréquents ont peu de chance de toujours tomber aux mêmes positions et leurs paires auront toujours une distance non nulle. Cependant, comme ils sont fréquents, la distance entre chaque mot ne peut pas beaucoup augmenter. Dans des essais heuristiques, nous avons voulu donner un avantage aux regroupements de mots fréquents par exemple en divisant la distance par le nombre de couples créés, mais d'emblée, la balance penche en faveur des mots fréquents et on n'a plus de mots rares parmi les candidats. L'utilité de couples de mots fréquents ou rares dépend de la visée finale de l'algorithme. En effet, si on veut aligner les paragraphes, il faut s'intéresser à des mots qui permettent d'aligner beaucoup de paragraphes, donc à des mots qui apparaissent dans un nombre considérable des paragraphe du texte mais pas dans un nombre trop élevé non plus, la plus discriminante des distributions étant proche de la moitié des paragraphes.

4. L'intégration du calcul de la distance dans un système d'alignement

Le système complet fonctionne de manière suivante :

- Préparation : Lecture des deux fichiers à aligner dans la mémoire[11], nettoyage léger (effacement des espaces traînant et des doubles espaces et passages à la ligne), transformation des fichiers en utf-8 en unicode interne, et calcul de la longueur du texte en caractères.

[11] Amélioration possible pour de très grands textes dépassant la mémoire de l'ordinateur en lisant et traitant les données successivement.

- Détermination si lingua continua (langue sans espaces) : si le texte contient moins de 10% d'espaces, traiter la langue en mode continua[12], c'est-à-dire que chaque caractère sera traité comme un mot à part.
- Dans un passage unique sur chacun des deux textes :
  - calcul de tables de hachage utiles : mot → liste d'indices du mot ; mot → liste d'indices des paragraphes où le mot apparaît ; mot → sa fréquence ; index du paragraphe → le texte du paragraphe
  - et certaines autres tables utiles : mot → liste de positions du mot dans le texte exprimées en fraction ; mot → liste de position de paragraphes ; fréquence → liste de mots ayant cette fréquence, …
- Calcul, pour chacun des deux textes
  - des cognats internes à la langue, c'est-à-dire des formes qui se ressemblent, à l'aide d'une distance de Jaro-Winkler (Winkler 1999), une distance de chaîne de caractère basée sur les nombres d'éditions nécessaires à la Levenshtein[13]. L'intuition derrière est que lors de l'alignement d'une langue flexionnelle comme le français, les groupes de mots similaires ressemblent à des allomorphes et devraient donc correspondre à une seule forme (ou encore un groupe de formes) dans la langue cible.
  - des mots ou des groupes de mots « bien distribués », c'est-à-dire des mots ou des groupes de mots qui apparaissent dans un nombre raisonnable de paragraphes, le minimum et le maximum de paragraphes étant paramétrable par l'utilisateur[14].
- Ensuite on itère à travers tous les couples de mots considérés comme intéressant :
  - On applique une heuristique afin de trier les paires qui sont des mauvais candidats (de traduction) avant de calculer la distance DTW : aucun couple dont le nombre d'occurrences du mot 1 est plus que le double du nombre d'occurrence du mot 2.
  - Aux « bons » couples, on applique DTW, expliqué ci-dessus, pour obtenir la distance entre les mots
  - Afin de réduire les besoins en mémoire, nous ne gardons qu'une liste des meilleures paires[15]. Si un des deux mots de la nouvelle paire à insérer apparaît déjà dans la liste avec une distance plus grande, on le rétire de la liste.

[12] En tant qu'exemple, on peut noter que le français contient près de 20% d'espaces en moyenne, l'allemand un peu moins et le chinois moins de 3%. La distinction est assez nette, même si les 10% constituent une heuristique qu'il faudrait vérifier sur d'avantage de langues.

[13] Cette distance donne un avantage aux débuts identiques des mots et elle n'est pas transitive. Pour le texte français utilisé ici, on trouve par exemple le groupe (*garde, gardais, grandes, gardâmes, gardes, garder*) mais aussi le groupe (*gardes, garder, garde*), strictement inclus dans le premier groupe. On constate qu'on pourrait créer des groupes de Jaro-Winkler pour toute valeur de distance, seule la lenteur de ce calcul nous en empêche. Bien qu'on utilise un module python en GPL extrêmement rapide, écrit par David Necas en C, le calcul des groupes de Jaro-Winkler reste couteux.

[14] L'atout principal de ses valeurs est de limiter l'espace de recherche : plus on inclut de mots, plus le calcul devient long, mais *a priori*, d'avantage de mots bien alignés améliorent l'alignement du texte entier. Comme dans la note précédente, sans avoir vérifié sur un nombre suffisant de textes, nous avons l'impression que les limites ne sont pas théoriques mais elles sont posées par la mémoire de l'ordinateur et la patience de l'utilisateur d'obtenir son alignement.

- Une liste obtenue à cette étape peut commencer comme suit : Melchior - 沃, Christophe - 克, Louisa - 莎, ombre - 学, presque - 差, pensée - 甚, lit - 床, …

- Post-traitement. Pour les couples retenus, on essaie d'élargir la fenêtre : on teste si l'ajout d'autres mots (ou d'autres caractères pour une *lingua continua*) adjacents à gauche ou à droite du mot améliore la distance entre les deux mots. Si c'est le cas, on remplace le couple par le couple élargi. Ce processus se poursuit jusqu'au moment où l'élargissement détériore la distance. On obtient ainsi des mots du chinois, surtout les noms propres (par exemple 鲁意莎 *Louisa* à partir d'un des caractères 弗 restants dans la liste de l'alignement). Les mots obtenus correspondent évidemment au découpage en mots dans l'autre langue. On trouve également des structures *N de N* du français alignées avec des noms composés de l'allemand.

4.1. L'alignement des paragraphes

Le premier pas est le même calcul que dans le cas de l'alignement par longueur de paragraphes : on crée un tableau croisé des positions des paragraphes des deux textes et on y entre la distance entre chaque paire de paragraphes. Pour chaque paire dans la liste retenue, on calcule la paire de vecteurs avec les positions des paragraphes contenant les mots de paire. Pour chaque couple possible, on ajoute la valeur de la distance du couple à la valeur déjà dans le tableau en la multipliant avec la valeur déjà contenue dans la cellule. Comme tous les nombres sont plus petits que 1, cela diminue la distance entre les paragraphes concernés et le parcours d'alignement passera plus facilement par la cellule concernée. Pour obtenir l'alignement final, il suffit de trouver le chemin optimal à travers ce tableau.

5. Résultats, projets et conclusion

Pour l'instant ce système est en constante évolution et il a été testé seulement sur un petit nombre de textes bilingues, surtout français-chinois et français-allemand. Les listes de mots retenus pour l'ancrage de l'alignement sont presque toutes correctes, et même les quelques mots qui ne sont pas des traductions mutuelles, ne gênent pas l'alignement, car il s'agit souvent de mots contextuellement proches dans les textes alignés[16]. Sur les textes bilingues testés, le système apporte des améliorations dans l'alignement, mais l'amélioration est difficilement chiffrable car très dépendante du type de texte.

Il est fastidieux de tester systématiquement ce système, car d'une part l'algorithme de dilatation temporelle ne donne des résultats satisfaisants que sur de grands textes (à partir de plusieurs dizaines de milliers de caractères) pour que les « empreintes » des mots deviennent visibles, et d'une part chaque calcul est assez long (souvent plusieurs minutes sur un Pentium 4 à 2,8 ghz avec 1 go de mémoire, même avec des heuristiques limitant l'espace de la recherche). D'autre part l'alignement simple par longueur de paragraphes donne déjà d'assez bons résultats. – Il faut donc systématiquement développer de grands corpus bilingues avec

[15] Pour l'instant la longueur de cette liste est tout simplement un paramètre libre, on garde par exemple les meilleures 50 pairs. Nous essayons actuellement d'améliorer l'algorithme pour qu'il ne garde en mémoire que des couples « utiles » pour l'alignement.

[16] Aussi bien pour les textes franco-chinois que franco-allemands, le système sort beaucoup de noms de personnes et de lieux – des mots qui semblent avoir des signaux les plus proprement. Pour l'alignement franco-allemand, on aurait donc aussi pu passer par les cognats. Il est peut-être souhaitable de tester d'abord la présence de cognats avant de lancer la grande moulinette de la dilatation temporelle.

des décalages (insertion de paragraphes sur un côté) pour voir l'avantage de cet algorithme à l'usage.

Des travaux sont en cours pour intégrer l'alignement par dilatation temporelle dans un système en ligne existant (elizia.net/alignator) qui pour l'instant permet d'aligner deux textes sur base de longueur de paragraphes, de corriger les alignements effectués à la main (interface javascript avec glisser-déposer des paragraphes) et d'exporter l'alignement pour un traitement ultérieur dans un logiciel de textométrie. Le système final devrait être disponible à la fin de l'année 2007 et permettra à toute personne intéressée d'aligner des corpus bilingues sans installer de logiciels et sans fournir de ressources linguistiques, les outils existants n'étant souvent pas à la portée d'un traductologue non informaticien.

Remerciements

Merci à Serge Fleury, André Salem et Miao Jun pour des discussions sur ce sujet qui m'ont inspiré à faire ce travail. Merci aussi à Karen Ferret et *last but not least* à deux relecteurs anonymes des JADT qui ont dû lire une version très préliminaire et peu corrigée de ce papier.

Références

Fung P. and K. McKeown. (1994). Aligning Noisy Parallel Corpora across Language Groups: Word Pair Feature Matching by Dynamic Time Warping, In *Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA-94)*, 81-88, Columbia, Maryland.

Harris Z. S. (1945). Discontinuous Morphemes, *Language*, Vol. 21, No. 3 (Jul. - Sep., 1945).

Jelinek F. (1997). *Statistical Methods for Speech Recognition*. MIT Press, 1997.

Knight K. and Graehl J. (2001). Machine Transliteration, *Computational Linguistics*, 24(4), 1998.

Lebart L. and Salem A. (1994). *Statistique textuelle*. Paris : Dunod.

Meng H., Lo W. K., Chen B. and Tang K. (2001). Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval, *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Trento, Italy, December 2001.

Ratanamahatana C. A. and Keogh E. (2004). Everything you know about Dynamic Time Warping is Wrong. *Third Workshop on Mining Temporal and Sequential Data*, 2004, Seattle, WA.

Ribeiro A., Dias G., Lopes G., Mexia J. (2001). Cognates Alignment. In Bente Maegaard (Ed.), *Proceedings of the Machine Translation Summit VIII*.

Simard M., Foster G. and Isabelle P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation TMI-92* (pp. 67-81), Montréal.

Somers H. (1998). Further experiments in bilingual text alignment. *International Journal of Corpus Linguistics* 3, 115-150.

Wagner R. A. and Fischer M. J. (1974). The String-to-String Correction Problem, *Journal of the ACM*, 21(1): 168-173.

Yarowsky D., Nag G. and Wicentowski R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. *First International Conference on Human Language Technologies*.

Zimina M. (2000). Alignement de textes bilingues par classification ascendante hiérarchique. *Actes des 5es JADT*, Lausanne.