Alignements divisifs de textes parallèles: données, algorithme et évaluation

Joanna Radoła François Yvon
Sorbonne Université, CNRS, ISIR, 4, Place Jussieu, 75005 Paris, France
radola@isir.upmc.fr, yvon@isir.upmc.fr

RÉSUMÉ

Nous présentons Alibi - un corpus d'alignements hiérarchiques sous-phrastiques français-anglais, annoté manuellement à l'aide d'une stratégie divisive. Nous comparons globalement les alignements ainsi obtenus avec plusieurs corpus parallèles alignés mot-à-mot et étalonnons sa difficulté en réalisant des alignements automatiques par des méthodes de l'état de l'art. Nous proposons également un algorithme exploitant des représentations neuronales des mots et des groupes de mots afin de reproduire les alignements hiérarchiques de référence. Enfin, nous proposons une métrique d'évaluation des arbres d'alignement avec laquelle nous comparons les performances de plusieurs variantes de l'algorithme d'alignement, obtenues en faisant varier les mesures d'appariemment de groupes de mots. Nos résultats montrent que (a) les arbres d'alignements de référence sont très ambigus et difficiles à reproduire automatiquement, cependant, les alignements mot-à-mot sont prédits de manière fiable ; (b) l'utilisation d'alternatives à la similarité cosinus pour évaluer l'appariemment de blocs permet d'améliorer significativement les résultats du système de base.

ABSTRACT

Divisive Alignment of Parallel Corpora - Data, Algorithm and Evaluation

We introduce Alibi - a corpus of French-English hierarchical sub-sentential alignments, human-annotated with use of the divisive methodology. We quantitatively compare these alignments to other parallel corpora aligned at word level and benchmark our corpus' difficulty using state-of-the-art alignment tools. Subsequently, we propose an algorithm that leverages neural representations of words and text segments to replicate the reference hierarchical alignments. Finally, we propose an evaluation metric for alignment trees and report scores obtained after testing multiple variations of the divisive alignment algorithm, using different metrics to pair sentence segments together. Our results suggest that (a) while reference alignment trees are highly ambiguous and difficult to reproduce automatically, word-level alignments are reliably predicted ; (b) using alternatives to the cosine similarity to match segment pairs allows us to significantly improve baseline results.

MOTS-CLÉS : corpus parallèles, alignements de mots, alignements hiérarchiques, similarité lexicale, plongements lexicaux.

KEYWORDS: parallel corpora, word alignments, hierarchical alignments, lexical similarity, word embeddings.

Contribution originelle

1 Introduction

L'alignement de mots au sein de segments parallèles est une tâche élémentaire en traitement automatique multilingue. Étant donné un couple de segments parallèles, il s'agit d'identifier des équivalents de traduction lexicaux entre la source et la cible (Tiedemann, 2011). Ces alignements sont utilisés pour extraire automatiquement des lexiques bilingues généralistes ou spécialisés (Liu *et al.*, 2013), pour superviser l'attention dans des modèles de traduction neuronale (Alkhouli & Ney, 2017), pour analyser les sorties de systèmes de traduction (Stahlberg *et al.*, 2018) ou encore pour détecter les confabulations des systèmes de traduction automatique (Wu *et al.*, 2024). Il existe de multiples manières de résoudre ce problème, soit en s'appuyant sur des cooccurrences entre vocables cibles et sources, soit en construisant des modèles probabilistes de ces associations lexicales, comme le formulent les modèles de la famille IBM (Brown *et al.*, 1993; Och & Ney, 2003) ou les modèles hiérarchiques de Wu (1997). Des évolutions récentes de ces méthodes exploitent des représentations denses calculées par des encodeurs neuronaux : par exemple, SimAlign (Jalili Sabet *et al.*, 2020) et AwesomeAlign (Dou & Neubig, 2021) s'inspirent des méthodes heuristiques d'alignement en s'appuyant sur des représentations lexicales denses calculés par des modèles de langue ; Wang *et al.* (2018) développent une version neuronale des modèles IBM. D'autres méthodes utilisent le mécanisme d'attention de modèles Transformer (Vaswani *et al.*, 2017) pour extraire de tels alignements (Garg *et al.*, 2019). Une autre approche récente consiste à entraîner des grands modèles de langue (LLMs) sur la tâche d'extraction de réponse. Cette méthode, introduite avec l'outil WSPAlign (Wu *et al.*, 2023), est faiblement supervisée.

Les méthodes d'alignement restent toutefois majoritairement non-supervisées, les rares corpus annotés par des alignements de référence étant réservés à l'évaluation des méthodes automatiques. Pour le français, les principales ressources existantes sont Blinker (La Bible, Melamed, 2001), les données parlementaires des Hansards (Och & Ney, 2003) et les transcriptions de débats au parlement européen (Europarl) de Graça *et al.* (2008). Des différences fondamentales existent toutefois entre ces corpus, dues à la formulation des consignes d'annotation.

Dans ce travail, nous présentons une nouvelle ressource pour le français, correspondant à l'alignement intégral de cinq nouvelles avec leur traduction en anglais. Elle se distingue des autres ressources de ce type par la manière dont les annotations sont collectées, en adoptant une procédure itérative consistant à raffiner de manière itérative des alignements de phrase, selon une proposition initialement formulée par Xu & Yvon (2016). Cette procédure présente l'avantage de ne pas chercher à construire à priori des alignements au niveau des mots. Ceci permet de mieux aligner des expressions polylexicales, en gardant toujours un nœud pour l'expression entière et en divisant d'une manière plus fine seulement lorsque cela est possible. Nous décrivons en détail cette procédure, en contrastant les alignements ainsi collectés avec les autres données de référence pour les alignements français-anglais.

Dans un second temps, nous établissons des scores de référence pour l'alignement automatique de ces textes en utilisant des méthodes de l'état de l'art, puis discutons de la question plus complexe du calcul automatique d'alignements hiérarchiques et de leur évaluation, en nous appuyant sur plusieurs implémentations « neuronales » de l'algorithme de Lardilleux *et al.* (2012), pour lequel nous proposons diverses évolutions et variantes.

Les contributions de ce travail sont donc plurielles : elles consistent en un nouveau corpus annoté, ainsi que des propositions algorithmiques pour construire et évaluer des alignements hiérarchiques. Les données, les scripts et les documentations associées peuvent être téléchargées depuis le site web du projet : `https://github.com/fyvo/AlibiWordAlignments`.

	phrases	mots (FR)	mots (EN)
[CHA] Le Chat botté (Charles Perrault, 1695) Version française : https://fr.wikisource.org/wiki/Le_MaÃ®tre_chat_ou_le_Chat_bottÃ1' Traduction : https://en.wikisource.org/wiki/Old_time_stories_(Perrault,_Robinson)/Puss_in_Boots	52	1 861	2 036
[BAR] La Barbe-Bleue (Charles Perrault, 1697) Version française : https://fr.wikisource.org/wiki/Contes_de_Perrault_(Ã1'd._1902)/La_Barbe-Bleue Traduction : https://en.wikisource.org/wiki/The_fairy_tales_of_Charles_Perrault_(Clarke,_1922)/Blue_Beard	68	2 145	2 234
[VIS] Vision de Charles XI (Prosper Mérimée, 1829) Version française : https://fr.wikisource.org/wiki/Vision_de_Charles_XI Traduction : https://en.wikisource.org/wiki/The_Writings_of_Prosper_M%C3%A9rim%C3%A9e/Volume_3/The_Vision_of_Charles_XI	106	2 912	2 790
[DER] La Dernière classe (Alphonse Daudet, 1873) Version française : https://fr.wikisource.org/wiki/Les_Contes_du_lundi Traduction : https://en.wikisource.org/wiki/The_Last_Lesson	92	2 045	1 935
[AUB] L'Auberge (Guy de Maupassant, 1886) Version française : https://fr.wikisource.org/wiki/Le_Horla_(recueil)/L'Auberge Traduction : https://en.wikisource.org/wiki/The_Inn	204	5 674	5 862

TABLE 1 – Informations sur les œuvres, longueur en phrases et en nombre de mots.

2 Un corpus littéraire d'alignements divisifs

2.1 Préparation du corpus

Nous avons sélectionné 5 courtes œuvres de la littérature française et leur traduction en anglais, choisies parmi un ensemble de textes libres de droits disponibles sur la toile [1]. Après suppression des balises HTML, ces textes ont été segmentés en phrases, puis annotés automatiquement en partie du discours et analysés syntaxiquement. Pour les versions françaises et anglaises, l'étiquetage morpho-syntaxique est calculé par le MaxentTagger de Stanford (Toutanova *et al.*, 2003) [2] ; les dépendances syntaxiques sont calculées respectivement par les modèles *frenchFactored* et *englishPCFG*.

Enfin, ces documents ont été alignés automatiquement au niveau des phrases avec l'outil hunalign (Varga *et al.*, 2007). Cet alignement a ensuite été révisé et validé manuellement. Les statistiques de base concernant ces données sont dans le Tableau 1.

2.2 Procédure d'alignement

L'annotation des alignements a été réalisé par une unique annotatrice, traductrice professionnelle et rémunérée pour cette tâche, avec des échanges réguliers avec les responsables scientifiques pour discuter des alignements problématiques, sur une durée d'environ deux mois.

Chaque étape d'annotation manuelle se déroule comme suit : des segments parallèles sont préparés et rendus disponibles sur une interface d'annotation. L'annotatrice identifie alors un point de segmentation unique dans les côtés source et cible de chaque segment, ainsi que leur *orientation relative* (monotone, si les deux fragments sources et cibles apparaissent dans le même ordre, anti-monotone, sinon). Sur cette base, les segments parallèles en entrée sont effectivement divisés et sont inclus dans le prochain lot de segments à annoter. Cette procédure est illustrée sur la Figure 1.

1. Les textes originaux sont disponibles sur Wikisource, comme leurs traductions en anglais. Les licences et restrictions d'usage sont documentées sur ce même site (voir https://fr.wikisource.org/wiki/Aide:Droit_d'auteur).
2. Voir https://nlp.stanford.edu/software/tagger.shtml pour la documentation des annotations.

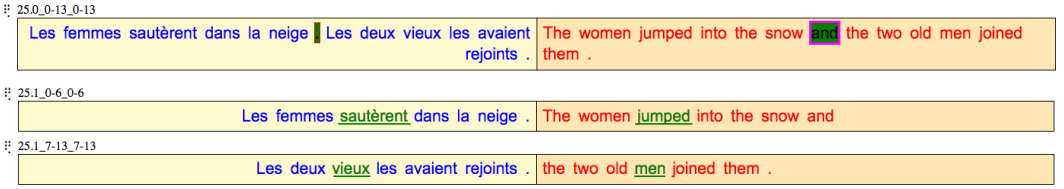

FIGURE 1 – Annotation d'alignements hiérarchiques. La partie supérieure montre le texte en cours d'annotation, où les mots qui correspondent à des points de segmentation sont marqués. La partie inférieure représente les deux couples de segments résultant de cette segmentation (monotone), qui feront l'objet de la phase suivante de segmentation.

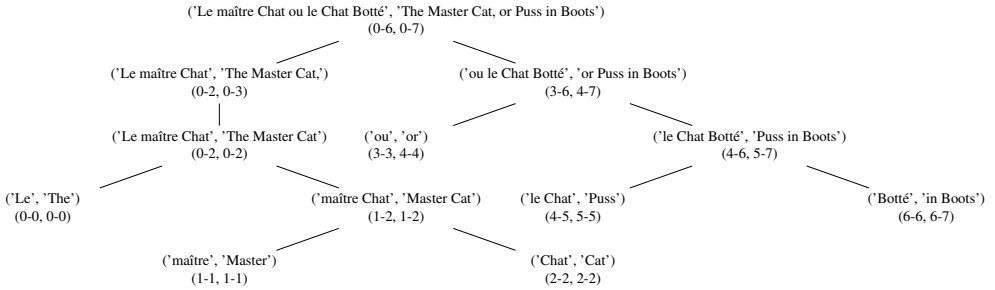

FIGURE 2 – Un exemple d'arbre de segments alignés.

Lorsque plusieurs points de segmentation sont possibles [3], les consignes sont de segmenter préférentiellement à la frontière de constituant syntaxique de plus haut niveau ; puis de construire des segmentations en fragments équilibrés en taille. La segmentation s'arrête quand il n'est plus possible de segmenter en deux blocs, soit du fait de l'impossibilité de décomposer l'association bilingue en deux sous-parties (cas d'une locution ou d'un idiome ou d'un terme complexe) ; soit parce que l'un des côtés du segment parallèle ne contient qu'un seul mot ; soit, dans de rares cas, parce que les associations entre sous-parties des segments exigeraient une segmentation en trois fragments. Des exemples de telles configurations difficiles à aligner et qui ne peuvent être prédites par l'algorithme divisif binaire sont donnés dans Kaeshammer & Westburg (2014), voir également Søgaard (2010).

La première de ces conditions d'arrêt est la principale justification de l'approche divisive et permet aux annotateurs de construire des associations lexicales au niveau le plus fin uniquement quand cela est pertinent, au lieu de prendre systématiquement les mots comme unités d'alignement.

Le résultat de cette procédure est un alignement hiérarchique complet des 5 œuvres, correspondant à un ensemble d'arbres binaires, où chaque phrase parallèle est répresentée comme un arbre (voir l'exemple de la Figure 2). Le corpus complet représente un total d'environ 23 000 segments (nœuds) alignés. Le détail, œuvre par œuvre, ainsi que les statistiques sur la profondeur des arbres d'alignement sont dans le Tableau 2.

Le résultat de l'annotation est diffusé sous la forme d'un document XML pour chaque texte, contenant à la fois les informations linguistiques associées à chaque phrase, ainsi que la hiérarchie complète

3. Ainsi, il serait possible de segmenter l'exemple de la Figure 1 après *(femmes, women)*, ou encore *(sautèrent, jumped)*.

texte	nb segments		profondeur d'arbre			mots/phrase	
	tous	feuilles	moy	max	min	FR	EN
CHA	3091	1453	7,4	10	4	36,5	39,9
BAR	3311	1522	7,1	11	1	32,0	33,3
VIS	4480	2165	6,8	12	2	27,7	26,6
DER	2902	1420	5,5	11	1	22,5	21,3
AUB	9144	4368	6,9	12	2	28,0	28,9
Tous	22928	10928	6,7	12	1	28,3	28,7

TABLE 2 – Nombre de tous les segments et des feuilles uniquement, profondeur des arbres d'alignement et longueur moyenne des phrases pour les 5 textes parallèles du corpus.

des points de segmentation. Ce corpus, ainsi que les éléments de documentation associés (contenu, formats, etc) est distribué sous licence *Creative Commons CC-BY-SA* [4].

3 Comparaison avec les corpus existants

3.1 Nombre et type des liens

Les ressources existantes pour l'alignement mot-à-mot en français sont représentées, pour chaque phrase parallèle, par un ensemble de couples d'indices identifiant les mots associés. Un premier facteur qui distingue les corpus existants entre eux est le découpage en mots qui est sous-jacent aux alignements. Les annotations manuelles distinguent les associations dites **sûres** et celles qui sont simplement **possibles**, avec une interprétation qui varie d'un corpus à l'autre [5], ce qui est un autre facteur majeur de différentiation.

Pour permettre la comparaison entre les alignements obtenus par l'annotation divisive et les autres alignements de référence, les heuristiques suivantes sont utilisées pour convertir les arbres d'alignement hiérarchiques en alignements mot-à-mot :

1. Un nœud feuille correspondant à un alignement un-à-un est transformé en un lien **sûr**.
2. Pour les feuilles contenant plus d'un mot, un lien **possible** est créé pour tout couple (mot source, mot cible) apparaissant dans ce bloc. Ainsi, un alignement de la feuille *(Barbe Bleue, Blue Beard)* associant deux segments de longueur 2 donnera lieu à quatre liens **possibles**.

Ainsi, l'alignement mot-à-mot correspondant à l'alignement de la Figure 2 est : `0-0 1-1 2-2 3-4 4p5 5p5 6p6 6p7`.

Comme montre le Tableau 3, notre corpus excède en taille les ressources existantes, représentant un ajout encore plus significatif lorsqu'on l'évalue en nombre d'unités textuelles. Ceci est dû au fait que les autres corpus contiennent des textes juridiques, tandis que le nôtre est constitué sur de textes littéraires, avec des phrases plus longues en moyenne.

On observe que le corpus Hansards contient trois fois plus de liens possibles que de liens sûrs, alors qu'à l'inverse Europarl en contient très peu. Nos annotations représentent une situation intermédiaire,

4. https://creativecommons.org/licenses/by-sa/4.0/deed.en
5. Documentant EPPS, (Graça *et al.*, 2008) distingue ainsi : « an S-alignment is used when a translation is possible in every context, (...) we considered P-alignments when a translation was possible in certain contexts ».

	nb	nb unités		nb liens		nb non-alignés	
	phrases	FR	EN	sûrs	poss.	FR	EN
HANS	447	7 761	7 020	4 038	13 400	349	327
EPPS	100	1 227	1 072	1 009	437	76	75
Alibi	522	14 637	14 857	8 452	14 147	975	1 212

TABLE 3 – Comparaison des corpus alignés mot-à-mot pour la paire anglais-français (HANS-Hansards, EPPS-Europarl).

avec environ 60% de liens étiquetés comme **possibles**.

On observe aussi un nombre de liens nuls (mots non-alignés) assez significatif dans notre corpus. Ceci est dû au caractère littéraire des textes, qui ont moins tendance à être traduits mot-à-mot. Par exemple : la phrase française « ***Un meunier ne laissa pour tous biens, à trois enfants qu'il avait, que (…)*** » est rendue en anglais par « ***Once upon a time there was a miller who left no more riches to the three sons he had than (…)*** ». Cette traduction entraîne des liens nuls pour les six premiers mots en anglais, le premier segment aligné étant *(un meunier, a miller)*.

3.2 Évaluation des alignements mot-à-mot

SimAlign (Jalili Sabet *et al.*, 2020) exploite des plongements lexicaux multilingues pour construire automatiquement des alignements (au niveau des unités sous-lexicales ou des mots). Cet outil ne nécessite aucun pré-entraînement sur des données parallèles, et peut en particulier être utilisé pour aligner des langues rares ou minoritaires. Par défaut, SimAlign obtient des plongements à partir de mBERT (Devlin *et al.*, 2019) ou XLM-R (Conneau *et al.*, 2020) et calcule les similarités cosinus entre toutes les unités des phrases source et cible. Trois heuristiques permettent ensuite de dériver de cette matrice des alignements unitaires : *Argmax*, *Itermax* and *Match*. Les résultats obtenus avec ces méthodes [6], en prenant les annotations hiérarchiques converties en format mot-à-mot comme référence, sont données dans le Tableau 4.

Les alignements mot-à-mot sont évalués à l'aide des métriques standard suivantes : précision, rappel, F-score, AER (pour *Alignment Error Rate*, taux d'erreur d'alignement), respectivement définies par :

$$\text{Pr} = \frac{|A \cap (S \cup P)|}{|A|}, \text{Ra} = \frac{|A \cap S|}{|S|}, F_1 = \frac{2\,\text{Pr} \times \text{Ra}}{\text{Pr} + \text{Ra}}, \text{AER} = 1 - \frac{|A \cap S| + |A \cap (S \cup P)|}{|A| + |S|}$$

où A est l'ensemble des alignements prédits et S et P sont respectivement les ensembles d'alignements sûrs et possibles de référence.

On observe que les résultats d'alignement sur les cinq nouvelles sont plutôt homogènes ce qui indique qu'il n'y avait pas de différences majeures dans les annotations divisives. La méthode Argmax, qui favorise la précision, est celle qui conduit aux meilleurs scores d'AER.

Comme discuté par Ngo Ho (2021), la profusion de liens possibles a pour effet de biaiser la métrique d'AER en faveur des méthodes d'alignement qui privilégient la précision sur le rappel. C'est bien ce que l'on observe : les résultats sur notre ressource sont en moyenne un peu moins bon que sur le

6. Le modèle est `xlm-roberta-base`, les alignements sont calculés au niveau des mots, les autres paramètres de SimAlign ont leur valeur par défaut.

	Argmax				Itermax				Match			
	Pr	Ra	F1	AER	Pr	Ra	F1	AER	Pr	Ra	F1	AER
HANS	96,0	90,6	93,2	6,4	91,0	93,4	92,2	8,0	85,3	93,4	89,2	11,7
EPPS	94,2	71,0	81,0	18,5	88,9	74,4	81,0	18,6	84,3	75,0	79,4	20,3
Alibi	94,8	89,0	91,8	7,7	90,0	93,8	91,4	8,5	83,5	95,6	89,1	12,0
CHA	95,8	92,2	94,0	5,8	90,7	93,7	92,2	8,1	84,6	97,1	90,4	10,5
BAR	93,7	89,9	91,8	7,9	88,7	94,3	91,4	9,1	80,7	95,5	87,5	14,0
VIS	94,7	88,0	91,2	8,3	90,4	93,3	91,8	8,4	83,2	94,2	88,4	12,6
DER	94,3	86,9	90,4	8,9	89,8	91,5	90,6	9,6	84,6	95,0	89,5	11,7
AUB	95,7	88,1	91,7	7,7	90,4	96,1	93,2	7,2	84,3	96,1	89,8	11,2

TABLE 4 – Résultats d'alignement au niveau des mots avec SimAlign pour les corpus de référence Hansards et Europarl et pour notre nouvelle ressource, avec les détails oeuvre par oeuvre. Une valeur plus faible de l'AER indique un meilleur alignement.

Hansards, mais bien meilleurs que pour Europarl, illustrant l'effet du nombre de liens possibles et de mots non alignés sur les scores absolus d'alignement.

4 Algorithme d'alignement divisif

4.1 Version initiale

L'algorithme divisif de Lardilleux *et al.* (2012) construit récursivement une segmentation jointe de phrases parallèles, en se basant sur les scores d'associations lexicales bilingues dérivés de leurs cooccurrences dans un corpus parallèle. Cet algorithme a été initialement introduit pour la segmentation d'images par Shi & Malik (2000) et ensuite utilisé avec succès pour la segmentation de documents (Zha *et al.*, 2001). Nous développons des variantes de cet algorithme, en l'appliquant aux mêmes matrices de similarité que celles qu'exploite SimAlign, pour reproduire les arbres d'alignement. L'algorithme complet se trouve en annexe A, avec une discussion de sa complexité.

De manière similaire à la procédure manuelle présentée §2.2, cet algorithme divise récursivement chaque couple de segments (S, T) en deux parties $S = A\bar{A}$ et $T = B\bar{B}$, puis aligne les nouveaux sous-segments soit de manière *monotone* (A avec B, $\bar{A}$ avec $\bar{B}$), soit de manière *anti-monotone* (A avec $\bar{B}$ et $\bar{A}$ avec B).

Chaque étape de l'algorithme implique deux décisions : (a) le choix du point de segmentation et (b) l'orientation associée. Pour optimiser ces choix, nous parcourons tous les indices possibles dans les deux segments et calculons le score `NCut` pour les deux orientations possibles. Pour expliciter ce calcul, nous commençons par définir le score d'association $W(S, T)$ du bitexte (S, T) selon $W(S, T) = \sum_{s \in S, t \in T} w(s, t)$, avec $w(s, t)$ une mesure d'association lexicale qui peut être définie au niveau des mots ou des unités sous-lexicales. Une première approche sélectionne la segmentation optimisant la coupure (`cut`) définie par [7] :

$$\texttt{cut}(A, B) = W(A, \bar{B}) + W(\bar{A}, B)$$

7. Ce critère étant symétrique en (A, B) et $(\bar{A}, \bar{B})$, il suffit de le minimiser sur l'un de deux blocs.

La meilleure segmentation monotone (resp. anti-monotone) minimise $\texttt{cut}(A, B)$ (resp. $\texttt{cut}(A, \bar{B})$) sur tous les points de coupure possibles ; on choisit finalement la meilleure des deux orientations. Cette approche conduit à minimiser les scores d'association des segments qui sont exclus de l'alignement ($(A, \bar{B})$ et $(\bar{A}, B)$ pour une segmentation monotone). Cependant, son utilisation tend à produire des segments déséquilibrés (Shi & Malik, 2000), puisque des segments non-alignés plus courts auront une association globale plus faible que des segments plus longs. C'est pourquoi Lardilleux *et al.* (2012) trouvent préférable de minimiser le score `NCut` défini (pour une segmentation monotone) par :

$$\texttt{Ncut}(A, B) = \frac{\texttt{cut}(A, B)}{\texttt{cut}(A, B) + 2 \times W(A, B)} + \frac{\texttt{cut}(\bar{A}, \bar{B})}{\texttt{cut}(\bar{A}, \bar{B}) + 2 \times W(\bar{A}, \bar{B})},$$

ce qui conduit à des segmentations plus équilibrées. L'algorithme termine lorsque l'un des segments a une longueur égale à 1. Le résultat final est un arbre binaire qui représente la structure hiérarchique des phrases parallèles ainsi que leurs alignements. En utilisant les heuristiques de conversion présentées en 3.1, on peut au besoin se ramener à des alignements de mots.

4.2 Introduction de similarités locales

L'algorithme de segmentation repose intégralement sur des calculs de similarité lexicales impliquées par $W(S, T)$: pour pouvoir comparer nos résultats avec ceux de SimAlign, nous utilisons comme score $w(s, t)$ le cosinus des plongements [8] associés respectivement à s et t, calculés en moyennant les plongements des unités sous-lexicales qui les composent. Comme le montrent Radovanovic *et al.* (2010), certains vecteurs dans l'espace de représentation des plongements lexicaux sont des *hubs* autour desquels d'autres vecteurs se concentrent : ces hubs sont les plus « populaires » parmi les plus proches voisins calculés par la similarité cosinus. En plus du phénomène de *hubness*, les données de grande dimension présentent également une tendance à l'uniformisation des similarités et concentration de distances entre toutes les paires de points, qui fait que tous les points sont presque également proches les uns des autres (Aggarwal *et al.*, 2001). De ce fait, la plage des scores de similarité sous-jacents aux alignements est assez étroite.

Pour atténuer ce phénomène, on remplace la similarité cosinus par une mesure locale, CSLS [9], qui ajuste la similarité cosinus entre deux unités source et cible x et y en fonction de la similarité moyenne de x (resp. y) avec son voisinage dans l'espace cible (resp. source). Cette opération a pour effet de « dilater »les régions où la densité de points est élevée, éloignant ainsi les *hubs* de leurs voisins (Conneau *et al.*, 2017; Joulin *et al.*, 2018). CSLS est défini par

$$\text{CSLS}(x, y) = 2\cos(x, y) - \frac{1}{k} \sum_{y' \in \mathcal{N}_y(x)} \cos(x, y') - \frac{1}{k} \sum_{x' \in \mathcal{N}_x(y)} \cos(x', y), \qquad (1)$$

avec $\mathcal{N}_u$ l'ensemble des k-plus proches voisins de u.

Un autre changement apporté au calcul des similarités lexicales consiste à ajouter un paramètre de température, qui a pour effet d'amplifier les différences entre similarités. On remplace donc $\text{CSLS}(x, y)$ par $\text{CSLS}(x, y)^\tau$, avec $\tau \in [2 : 10]$, ce qui augmente l'impact des hautes valeurs de similarité et diminue les plus basses. Une comparaison de différents calculs des matrices de similarité est représenté sur la Figure 3.

8. Calculés, de nouveau, par `xlm-roberta-base` ou `bert-base-multilingual-cased`.
9. Pour *Cross-domain Similarity Local Scaling*.

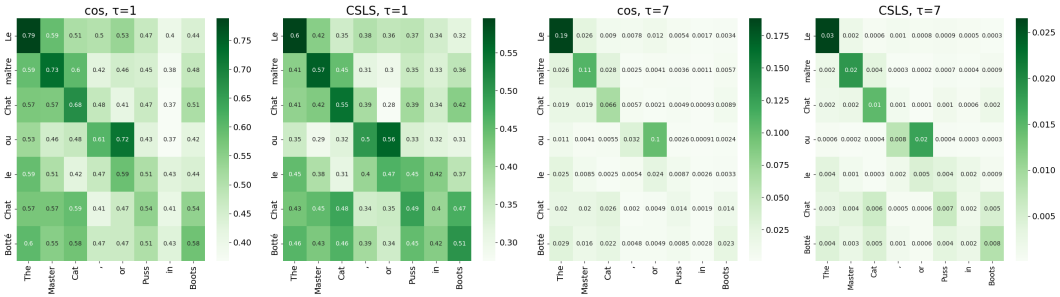

FIGURE 3 – Visualisation de différentes matrices de similarité. L'utilisation de CSLS, avec un paramètre de température élevé, permet de concentrer les similarités sur la diagonale, où sont localisés les alignements corrects pour cet exemple.

4.3 Seuil minimal de similarité

Une méthode pour améliorer la précision de l'algorithme consiste à introduire un seuil minimal de similarité ε dans une étape de post-traitement. Si la similarité entre deux mots est inférieure à ce seuil, l'alignement est rejeté. Ce seuillage est appliqué séparément aux similarités entre chaque alignement mot-à-mot inféré d'une feuille. L'introduction de ce seuillage permet d'éviter l'alignement de certains mots, ce qui n'est pas possible avec la version de base de l'algorithme.

4.4 Résultats

Afin de comparer les arbres obtenus à partir de l'algorithme d'alignement divisif aux arbres de référence, on représente chaque arbre sous la forme d'une liste de couples de segments alignés : au plus haut niveau le couple de phrases parallèles, au plus bas les feuilles de l'arbre. Pour comparer un arbre de référence avec un arbre prédit, on utilise alors les métriques classiques (précision, rappel, F-score) sur ces deux listes, en considérant pour les calculs :

— tous les segments ;
— uniquement les feuilles ;
— uniquement les segments non-feuilles.

Avec les améliorations introduites dans les sections 4.2 et 4.3, les résultats d'évaluation sur le format mot-à-mot et sur les arbres d'alignement sont dans le Tableau 5. Les méta-paramètres ont été optimisés sur *Le Chat Botté*, qui a servi de corpus de développement. Les précisions sur la recherche de méta-paramètres se trouvent en annexe C.

Les scores d'alignement mot-à-mot restent bien en deçà de ceux obtenus avec SimAlign (Tableau 4), essentiellement du fait d'un trop faible rappel, puisque la précision est dans l'ensemble très satisfaisante. Une indication est que notre algorithme ne prédit que 16 455 liens, quand le corpus total en comprend plus de 22 500 (Tableau 2). Ceci suggère que la procédure divisive automatique produit des segments trop fins [10]. Le seuil de similarité choisi est un autre facteur qui privilégie trop fortement la précision au détriment du rappel. Pour ce qui concerne la métrique de comparaison d'arbre, on constate que tous les scores sont assez faibles, en particulier pour les résultats relatifs aux segments

10. Les alignements fins produisent moins de liens que les alignements grossiers, qui construisent un nombre quadratique de liens en fonction de leur longueur : un alignement 1-1 produit 1 lien, un alignement 2-2 en produit 4, etc.

	mot-à-mot				tous les segments			feuilles			non-feuilles		
	Pr	Ra	F1	AER	Pr	Ra	F1	Pr	Ra	F1	Pr	Ra	F1
base	58,2	71,5	64,1	37,4	29,3	32,1	30,6	40,0	47,2	43,2	15,8	16,3	16,0
best	**93,2**	**76,6**	**85,0**	**13,8**	32,2	34,7	33,5	46,1	53,0	49,3	15,6	15,7	15,6
BAR	91,0	77,1	83,5	15,4	28,0	30,4	29,2	40,2	48,4	43,9	12,2	12,0	12,1
VIS	94,9	78,0	85,6	13,1	34,6	36,6	35,6	49,4	55,3	52,2	17,0	17,1	17,0
DER	93,8	79,5	86,1	12,6	32,3	36,3	34,2	45,8	53,9	49,5	16,0	17,1	16,5
AUB	93,2	77,7	84,7	14,1	34,0	35,6	34,8	48,8	54,5	51,5	17,0	16,7	16,8
CHA	93,2	77,4	84,6	14,4	33,5	34,7	34,1	50,0	56,0	52,8	14,1	13,6	13,8

TABLE 5 – Les résultats moyens de l'algorithme divisif sur le corpus de test, obtenus avec la version initiale (*base* : $\tau = 1, \varepsilon = 0.0, \text{sim} = \cos, \text{model} = \text{mbert}$) et la meilleure version (*best* : $\tau = 7, \varepsilon = 0.008, \text{sim} = \text{csls}, k = 2, \text{model} = \text{mbert}$). Pour le meilleur jeu de paramètres on donne également le détail par œuvre. On compare les alignements mot-à-mot, ainsi que les résultats correspondants pour comparaisons exactes des nœuds des arbres de référence avec les arbres générés.

non-feuilles. Les segments initiaux produits par l'algorithme restent parfois très déséquilibrés en dépit du terme de normalisation introduit par `Ncut`, ce qui fait que les segments intermédiaires sont rarement retrouvés [11]. Une exploration d'alternatives à `NCut` est dans l'annexe B. Ces faibles valeurs sont également liées à la procédure d'évaluation très stricte implémentée dans la métrique, qui ne vérifie que les correspondances exactes : elle considère donc qu'un segment comme (0,5-0,5) est incorrect aussi bien quand la référence est (0,5-0,4), que quand elle est (0,5-10,20). Si les alignements internes étaient aussi erronés que la métrique le suggère, les scores pour les feuilles seraient au moins aussi mauvais ; or, ils sont nettement meilleurs. Une amélioration serait alors de mieux prendre en compte les correspondances partielles, par analogie avec les propositions de Véronis & Langlais (2000) pour l'alignement de phrases.

5 Conclusion et perspectives

Dans cet article, nous avons introduit un corpus annoté d'alignements hiérarchiques, également disponibles dans un format « mot-à-mot ». La méthodologie d'annotation hiérarchique présente l'avantage de fournir des alignements de groupes de mots, qui, couplés aux annotations syntaxiques disponibles, permettront de mieux analyser les divergences entre langues dans des textes parallèles. Cette ressource, librement disponible, peut servir comme un jeu de test pour des systèmes d'alignement des mots, ajoutant aux données existantes un autre genre textuel et une autre procédure d'annotation manuelle des alignements.

Nous avons ensuite utilisé un algorithme divisif s'appuyant sur des plongements multilingues pour reproduire ces annotations. Comme noté dans la section 4.4, la fonction `NCut` produit souvent des segments déséquilibrés. Nos résultats confirment que le remplacement de la similarité cosinus avec CSLS, en utilisant le méta-paramètre de température, rend les différences entre les valeurs dans les matrices de similarité plus prononcées, ce qui fait aussi que ces valeurs sont mieux alignées avec les intuitions humaines. Cette étude souligne l'importance de la recherche sur les propriétés

11. Il arrive ainsi souvent que le premier niveau de segmentation isole un seul mot – par exemple la ponctuation finale – du reste de la phrase.

de l'espace des plongements, dans le but de rendre la plage des similarités plus large et rendre cet espace plus facilement interprétable. Atténuer les phénomènes de concentration des distances et de *hubness* pourrait rendre de nombreux outils de TAL plus performants. Au final, ces expériences avec l'algorithme divisif ont apporté une amélioration significative de plus de 20 points de pourcentage d'AER par rapport à sa version de base.

Ce travail offre plusieurs perspectives. Du point de vue algorithmique, nous souhaitons poursuivre l'analyse approfondie de `NCut` et de ses variantes. Cela permettrait de produire des segments plus équilibrés. Du point de vue des similarités, au-dela de la question de la métrique de comparaison, nous souhaitons évaluer l'utilisation de plongements calculés directement au niveau des segments, plutôt que les déduire des unités qui les composent. Les plongements de phrases comme ceux calculés avec LASER (Artetxe & Schwenk, 2019) ou LaBSE (Feng *et al.*, 2022) pourraient être utilisés à cet effet, avec toutefois un surcoût computationnel important, du fait de l'impossibilité d'utiliser des méthodes de programmation dynamique.

Une autre extension de ce travail consistera à explorer des métriques alternatives pour la comparaison d'arbres. Au lieu de rechercher des correspondances exactes entre les nœuds, d'autres caractéristiques pourraient être comparées, par exemple la profondeur ou le niveau d'équilibre des arbres. Prendre en compte la différence entre les tailles des segments pourrait limiter la chute des scores lorsque une ou deux unités se retrouvent dans le segment opposé. Une alternative sera de considérer une métrique comme *Tree Edit Distance* (TED, Schwarz *et al.*, 2017).

Remerciements

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2024-AD011011717R3 attribuée par GENCI. Ces travaux ont été réalisés dans le cadre du programme « Langues et numérique » 2016-2017 de la Délégation Générale à la Langue Française et aux langues de France.
Nous souhaitons remercier les relecteurs et relectrices pour leurs commentaires, Yong Xu pour sa gestion de la collecte des alignements, et Doriana Rüesch qui a réalisé les annotations manuelles.

Références

AGGARWAL C. C., HINNEBURG A. & KEIM D. A. (2001). On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, p. 420–434, Berlin, Heidelberg : Springer-Verlag.

ALKHOULI T. & NEY H. (2017). Biasing attention-based recurrent neural networks using external alignment information. In O. BOJAR, C. BUCK, R. CHATTERJEE, C. FEDERMANN, Y. GRAHAM, B. HADDOW, M. HUCK, A. J. YEPES, P. KOEHN & J. KREUTZER, Éds., *Proceedings of the Second Conference on Machine Translation*, p. 108–117, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4711.

ARTETXE M. & SCHWENK H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, **7**, 597–610. DOI : 10.1162/tacl_a_00288.

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

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2020). Unsupervised cross-lingual representation learning at scale. In D. JURAFSKY, J. CHAI, N. SCHLUTER & J. TETREAULT, Éds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.747.

CONNEAU A., LAMPLE G., RANZATO M., DENOYER L. & JÉGOU H. (2017). Word translation without parallel data. *arXiv preprint arXiv :1710.04087*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In J. BURSTEIN, C. DORAN & T. SOLORIO, Éds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DOU Z.-Y. & NEUBIG G. (2021). Word alignment by fine-tuning embeddings on parallel corpora. In P. MERLO, J. TIEDEMANN & R. TSARFATY, Éds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 2112–2128, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-main.181.

FENG F., YANG Y., CER D., ARIVAZHAGAN N. & WANG W. (2022). Language-agnostic BERT sentence embedding. In S. MURESAN, P. NAKOV & A. VILLAVICENCIO, Éds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 878–891, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.62.

GARG S., PEITZ S., NALLASAMY U. & PAULIK M. (2019). Jointly learning to align and translate with transformer models. In K. INUI, J. JIANG, V. NG & X. WAN, Éds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 4453–4462, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1453.

GRAÇA J., PARDAL J. P., COHEUR L. & CASEIRO D. (2008). Building a golden collection of parallel multi-language word alignment. In N. CALZOLARI, K. CHOUKRI, B. MAEGAARD, J. MARIANI, J. ODIJK, S. PIPERIDIS & D. TAPIAS, Éds., *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA).

JALILI SABET M., DUFTER P., YVON F. & SCHÜTZE H. (2020). SimAlign : High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, p. 1627–1643, Online.

JOULIN A., BOJANOWSKI P., MIKOLOV T., JÉGOU H. & GRAVE E. (2018). Loss in translation : Learning bilingual word mapping with a retrieval criterion. In E. RILOFF, D. CHIANG, J. HOCKENMAIER & J. TSUJII, Éds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2979–2984, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-1330.

KAESHAMMER M. & WESTBURG A. (2014). On complex word alignment configurations. In N. CALZOLARI, K. CHOUKRI, T. DECLERCK, H. LOFTSSON, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 1773–1780, Reykjavik, Iceland : European Language Resources Association (ELRA).

LARDILLEUX A., YVON F. & LEPAGE Y. (2012). Hierarchical sub-sentential alignment with anymalign. In M. CETTOLO, M. FEDERICO, L. SPECIA & A. WAY, Éds., *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, p. 279–286, Trento, Italy : European Association for Machine Translation.

LIU X., DUH K. & MATSUMOTO Y. (2013). Topic models + word alignment = a flexible framework for extracting bilingual dictionary from comparable corpus. In J. HOCKENMAIER & S. RIEDEL, Éds., *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, p. 212–221, Sofia, Bulgaria : Association for Computational Linguistics.

MELAMED I. D. (2001). *Manual Annotation of Translational Equivalence*, In *Empirical Methods for Exploiting Parallel Texts*, p. 65–77.

NGO HO A. K. (2021). *Generative Probabilistic Alignment Models for Words and Subwords : a Systematic Exploration of the Limits and Potentials of Neural Parametrizations*. Theses, Université Paris-Saclay. HAL : tel-03210116.

OCH F. J. & NEY H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, **29**(1), 19–51.

RADOVANOVIC M., NANOPOULOS A. & IVANOVIC M. (2010). Hubs in space : Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, **11**, 2487–2531.

SCHWARZ S., PAWLIK M. & AUGSTEN N. (2017). A new perspective on the tree edit distance. In C. BEECKS, F. BORUTTA, P. KRÖGER & T. SEIDL, Éds., *Similarity Search and Applications*, p. 156–170, Cham : Springer International Publishing.

SHI J. & MALIK J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 888–905. DOI : 10.1109/34.868688.

SØGAARD A. (2010). Can inversion transduction grammars generate hand alignments. In F. YVON & V. HANSEN, Éds., *Proceedings of the 14th Annual Conference of the European Association for Machine Translation*, Saint Raphaël, France : European Association for Machine Translation.

STAHLBERG F., SAUNDERS D. & BYRNE B. (2018). An operation sequence model for explainable neural machine translation. In T. LINZEN, G. CHRUPAŁA & A. ALISHAHI, Éds., *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 175–186, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5420.

TIEDEMANN J. (2011). *Bitext alignment*. Synthesis lectures on human language technologies. Morgan & Claypool Publishers.

TOUTANOVA K., KLEIN D., MANNING C. D. & SINGER Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, p. 252–259.

VARGA D., HALÁCSY P., KORNAI A., NAGY V., NÉMETH L. & TRÓN V. (2007). Parallel corpora for medium density languages. In N. NICOLOV, K. BONTCHEVA, G. ANGELOVA & R. MITKOV, Éds., *Selected papers from RANLP 2005*, p. 247–258 : John Benjamins Publishing Company. DOI : doi :10.1075/cilt.292.32var.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is All you Need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 30*, NeurIPS, p. 5998–6008 : Curran Associates, Inc.

VÉRONIS J. & LANGLAIS P. (2000). Evaluation of parallel text alignment systems. Text speech and language technology series, p. 369–388. Kluwer Academic Publishers.

WANG W., ZHU D., ALKHOULI T., GAN Z. & NEY H. (2018). Neural hidden Markov model for machine translation. In I. GUREVYCH & Y. MIYAO, Éds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 377–382, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-2060.

WU D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, **23**(3), 377–403.

WU Q., NAGATA M., MIAO Z. & TSURUOKA Y. (2024). Word alignment as preference for machine translation. In Y. AL-ONAIZAN, M. BANSAL & Y.-N. CHEN, Éds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, p. 3223–3239, Miami, Florida, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2024.emnlp-main.188.

WU Q., NAGATA M. & TSURUOKA Y. (2023). WSPAlign : Word Alignment Pre-training via Large-Scale Weakly Supervised Span Prediction. In A. ROGERS, J. BOYD-GRABER & N. OKAZAKI, Éds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 11084–11099, Toronto, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/2023.acl-long.621.

XU Y. & YVON F. (2016). Novel elicitation and annotation schemes for sentential and sub-sentential alignments of bitexts. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, M. GROBELNIK, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Ninth Language Resources and Evaluation Conference (LREC 2016)*, p.1̃0, Portorož, Slovenia : European Language Resources Association (ELRA).

ZHA H., HE X., DING C., SIMON H. & GU M. (2001). Bipartite graph partitioning and data clustering. p. 25–32. DOI : 10.1145/502585.502591.

6 Annexe

A Algorithme d'alignement divisif

Étant donné un couple de phrases (S, T) de longueurs respectives I et J, l'algorithme divisif demandera au plus $\min(I, J)$ étapes de segmentation. Chaque étape de segmentation demande d'évaluer $O(I \times J)$ points de segmentation ; chaque évaluation demandant de sommer $O(I \times J)$ scores d'association lexicale $w(s_i, t_j)$ (cf. section 4.1). Au total, en supposant que $J < \alpha I$, où α est une constante proche de 1, ce qui est une hypothèse raisonnable pour des phrases parallèles, la complexité sera $O(I^5)$. En pré-calculant la table auxiliaire W_c dont le terme général est $W(i, j) = \sum_{i' \leq i, j' \leq j} w(s_i, t_j)$, ce qui peut être fait une fois pour toutes avec une complexité $O(I^2)$ par programmation dynamique, l'évaluation de chaque point de segmentation peut être réalisée en temps constant, ce qui ramène la complexité globale à $O(I^3)$.

Algorithm 1 Algorithme divisif

procedure ALIGN(S, T)
  if $\text{length}(S) = 1$ **or** $\text{length}(T) = 1$ **then**
    lier chaque mot de S à chaque mot de T
    stop
  end if
  $minNcut \leftarrow 2$
  $(X, Y) \leftarrow (S, T), I \leftarrow |S|, J \leftarrow |T|$
  for each $(i, j) \in \{2, \ldots, I\} \times \{2, \ldots, J\}$ **do**
    $A \leftarrow [s_1 ... s_{i-1}], \bar{A} \leftarrow [s_i ... s_I]$
    $B \leftarrow [t_1 ... t_{j-1}], \bar{B} \leftarrow [s_j ... s_J]$
    if $Ncut(A, B) < minNcut$ **then**
      $minNcut \leftarrow Ncut(A, B)$
      $(X, Y) \leftarrow (A, B), (\bar{X}, \bar{Y}) \leftarrow (\bar{A}, \bar{B}),$
    end if
    if $Ncut(A, \bar{B}) < minNcut$ **then**
      $minNcut \leftarrow Ncut(A, \bar{B})$
      $(X, Y) \leftarrow (A, \bar{B}), (\bar{X}, \bar{Y}) \leftarrow (\bar{A}, B)$
    end if
  end for
  ALIGN(X, Y)
  ALIGN($\bar{X}$, $\bar{Y}$)
end procedure

B Versions alternatives de `NCut`

La normalisation utilisée dans la fonction `NCut` a été introduite afin d'éviter de choisir des points de coupure produisant des segments déséquilibrés. Cependant, après une analyse des sorties de l'algorithme, il a été observé que, malgré cela, de nombreuses divisions se contentaient de séparer le dernier symbole du reste du segment.

Deux alternatives à la fonction `NCut`, $\texttt{NCut}_2$ et $\texttt{NCut}_3$, issues de (Zha *et al.*, 2001) et (Shi & Malik, 2000), ont alors été testées. Une visualisation du calcul de ces valeurs est dans la Figure 4.

$$\texttt{Ncut}_2(X, Y) = \frac{\text{cut}(X, Y)}{W(X, Y)} + \frac{\text{cut}(\bar{X}, \bar{Y})}{W(\bar{X}, \bar{Y})} \tag{2}$$

$$\texttt{Ncut}_3(X, Y) = \frac{\text{cut}(X, Y)}{W(X, Y) + W(X, \bar{Y})} + \frac{\text{cut}(X, Y)}{W(X, Y) + W(\bar{X}, Y)} \tag{3}$$

Les résultats obtenus avec $\texttt{NCut}_3$ étaient très bas par rapport à ceux de `NCut` (AER=76,0 sur *Le Chat Botté*). En revanche, les scores de $\texttt{NCut}_2$ étaient quasiment identiques à ceux de `NCut`, avec des écarts de seulement 0,3 points d'AER.

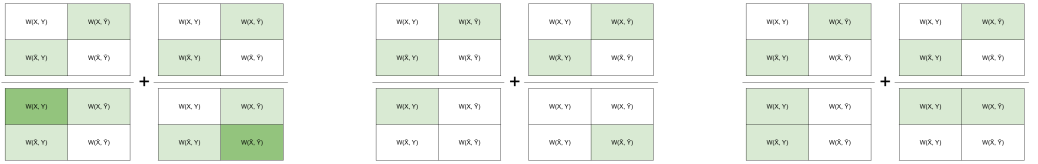

FIGURE 4 – Visualisation des fonctions `NCut`, $\texttt{NCut}_2$ and $\texttt{NCut}_3$ respectivement.

C Exploration des méta-paramètres

Les valeurs de paramètres testés :

1. température τ : $\tau \in [2 : 10]$
2. seuil de similarité minimale ε : $\varepsilon \in [0 : 0.4]$
3. similarité *sim* : $\text{sim} \in \{\cos, \text{CSLS}\}$
4. nombre de plus proches voisins *k* (seulement pour CSLS) : $k \in [1 : 4]$
5. modèle utilisée pour calculer les plongements *model* : $\text{model} \in \{\text{mbert}, \text{xlmr}\}$
6. variant de la fonction de normalisation *ncut* : $\text{ncut} \in \{\text{ncut}, \text{ncut2}, \text{ncut3}\}$

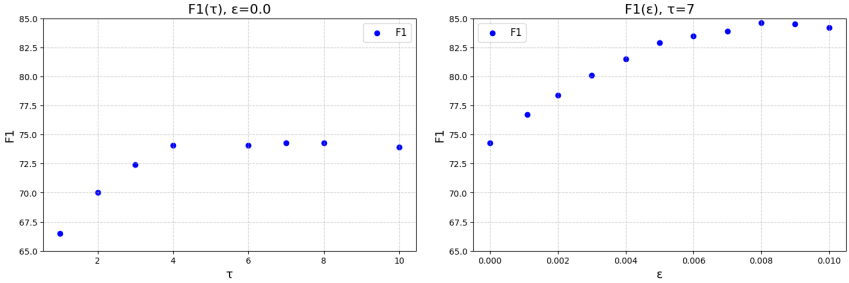

FIGURE 5 – Résultats de recherche de meilleurs valeurs de F1 score en variant d'abord la température, ensuite le seuil ε avec la température fixée.

La version initiale :

$\tau = 1, \text{sim} = \cos, \varepsilon = 0.0, k = 2, \text{model} = \text{mbert}, \text{ncut} = \text{ncut}$

La meilleure version trouvée :

$\tau = 7, \text{sim} = \text{CSLS}, \varepsilon = 0.008, k = 2, \text{model} = \text{mbert}, \text{ncut} = \text{ncut}.$

On a observé des performances légèrement meilleures avec `mbert`, néanmoins, les différences entre les deux modèles n'étaient pas très prononcées. Les résultats de recherche d'ε et τ sont dans la Figure 5.