Alignement bi-textuel adaptatif basé sur des plongements multilingues

Olivier Kraif

Univ. Grenoble Alpes, LIDILEM , F-38000 Grenoble, France

olivier.kraif@univ-grenoble-alpes.fr

RÉSUMÉ

Nous présentons dans cet article un système d'alignement bi-textuel adaptatif nommé AIlign. Cet aligneur s'appuie sur les embeddings de phrases pour extraire des points d'ancrage fiables susceptibles de guider le chemin d'alignement, même pour des textes dont le parallélisme est fragmentaire et non strictement monotone. Dans une expérimentation sur plusieurs jeux de données, nous montrons qu'AIlign obtient des résultats équivalents à l'état de l'art, avec une complexité quasi linéaire. En outre, AIlign est capable de traiter des textes dont les propriétés de parallélisme et de monotonie ne sont satisfaites que localement, contrairement à des systèmes tels que Vecalign ou Bertalign.

ABSTRACT

Adaptative Bitextual Aligning Using Multilingual Sentence Embedding

In this paper, we present an adaptive bitextual alignment system called AIlign. This aligner relies on sentence embeddings to extract reliable anchor points that can guide the alignment path, even for texts whose parallelism is fragmentary and not strictly monotonic. In an experiment on several datasets, we show that AIlign achieves results equivalent to the state of the art, with quasi-linear complexity. In addition, AIlign is able to handle texts whose parallelism and monotonicity properties are only satisfied locally, unlike recent systems such as Vecalign or Bertalign.

MOTS-CLÉS : alignement bi-textuel, corpus parallel, plongement de phrases

KEYWORDS : bitextual alignment, parallel corpora, sentence embeddings

ARTICLE **: Accepté à TALN**

1. Introduction

L'alignement bi-textuel phrastique consiste à repérer, entre deux textes en relation d'équivalence traductionnelle, les phrases ou groupe de phrases équivalents. Cette relation d'équivalence traductionnelle peut-être considérée d'un point de vue élargi, sans se restreindre au cas d'un texte source aligné à sa traduction dans une langue cible : elle peut aussi concerner deux textes issus de la traduction d'un même texte original, dans deux langues différentes, voire dans la même langue cible (c'est pourquoi nous parlons ici d'alignement bi-textuel plutôt que d'alignement bilingue).

Les techniques d'alignement bi-textuel sont apparues dans les années 1990 (Gale & Church, 1991; Brown *et al.*, 1991; Kay & Roscheisen, 1988) et ont par la suite largement participé à l'essor de la traduction statistique, en permettant d'aligner de vastes corpus parallèles. Bien que la SMT se soit révèlée assez robuste vis-à-vis des erreurs d'alignement dans ses corpus parallèles, il a été montré que la traduction neuronale pouvait fortement pâtir de ces erreurs (Khayrallah and Koehn, 2018). La qualité de l'alignement bilingue utilisé en entrée d'un système de NMT reste donc un enjeu, d'autant que, comme l'avait déjà signalé Davis *et al.* (1993), et comme nous l'avons montré avec l'alignement de corpus issus de Wikipedia (Kraif, 2024), de nombreuses traductions sont bruitées et constituent un défi pour l'alignement phrastique.

Dans cet article, nous proposons une architecture adaptative s'appuyant sur un alignement en deux étapes, en utilisant les plongements de phrase multilingue pour identifier les zones alignables avant d'utiliser les méthodes plus couteuses de programmation dynamique.

Nous montrons de la sorte que l'on peut atteindre l'état de l'art en réduisant considérablement le cout algorithmique, voire le dépasser pour des textes ne respectant pas les contraintes de monotonie.

2. Travaux antérieurs

Historiquement, les premiers systèmes s'appuyaient sur des indices superficiels tels que les longueurs de phrases, les cognats (Church, 1993, Simard, Foster & Isabelle, 1992 ; Mc Enery & Oakes, 1995 ; Kraif, 2001 ; Lamraoui & Langlais, 2013) ou des lexiques bilingues externes (Varga *et al.*, 2007) ou dérivés des corpus (Moore, 2002).

Plus récemment, plusieurs auteurs ont montré qu'on pouvait atteindre un nouvel état de l'art en utilisant les plongements de phrase multilingues, tel que ceux du système Laser (Artetxe and Schwenk, 2019) ou Labse (Feng *et al.*, 2020). Thomson & Koehn (2020), avec le système Vecalign, développent une mesure de distance basée sur le cosinus des vecteurs des phrases comparés, normalisé par une sélection aléatoire de vecteurs (ils utilisent Laser). Ils remarquent que le cosinus peut être calculé pour des blocs de phrases, par simple sommation des vecteurs. De la sorte, ils proposent d'utiliser une approche de “recursive DP approximation”, afin d'identifier le meilleur chemin par étape, en alignant d'abord des blocs de phrases, puis en augmentant progressivement la résolution avec des blocs de plus en plus petits. En appliquant cette méthode récursivement, ils montrent que la complexité, de quadratique, devient linéaire. Dans l'évaluation de leur système, ils obtiennent les meilleurs résultats sur le dataset Text+Berg (Volk *et al.*, 2010) par rapport à 5 autres aligneurs en compétition: Gale & Church (1991), BMA (Moore, 2002), Hunalign (Varga *et al.*, 2007), Bleualign (Sennrich & Volk, 2010), Gargantua (Braune & Fraser, 2010), and Coverage-Based (Gomes & Lopes, 2016). Sur un corpus extrait du dataset Bible (Christodoupoulus & Steedman, 2015), ils obtiennent une amélioration de 28 point de F1 par rapport à Hunalign.

Liu & Zhu (2022), ont proposé une architecture s'appuyant sur les vecteurs de Labse, avec une stratégie en deux étapes : d'abord un chemin optimal est extrait à base d'appariement 1-1, 0-1 et 1-0 ; puis les alignements n-n sont extraits entre les points obtenus, avec un algorithme de DP. Sur le dataset Bible un ainsi que sur un corpus littéraire anglais-chinois, le système Bertalign obtient le meilleur score F1 par rapport à 5 autre systèmes, incluant Vecalign.

3. Le système AIlign

Pour réduire l'espace de recherche, et s'adapter à des textes comportant d'importants phénomènes de rupture de parallélisme (suppressions, ajouts, interversions de passages), nous proposons de mettre en oeuvre la technique développée par Church (1993). Ce dernier proposait d'identifier des points d'ancrage fiables susceptibles de guider le chemin d'alignement, en s'appuyant sur des correspondances ponctuelles. En l'absence de plongements, la méthode *Char_align* s'appuyait sur la correspondance de 4-grams de caractères. Du fait de la forte présence de cognats (entitées nommées, dates, quantités) dans les paires de phrases alignées, on constate que le chemin d'alignement est caractérisé par une forte densité de points. Un filtre passe bas ainsi qu'un seuillage permet ainsi de conserver les meilleurs points autour du chemin. Notre idée est d'appliquer la même méthode, mais en utilisant une source d'information beaucoup plus riche et moins bruitée que des n-grams : les plongements multilingues.

Nous proposons ainsi une architecture en deux étapes : tout d'abord, nous extrayons des points d'ancrage à partir des appariements de phrase ayant dépassé un certain seuil de similarité ; ces points d'ancrage permettent d'identifier des zones alignables, quand ces points d'ancrage sont suffisamment denses et alignés selon une diagonale locale ; ensuite, à l'intérieur des zones alignables, nous lançons un algorithme de programmation dynamique guidé par ces points d'ancrage.

1.1 Extraction des points d'ancrage

Après avoir calculé les plongements de toutes les phrases des deux textes (avec Labse, Laser ou un encodeur quelconque), une matrice de similarité est calculée à partir du cosinus des vecteurs. Pour chaque phrase source S_i, on calcule $kBest(S_i)$, les k phrases $T_{j1}..T_{jk}$ obtenant les meilleurs scores.

On applique alors un critère de marge : la différence entre les deux meilleurs ne doit pas être inférieure à un seuil (0.05) sans quoi les candidats sont ignorés. On ne retient ensuite que les candidats dont la similarité dépasse un certain seuil *cosThreshold* (0.4). On effectue ensuite le même calcul les phrases cibles T_j, on extrait $kBest(T_j)$ et l'on applique les mêmes critères. Enfin, on ne retient que les points (i,j) tels que $i \in kBest(T_j)$ et $j \in kBest(S_i)$.

Pour chaque point candidat, un filtre passe haut est appliqué : on calcule la densité de points dans un zone centrée sur le point, parallèle à la diagonale, d'une longueur *deltaX* (20) et d'une hauteur *deltaY* (3). Si le rapport entre cette densité et la densité moyenne est inférieure à un certain seuil (*minDensityRatio*=0.3), le point est éliminé, comme on le voit figure 1.

Une deuxième étape de filtrage permet ensuite de résoudre les conflits lorsque deux ou plusieurs points sont situés sur une même ligne ou une même colonne : on ne retient que le point qui obtient la mesure de densité locale la plus élevée.

3.1 Détermination des intervalles alignables

Optionnellement, on peut s'appuyer sur les points d'ancrage pour déterminer les intervalles alignables entre la source et la cible.

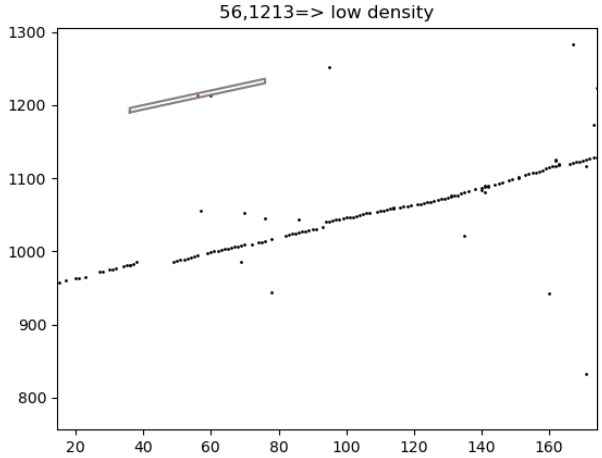

Figure 1: Elimination des points d'ancrage candidats en fonction de la densité locale

Si on note *Anchors* la liste des points d'ancrage triés par ordonnée croissante, l'algorithme d'extraction des intervalles est le suivant :

- Pour chaque point d'ancrage Anchors[i]=(x_i,y_i), on calcule la déviation de (x_i,y_i) par rapport à la diagonale passant par le point précédent (x_{i-1},y_{i-1}) :

$$deviation = \left(\left(y_i - \left(y_{i-1} + \left(x_i - x_{i-1} \right) * sentRatio \right) \right) \right)$$

- Le point (x_i,y_i) est ignoré dans deux cas de figure :
  - s'il ne respecte pas la contrainte de monotonie par rapport aux deux points qui précèdent Anchors[i-2], Anchors[i-1] et deux qui suivent Anchors[i+1], Anchors[i+2], alors que ceux-ci sont monotones ($x_{i-2} \leq x_{i-1} \leq x_{i+1} \leq x_{i+2}$ et $y_{i-2} \leq y_{i-1} \leq y_{i+1} \leq y_{i+2}$)
  - si la déviation est supérieure à un seuil (fixé à 10) ou que $y_i < y_{i-1}$ et que la densité locale de (x_i,y_j) est inférieure à la moyenne.
- Si la déviation de (x_i,y_i) est inférieure à 20 (*maxDistToTheDiagonal*), il est inclus dans l'intervalle courant. Sinon un point (x_i,y_i) déviant peut donner lieu à la création d'un nouvel intervalle dans les deux cas de figure :
  - si (x_i,y_i) est aligné avec (x_{i+1},y_{i+1}) (en termes de déviation inférieure au seuil) et sa densité locale est supérieure à 1,5 * la moyenne.
  - si la distance euclidienne entre (x_{i-1},y_{i-1}) et (x_i,y_i) est supérieure à un certain seuil maxGapSize (fixée à 100).

Un point déviant ne satisfaisant pas une de ces conditions est ignoré.

Quand un nouvel intervalle est créé à partir de x_i,y_j, le précédent est cloturé jusqu'à x_{i-1},y_{i-1}. Seuls les intervalles contenant plus d'un point d'ancrage et suffisamment denses (paramètre *minHorizontalDensity*=0.15) sont retenus *in fine*.

L'extraction automatique des intervalles permet de mettre en oeuvre, en option, un mode adaptatif, utile lorsque une partie seulement des textes est alignable : *charRatio* (rapport des longueurs de phrases en caractères) et *sentRatio* (rapport des nombres de phrases entre cible et source) sont recalculés après l'étape d'extraction des intervalles, et l'extraction des points d'ancrage est intégralement relancée avec ces nouvelles valeurs (utiles notamment pour calculer avec finesse la pente des diagonales locales).

3.2 Etape de programmation dynamique

Une fois les points d'ancrage extraits (correspondant en général à un nuage de points assez resserés autour de la diagonale, comme on le voit figure 1), et les intervalles d'alignement définis, un algorithme de programmation dynamique est lancé pour calculer, depuis chaque point d'ancrage, le chemin optimal permettant de mener à ce point.

Les groupements autorisés sont de type 1-0, 0-1, 1-1, 1-2, 2-1, …, n-1, 1-n (n étant contrôlé par le paramètre *maxGroupSize*, fixé à 4 dans nos expériences). On peut optionnellement considérer également les groupements 2-2, et ignorer, toujours en option, les groupements vides (0-1, 1-0). Le calcul du meilleur chemin revient à trouver la meilleure suite de groupements entre les deux extrêmités de la zone alignable, afin de minimiser une mesure de distance entre chaque groupe.

Cette mesure de distance est calculée comme :

$$d_{embed}\left(g_i, g_j\right) = \left(1 - \cos\left(embed\left(g_i\right), embed\left(g_j\right)\right)\right)$$

Contrairement à Bertalign, on n'encode que les phrases prises individuellement, les vecteurs de groupes de phrases étant obtenues par simple addition des vecteurs. Pour les groupes vides (1-0), (0-1) on définit une distance fixe de 1.

Comme chez Liu & Zhu (2022) la mesure de similarité cosinus est diminuée en fonction des cosinus marginaux : on calcule la similarité du groupe1 avec les deux phrases qui entourent le groupe2, la similarité du groupe2 avec les deux phrases qui entourent le groupe1, on prend la moyenne de ces similarités, que l'on multiplie par le coefficient *c* (empiriquement fixé à 0.6).

$$d_{embed'}\left(g_i, g_j\right) = d_{embed}\left(g_i, g_j\right) + c * neighbourSim$$

avec $neighbourSim = 1/4 * (\cos(embed(prec(g_i)), embed(g_j)) + \cos(embed(succ(g_i)), embed(g_j)) + \cos(embed(g_i), embed(prec(g_j))) + \cos(embed(g_i), embed(succ(g_j))))$

Où $prec(g_i)$ est la phrase qui précède g_i et $succ(g_i)$ la phrase qui lui succède.
A l'instar de Thomson & Koehn (2019), comme le cosinus a tendance à avantager les gros groupes (p.ex. 2-2 au lieu deux fois 1-1) on applique une pénalité proportionnelle au nombre de phrases impliquées .

$$d_{embed''}\left(g_i, g_j\right) = d_{embed'}\left(g_i, g_j\right) + p * \left(\text{size}\left(g_i\right) + \text{size}\left(g_j\right)\right)$$

Pour tenir compte également des longueurs de phrases (comme chez Gale & Church, 1991, mais aussi Liu & Zhu, 2022), on définit une deuxième mesure de distance.

$$d_{length}(g_i,g_j)=1-\log_2\left(1+\frac{len_{min}}{len_{max}}\right)$$

où len_{min}= *min(length(g_i),length(g_j))* et len_{max}= *max(length(g_i),length(g_j))* et les longueurs en nombre de charactères sont normalisées par le paramètres *charRatio*, qui peut être calculé automatiquement ou fixé par l'utilisateur.
Par la suite, la mesure de distance finale est calculée comme la somme pondérée des deux distances (empiriquement, w est fixé à 0.33) :

$$d(g_i,g_j)=(1-w)*d_{embed''}(g_i,g_j)+w*d_{length}(g_i,g_j)$$

Enfin, la contribution de chaque groupement à distance totale est multipliée par le nombre de phrases concernés (afin de ne pas favoriser les chemins effectuant quelques grands pas par rapport à de nombreux de petits pas).

$$d_{final}(g_i,g_j)=d(g_i,g_j)*(\text{size}(g_i)+\text{size}(g_j))$$

De la sorte, la distance totale d'un chemin divisée par la somme du nombre de phrases des deux textes donne la distance moyenne de chaque appariement, entre 0 et 1 : ce score peut être utile pour indique la proximité relative des textes après l'alignement.

Pour le calcul du meilleur chemin, on lance un calcul récursif entre chaque point d'ancrage : pour chaque point d'ancrage (x_i,y_i) de *Anchors*, on calcule la deviation de (x_i,y_i) par rapport à la diagonale passant par le point précédent (x_{i-1},y_{i-1}), et si cette déviation est supérieure à un certain seuil (localDiagBeam) on ignore le point (x_i,y_j). Sinon, on lance le calcul récursif du meilleur chemin menant à (x,y) (les chemins optimaux menant jusqu'à (x_{i-1},y_{i-1}) étant stockés dans un tableau, il ne sera pas nécessaire de les recalculer).

4. Expérimentation

Dans une évaluation préliminaire, nous avons comparé les résultats d'AIlign avec 4 aligneurs sur un corpus chinois-français : YASA (Lamraoui & Langlais, 2013), Alinéa (Kraif, 2001, Véronis *et al.*, 2008), LF Aligner[1], et enfin Bertalign. Les résultats figurent dans le tableau 1 ci-après.

	P	R	F
Yasa	0,0886	0,0886	0,0886
Alinea	0,4281	0,2623	0,3253
LF Aligner	0,8273	0,8273	0,8273
AIlign	0,9882	0,9882	0,9882
Bertalign	0,9886	0,9886	0,9886

Tableau 1: Résultats comparés de Bertalign et AIlign sur un corpus chinois-français[2]

[1] Cet aligneur, doté de lexiques bilingues, s'appuie sur HunAlign (Varga *et al.*, 2007). Il est disponible ici : https://sourceforge.net/projects/aligner/

Le système Bertalign représentant l'état de l'art, pour la suite, nous avons seulement comparé les résultats d'AIlign à ce dernier. Afin de varier les couples de langues et les genres textuels, nous avons utilisé les jeux de données suivants :

- Text+Berg (Volk et al., 2010) : un corpus consistant d'articles publiés en français et allemand par le Swiss Alpine Club et déjà utilisé dans plusieurs tâche d'évaluation.

- MD.fr-ar (Veronis et al., 2008) : un corpus d'article du Monde diplomatique traduit du français vers l'arabe, et manuellement aligné pour la campagne Arcade 2.

- BAF (Isabelle, 1992) : un des premiers corpus parallèles anglais et français aligné manuellement, incluant différents genres de texte représentant des difficultés d'alignement variables.

- Grimm : la concaténation des deux volumes des contes de Grimm selon l'édition de *Kinder und Hausmärchen* de 1857, et les *Contes choisis des frères Grimm* traduits en 1864 par D. Baudry. Cette traduction a la particularité de constituer une sélection de 40 contes sur 210, reproduits selon un ordre complètement différent des contes originaux. Pour ces deux ouvrages, nous avons produits un alignement de référence au niveau des contes (et non des phrases), afin d'évaluer l'étape de détection des zones alignables.

Nous avons utilisé les mêmes paramètres par défaut pour tous ces corpus, sauf pour Grimm pour lequel nous avons utilisé le mode adaptatif avec détection d'intervalles. Pour ce dernier corpus, nous ne donnerons que les résultats concernant l'alignement des contes, en l'absence de référence au niveau des phrases.

5. Résultats et discussion

Les valeurs de précision, rappel et F-mesure ont été calculées avec les scores stricts à partir du script fourni par Liu & Zhu (2022) sur leur dépot[3].

	Bertalign			Allign		
Dataset	P%	R%	F%	P%	R%	F%
Text+Berg	93.2	94.1	93.6	90.9	93.4	92.1
MD.ar-en	95.0	95.7	95.4	94.7	96.1	95.4
BAF	92.1	95.6	93.8	91.9	95.9	93.8
Grimm Tale level				98.7	92.8	95.7

Tableau 1: Résultats comparés de Bertalign et AIlign

[2] Il s'agit du roman "Le Clan du Sorgho rouge" de Mo Yan, aligné manuellement par une étudiante de master, Yuhe Tang, que nous remercions. La version française est celle publiée en 2014 aux éditions du Seuil. Pour une question de droit d'auteur, nous ne pouvons fournir ce corpus dans les jeux de données que nous publions en complément de cet article.

[3] https://github.com/bfsujason/bertalign/tree/main

	Bertalign	AIlign
Text+Berg	590 s.	119 s.
MD.ar-en	8114 s.	2166 s.
BAF	10882 s.	1437 s.

Table 2: Temps d'exécution comparé sur CPU de Bertalign et AIlign[4]

Nous avons aussi comparé les performances de Bertalign et AIlign en utilisant une GPU sur une machine plus récente[5], en faisant varier incrémentalement la taille des corpus à aligner (en concaténant les articles successifs du corpus MD), afin de vérifier si la complexité des deux algorithmes est, comme nous le supposons, linéaire. On obtient les courbes de la figure 2 ci-après :

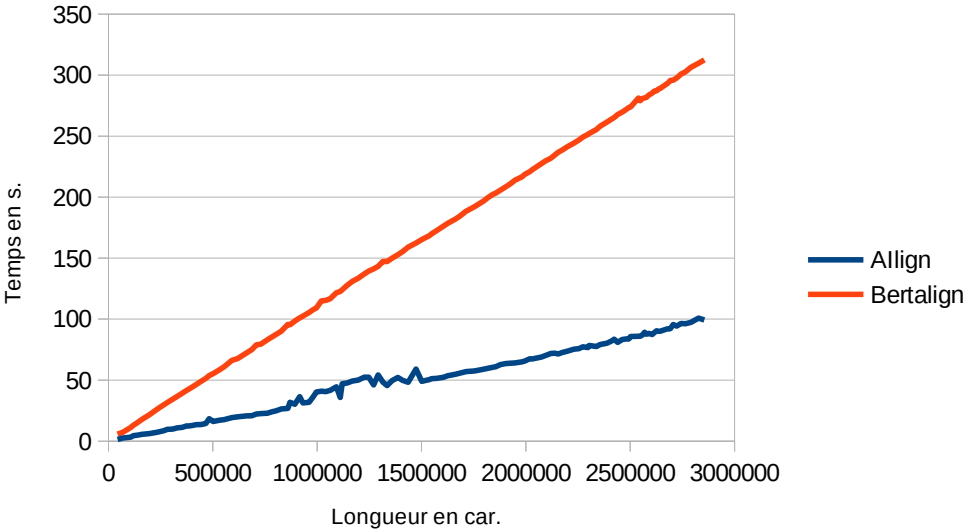

Figure 2: Evolution du temps d'exécution en fonction de la longueur des textes en nombre de caractères (somme des deux textes à aligner), avec une GPU NVIDIA RTX 2000.

5.1 Discussion

On constate qu'AIlign se situe à l'état de l'art, avec des résultats très légèrement inférieurs à ceux obtenus par Bertalign. Son temps d'éxecution est par ailleurs notablement inférieur à celui de Bertalign. La partie la plus couteuse de l'algorithme, pour l'un comme pour l'autre, est le calcul des plongements de phrases par Labse. Mais AIlign ne calcule les plongements que pour les phrases prises individuellement, et non pour les séquences de 1, 2, 3 et 4 phrases consécutives, ce qui pénalise fortement le temps d'éxecution de Bertalign.

[4] Le script a été exécuté sur un ordinateur portable avec Intel Core i7-6700HQ CPU @ 2.60GHz × 8, 16Go RAM.

[5] Dell Précision 7680 avec INTEL CORE i9-13950HX × 32, 32 Go de RAM, équipé d'une GPU NVIDIA RTX 2000 Ada avec 8Go de VRAM.

Vue la densité importante de points d'ancrage, même entre des langues très éloignées (comme le français et l'arabe, ou le français et le chinois), l'exécution de l'algorithme de programmation dynamique devient quasi-linéaire. La seule étape quadratique est le calcul du cosinus pour la matrice de similarité, et la recherche des *kBest* points en ligne et en colonne, mais ces calculs s'avèrent très brefs (moins d'une seconde) même pour des textes longs, et ne représentent généralement qu'une très petite fraction de l'ensemble (p.ex. 6 s. pour Grimm). Ils sont en outre aisément parallélisables.

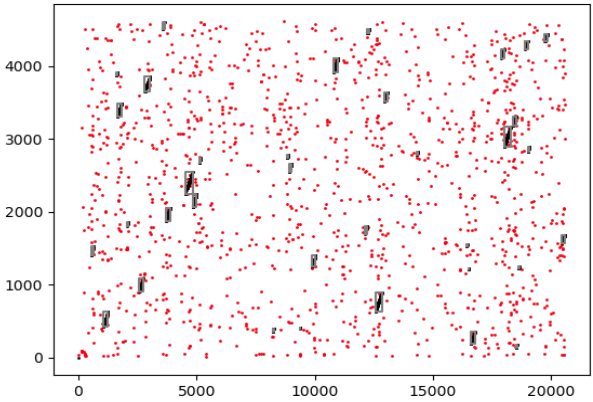

Figure 2: Identification des intervalles alignables pour le corpus Grimm (rectangles gris)

En ce qui concerne la détection des intervalles alignables, on constate que les résultats sur le corpus Grimm sont de très bonne qualité. Seuls trois contes sont complétement absents : deux très courts (6 phrases pour *Der undankbare Sohn*, 13 phrases pour *Gottes Speise*) et un conte écrit en allemand dialectal (*Das Bürle im Himme*l). Un 4e conte en dialecte est tronqué pour moitié (*Von dem Fischer un syner Fru*), ce qui explique une perte relative de rappel alors que la précision reste très élevée.

6. Conclusion

Dans le domaine de l'alignement bi-textuel, nous avons montré comment tirer parti au mieux de cette nouvelle source d'information que constituent les plongements multilingues tels que ceux produits par Labse. Grâce à ces *transformers* multilingues, un nouvel état de l'art a pu être atteint sur cette tâche, et il est désormais possible d'obtenir un alignement de haute qualité pour des bi-textes comportant d'importantes ruptures de parallélisme, comme dans le cas de la publication d'une sélection des contes de Grimm sans respecter l'ordre de la publication original. Dans de futurs travaux nous chercherons à appliquer notre approche sur des traductions bruitées issues de Wikipedia. Par ailleurs, plutôt que des heuristiques reposant sur des paramétrages empiriques difficiles à généraliser, nous prévoyons d'expérimenter des algorithmes plus génériques et plus robustes pour le filtrage des points d'ancrage, en nous appuyant soit sur l'algorithme DBSCAN pour repérer les zones denses, soit sur la transformée de Hough pour l'identification de segments de droites au sein du nuage de point.

Pour assurer la reproductibilité des résultats, nous fournissont les codes d'AIlign, un fork des codes de Bertalign, ainsi que tous les jeux de données utilisés dans nos expérimentations, à l'adresse suivante: https://gricad-gitlab.univ-grenoble-alpes.fr/kraifo/

7. Références

ARTETXE M. & SCHWENK H. (2019). Margin based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

BRAUNE F. & FRASER A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Coling 2010: Posters*, pages 81–89, Beijing, China.

BROWN P., LAI J. & MERCER R. (1991). Aligning Sentences in Parallel Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, ACL-91*, pages169-176 Morristown, NJ, June. Association for Computational Linguistics.

CHRISTODOULOUPOULOS C. & STEEDMAN M. (2015). A massively parallel corpus: the Bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

CHURCH K. W. (1993). Char align : A program for Aligning Parallel Texts at the Character Level. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, ACL-93*, pages 1-8, Columbus Ohio, June. Association for Computational Linguistics.

DAVIS M. W., DUNNING T. E. & OGDEN W. C. (1995). Text Alignment in the Real World : Improving Alignments of Noisy Translations Using Common Lexical Features. In *Proceedings of EACL 95*, Dubli, Ireland, May. Association for Computational Linguistics.

FENG F., YANG Y., CER D., ARIVAZHAGAN N. & WANG, W. (2022). Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers*, pages 878 - 891, Dublin, Ireland, May. Association for Computational Linguistics.

GALE W. A. & CHURCH K. W. (1991). A Program for Aligning Sentences in Bilingual Corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 177-184, Morristown, NJ, June. Association for Computational Linguistics.

GOMES L. & PEREIRA LOPES G. (2016). First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2228–2231, Portorŏz, Slovenia. European Language Resources Association (ELRA)

KHAYRALLAH, H. AND KOEHN, P. (2018). On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

KRAIF O. (2001). Exploitation des cognats dans les systèmes d'alignement bi-textuel : architecture et évaluation. *TAL* 42 :3, 833-867.

KRAIF, O. (2024). Mining parallel corpora in Wikipedia. In Poudat, C., Lungen, H., Hertzberg, L. (eds). *Investigating Wikipedia : linguistic corpus building, exploration and analyses*. Coll. Studies in Corpus Linguistics, John Benjamins Publisher.

LAMRAOUI F. & LANGLAIS P. (2013). Yet Another Fast and Robust and Open Source Sentence Aligner. Time to Reconsider Sentence Alignment ?, In *Proceedings of the XIV Machine Translation Summit*, pages 77-84, Nice, France, September.

LIU L. & ZHU M. (2022). Bertalign: Improved word embedding-based sentence alignment for Chinese–English parallel corpora of literary texts. *Digital Scholarship in the Humanities*, Volume 38, Issue 2, June 2023, 621–634.

MCENERY, A. M. & OAKES M. P. (1996). Sentence and word alignment in the CRATER project : methods and assessment. *Using corpora for language research*, Longman, 211-231.

MOORE R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144, Tiburon, USA, October. Springer.

SENNRICH R. & VOLK M. (2010). MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, Denver, Colorado, USA. Association for Machine Translation in the Americas.

SIMARD M., FOSTER G. & ISABELLE P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. In *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, TMI-92*, pages 67-81, Montréal, Canada. CCRIT.

THOMSON B. & KOEHN P. (2020). Vecalign: Improved Sentence Alignment in Linear Time and Space. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1342–1348, Hong Kong, China, November 3–7, 2019.

VARGA, D., HALÁCSY, P., KORNAI, A., NAGY, V., NÉMETH, L. & TRÓN V. (2007). Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science* Series 4, 292:247.

VÉRONIS J., HAMON O., AYACHE, C., BELMOUHOUB, R., KRAIF, O., LAURENT, D., NGUYEN T., SEMMAR N., STUCK. F & ZAGHOUANI, W. (2008). La campagne d'évaluation ARCADE 2. In Stephane Chaudiron, Khalid Choukry (sous la dir. de) *L'évaluation des technologies de traitement de la langue*, Hermès, Lavoisier, Paris, pp. 47-69

VÉRONIS J. & LANGLAIS P. (2000). Evaluation of parallel text alignment systems: the ARCADE project. In J. Véronis (Ed.), *Parallel text processing: Alignment and use of translation corpora*. Dordrecht: Kluwer Academic Publishers, pp. 369--388.

VOLK M., BUBENHOFER N., ALTHAUS A., BANGERTER M., FURRER L. & RUEF B. (2010). Challenges in building a multilingual alpine heritage corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Languages Resources Association (ELRA)