Topologie et génétique textuelles : un dialogue médié par la machine

Julien Bourdaillet, Jean-Gabriel Ganascia

Laboratoire d'Informatique de Paris 6

Université Pierre et Marie Curie

104 avenue du président Kennedy - 75016 Paris

Jean-Louis Lebrave

Institut des Textes et Manuscrits Modernes

Ecole Normale Supérieure

45 rue d'Ulm – 75005 Paris

ABSTRACT. *A joint work between genetic criticism and informatics involved the development of the software application MÉDITE. Its algorithmic foundations rely only on the character sequence composing the text. These foundations, as well as their close relationship with the notions of textual topology and topography, are presented.*

KEYWORDS : *Textual Topology, Genetic Criticism, Stringology, Monolingual Alignment, Neighbourhood, Sequence, Network.*

RESUME. *A partir de nos travaux conjoints en critique génétique et informatique, nous avons développé un logiciel d'alignement textuel nommé MÉDITE. Les principes algorithmiques sous-jacents considèrent le texte comme une simple séquence de caractères. Nous présentons ces principes et montrons en quoi ceux-ci sont intrinsèquement basés sur des notions de topologie et topographie textuelles.*

MOTS-CLES : *topologie textuelle, critique génétique, algorithmique textuelle, alignement monolingue, voisinage, séquence, réticularité.*

1. Introduction

Nous avons développé un logiciel d'alignement textuel nommé MÉDITE. Dans ce cadre, les concepts de la critique génétique ont été modélisés informatiquement grâce à l'algorithmique textuelle. Il en résulte une approche originale des études textuelles diachroniques, à la fois linéaire et réticulaire. Cet article montre comment l'analyse des données textuelles peut s'enrichir de cette approche qui s'inscrit pleinement dans le cadre émergent de la topologie textuelle.

Dans la section 2, nous présentons les spécificités de l'aide informatisée à la critique génétique. Dans la section 3, les outils de l'algorithmique textuelle sous-jacents à l'algorithme de MÉDITE sont introduits. Enfin dans la section 4, nous détaillons plus spécifiquement la convergence avec la topologie textuelle.

2. Aide informatisée à la critique génétique

2.1 Des corpus

La critique génétique est l'étude des brouillons d'écrivains [Bia00]. A partir des différents états du texte, le généticien tente de reconstituer le cheminement créatif de l'auteur. Ce dernier lors des phases de réécriture annote, biffe et rature sa page manuscrite. C'est l'étude et l'analyse diachronique de ces marques scripturales qui permettent d'appréhender et de comprendre la genèse de l'œuvre.

Le matériau d'étude du généticien est constitué d'un ensemble de brouillons d'une œuvre d'un écrivain, soit le dossier génétique de l'œuvre. Une fois le dossier classé et ordonné chronologiquement, on obtient l'avant-texte : l'ensemble des différents états (ou versions) du texte. En étudiant, analysant et interprétant ceux-ci, le généticien est alors à même de reconstituer la genèse de l'œuvre.

L'étude manuelle de l'avant-texte est un long labeur qui va croissant avec le nombre de brouillons et leur taille. Nos travaux portent sur la production d'outils informatiques facilitant les études génétiques. Afin d'obtenir une version numérique des brouillons qui soit traitable informatiquement, nous nous appuyons sur l'hypothèse qu'une transcription linéarisée des brouillons permet d'en simuler le caractère diachronique. Comme le remarque Pierre-Marc de Biasi [2000, p.65], « ce mode de transcription ne reproduit pas la mise en page autographe ni les phénomènes génétiques (ratures, ajouts, insertions, etc.) mais reconstitue une image des étapes successives de l'écriture. » En revanche, à partir de telles transcriptions linéarisées, un outil informatique est capable d'exhiber automatiquement les phénomènes génétiques décrivant le passage d'un état du texte au suivant.

Ces études textuelles sont donc des études en diachronie de corpus fortement cohérents consistant en l'ensemble des états d'un texte (dans l'idéal, du premier jet sur le papier à la version éditée, voire au-delà). On peut les assimiler à un cas particulier des séries textuelles

chronologiques [HNS97]. En effet, leur caractère génétique peut être interprété comme une contrainte de cohérence forte, et même très forte, entre les différents textes du corpus.

Il est intéressant de présenter les corpus génétiques sous ce jour ; en effet, on constate ainsi que l'objet d'étude est très proche de celui de l'Analyse des Données Textuelles (ADT). Habituellement, cette discipline traite des corpus d'œuvres littéraires d'une part, et des corpus de discours politiques ou syndicaux d'autre part. Contrastivement, nos corpus sont des corpus d'œuvre littéraire, mais au singulier cette fois. Par ailleurs, les études de corpus sociopolitiques, du moins dans leur acceptation par la tradition de l'ENS Fontenay Saint-Cloud, ont montré que ceux-ci consistent fréquemment en des réécritures de certaines idées ou thèmes. Ainsi, on pourrait être tenté de les considérer comme une forme de corpus génétique. Nous nous garderons de franchir ce pas qui mériterait une étude spécifique. Néanmoins, nous conserverons un intérêt certain pour l'intuition sous-jacente qui rapproche ces deux types de corpus.

On remarquera que la méthode d'analyse génétique de corpus textuel n'est en rien limitée aux corpus littéraires. En effet, de la même manière, peuvent être étudiés des corpus où la contrainte de cohérence est moins forte ; tels les corpus habituels d'ADT mentionnés ci-dessus. Néanmoins les travaux présentés dans cet article donnent de meilleurs résultats sur des corpus ayant une cohérence minimale. Nous définissons celle-ci comme une parenté phylogénétique entre les textes du corpus, de façon à ce qu'au minimum, les textes composant le corpus soient une transformation ou une réécriture de l'un vers l'autre. Par exemple, on peut considérer qu'il existe une telle parenté entre des notes prises lors d'une réunion et un compte-rendu de la même réunion rédigé à partir de ces notes. Cette contrainte de cohérence est plus forte que celle des séries textuelles chronologiques habituellement considérées en ADT.

2.2 Et des traitements

Une fois l'objet d'étude cerné, reste à définir les outils. La critique génétique recense quatre opérations de réécriture : les insertions, suppressions, remplacements et déplacements. Ces opérations sont relatives à des segments de texte de taille variable, du simple caractère à une séquence de mots, de paragraphes, voire de pages.

Nos travaux portent sur l'identification automatique de ces opérations entre deux états du texte. Ces états doivent auparavant avoir été linéarisés (un brouillon peut contenir des ratures hors-linéarité) et être disponibles sous forme électronique. Nous avons développé le logiciel MÉDITE qui se charge de cette identification automatique [GB06 ; BG07].

Ainsi, étant donné deux états du texte, MÉDITE recense automatiquement les suppressions, insertions, remplacements, et déplacements de segments de texte. Recenser l'ensemble de ces opérations revient finalement à établir un alignement entre les deux états. La particularité de cet alignement est que les déplacements sont identifiés de façon fiable, ce qui n'existe dans aucun autre logiciel d'alignement.

En effet, mis à part dans le champ de la critique génétique, l'étude des déplacements entre deux états textuels n'a pas suscité beaucoup d'intérêt, ni en ADT, ni plus largement en informatique. C'est pourquoi l'on ne dispose pas d'outil d'alignement textuel avec recherche de déplacements.

Or nous montrons dans les lignes qui suivent que la recherche des déplacements contribue particulièrement au renouvellement de la notion d'alignement, et à l'inscription de ce travail dans le champ de la topologie textuelle.

3. Algorithmique textuelle

D'un point de vue informatique, l'algorithme qui constitue le cœur de MÉDITE est basé sur les travaux issus d'une branche de l'informatique théorique, appelée algorithmique textuelle [CHL01]. Celle-ci considère le texte comme une simple séquence de caractères, sans plus d'hypothèse linguistique. C'est-à-dire que le texte est brut et que l'on ne tire pas profit des informations qu'apporteraient un étiqueteur morpho-syntaxique, un lemmatiseur ou encore un analyseur syntaxique. Mais plus encore, le texte n'est même pas segmenté en mots à l'aide des séparateurs et signes de ponctuation comme habituellement en ADT ; une simple séquence de caractères constitue donc l'entrée de notre algorithme (ou plus précisément, deux séquences correspondant aux deux textes à aligner).

Nous invitons le lecteur à prêter attention à cette considération préliminaire. En effet, c'est à ce point précis que se situe la divergence entre nos travaux et les travaux habituels d'ADT : le texte comme une simple séquence de caractères *vs* le texte comme un schéma d'urne. Par contre, on remarquera que ce qui rapproche ces deux points de vue sur le texte, c'est leur simplicité. En effet, ces deux modèles sont d'une extrême simplicité et font fi des considérations linguistiques, en vue naturellement d'un traitement automatisé qui a pour but d'aider le linguiste.

Une des spécificités de notre approche est donc que la granularité minimale de l'alignement n'est pas le syntagme mais le caractère. Ceci implique l'identification d'opérations de réécriture portant sur des segments inférieurs au syntagme : par exemple, on peut identifier la correction par l'écrivain de fautes d'orthographe entre deux versions d'un texte. Cette spécificité est en accord avec la réalité des réécritures dans les brouillons d'écrivains, qui portent très souvent sur des fragments de mots graphiques.

3.1 Deux types de segments répétés : les invariants et les déplacements

Notre algorithme est basé sur l'identification des segments répétés, notion mise en évidence dans [LS83, LS94]. Un segment est dit répété si il existe une occurrence de celui-ci dans chacun de deux textes que l'on cherche à aligner. Cette notion est appliquée aux séries textuelles chronologiques dans le logiciel Lexico, où l'on cherche à exhiber leurs différentes occurrences. On remarquera également que cette notion a été largement étudiée en algorithmique textuelle, voir [CHL01] ou [Gus97] par exemple.

Dans notre cas, nous affinons la notion en distinguant deux types de segments répétés. D'une part, les segments qui ne sont concernés par aucune des quatre opérations de réécriture, soit les segments invariants ; ceux-ci sont effectivement des segments répétés, et à la même position, dans les deux textes. Ils forment « l'ossature » de l'alignement entre les deux textes. D'autre part, les déplacements ; ceux-ci sont bien des segments répétés, mais déplacés d'une position dans le premier texte vers une autre position dans le second.

Dès lors qu'est introduite la notion de déplacement, la question de leur empan se pose. Comme le souligne Étienne Brunet, les cooccurrences n'ont de sens que sur un empan de taille limitée de l'ordre de la phrase, du paragraphe ou de la page [Bru08]. Examinons ce qu'il en est pour les déplacements.

Dans les textes littéraires, la génétique textuelle a montré que les déplacements consistaient souvent en la libération d'un terme (ou d'une séquence de termes) afin de pourvoir l'utiliser à nouveau dans un voisinage proche. L'auteur travaille ainsi son style. Mais celui-ci peut également réorganiser l'architecture générale de son texte, auquel cas des déplacements sur un empan long prennent tout leur sens. De même, dans un texte d'idées (un article scientifique par exemple), les déplacements sur un vaste empan peuvent apparaître très facilement par le biais d'un réagencement des idées au sein de l'article. Par ailleurs, la taille du segment déplacé est primordiale. En effet, plus la taille de ce segment augmente, plus il est vraisemblable que l'on ait effectivement affaire à un déplacement voulu par l'auteur, et non à un segment répété fortuit résultant de la combinatoire des segments répétés (voir section 3.2).

Repérer ces déplacements non fortuits dans de tels contextes autorise à étudier, par exemple, la réorganisation du récit au sein d'un corpus génétique, ou la construction de l'argumentation scientifique d'un point de vue épistémologique. C'est pourquoi nous avons conçu notre méthode informatique en ne limitant en rien l'empan des déplacements. Or en autorisant l'accroissement de l'empan, la combinatoire explose. C'est là qu'intervient l'usage d'outils algorithmiques puissants.

D'un point de vue algorithmique, les segments répétés sont recherchés en construisant un arbre des suffixes [Ukk95]. Cette structure de données est un arbre (au sens de la théorie des graphes) qui représente un texte de façon compacte. Le texte est considéré, rappelons-le, comme une simple séquence de caractères. Chaque chemin dans l'arbre, de la racine jusqu'à une feuille, représente un suffixe de la séquence. Ainsi, une séquence de n caractères comporte n suffixes de taille croissante de 1 à n. Les feuilles de l'arbre pointent vers les occurrences du texte représentant un suffixe. La particularité des arbres des suffixes est que les segments répétés sont factorisés sur une même sous-branche de l'arbre. Ceci permet alors de les identifier facilement en parcourant l'arbre en profondeur. De plus, des algorithmes très efficaces existent, comme celui d'Ukkonen. Etant donné les deux séquences en entrée, ils permettent de construire ces arbres en un temps de calcul qui est linéaire en fonction de la taille de ces séquences ; de même, l'espace mémoire requis pour stocker ces arbres en mémoire est linéaire en fonction de la taille des séquences. Ceci autorise l'identification des segments répétés dans de grands corpus pour un coût computationnel faible et rend ces algorithmes utilisables pratiquement.

3.2 Segmentation émergente

Les outils lexicométriques traditionnels segmentent le texte en mots ou n-grammes (séquences de n caractères ou n mots). Cette segmentation préalable des textes permet de faire chuter la combinatoire lors du processus d'alignement ou de recherche des segments répétés. Finalement, ceci permet d'obtenir des logiciels capables de traiter rapidement de grands corpus textuels. On peut définir cela comme une segmentation a priori ou statique du texte.

Notre algorithme n'est pas basé sur ce principe de segmentation a priori du texte, mais sur un principe de segmentation émergente du texte suivant les segments répétés. Il recherche automatiquement l'ensemble de tous les segments répétés du plus long au plus court afin de segmenter le texte. Ceci permettrait de segmenter le texte de façon optimale suivant ce principe si ne surgissaient deux problèmes : l'explosion combinatoire du nombre de segments répétés d'une part, et l'existence de nombreux recouvrements entre segments répétés.

Illustrons ces deux problèmes en cherchant à aligner ces deux brefs textes A = [Ce matin le chat observa de petits oiseaux dans les arbres.] et B = [Le chat était en train d'observer des oiseaux dans les petits arbres ce matin. Il observa les oiseaux pendant deux heures.]. Entre A et B, il existe les occurrences de segments répétés suivantes :

- 2 occurrences de taille 19, soit le segment répété [s oiseaux dans les] qui est présent dans A et B aux positions 33 et 36 respectivement ;
- 4 occurrences de taille 18, soit les segments répétés [oiseaux dans les] et [s oiseaux dans les] ;
- 6 occurrences de taille 17, soit les segments répétés [oiseaux dans les], [oiseaux dans les] et [s oiseaux dans le];
- …
- 62 occurrences de taille 6, pour 28 segments répétés différents;
- …
- 149 occurrences de taille 2, pour 45 segments répétés différents;
- 176 occurrences de taille 1, pour 20 segments répétés différents.

Il y a un total de 899 occurrences de segments répétés pour ces deux séquences A et B de 59 et 122 caractères respectivement. On a là une manifestation caractéristique de l'explosion combinatoire se produisant lors de la recherche de segments répétés ; ces deux textes sont pourtant très courts. Dès que l'on a affaire à des corpus plus intéressants, des méthodes algorithmiques très efficaces sont requises, comme celle détaillée dans la section 3.1.

Avec cet exemple, on constate également qu'il existe des recouvrements entre les segments de taille 18 et 19. Dans ce cas, les segments de taille 18 sont inclus dans celui de taille 19. Si notre intérêt se porte sur les segments les plus longs, alors on conservera comme segment répété uniquement celui de taille 19 et les deux segments répétés de taille 18 ne figureront pas dans l'alignement de A et B. On peut aussi remarquer que le segment [s oiseaux dans les] chevauche partiellement le mot [petits] et ne respecte pas la segmentation du texte en mots. On peut ainsi préférer une segmentation qui respecte le découpage en mots, auquel cas on conservera uniquement le segment répété de taille 18 [oiseaux dans les] au détriment du segment de taille 19 et de l'autre segment de taille 18.

Cet exemple illustre deux stratégies d'élimination des recouvrements entre segments répétés : la première favorise les segments de longueur maximale, et la seconde les segments dont les extrémités sont soit des séparateurs (caractère d'espacement ou signe de ponctuation faible ou forte), soit la première ou la dernière lettre d'un mot. D'autres stratégies sont sans nul doute envisageables. Néanmoins, nous nous sommes basés sur celles-ci pour réaliser un algorithme d'élimination automatique des recouvrements entre

segments répétés. Cet algorithme combine ces deux stratégies de manière efficace. Pour cela, les segments répétés de longueur maximale sont extraits de l'arbre des suffixes résultant des deux textes, respectant ainsi la première stratégie. Dans un second temps, les recouvrements existants entre ces segments répétés sont résolus en utilisant la seconde stratégie. Finalement, en appliquant cet algorithme sur les deux textes, émerge une segmentation suivant les segments répétés de longueur maximale. Grâce aux outils de l'algorithmique textuelle, avec une complexité linéaire en temps et en espace, cet algorithme est efficace et permet de traiter de gros corpus, comme ceux de l'ADT.

Enfin, ouvrons un bref aparté historique. La question de la combinatoire sous-jacente aux segments répétés, composés uniquement de mots graphiques, a déjà été soulevée dans les travaux séminaux de Lafon & Salem [LS83]. A l'époque cette combinatoire était déjà source de questionnements relatifs à la stratégie de segmentation du texte. Ceux-ci furent résolus en ne conservant que les segments répétés de longueur maximale. La question soulevée ici est donc peu ou prou la même ; mais ce qui change radicalement, c'est l'ordre de grandeur. En effet, en traitant le texte directement au niveau des caractères, la combinatoire est supérieure d'au moins un ordre de grandeur. En 25 ans, sur cette même question de la combinatoire des segments répétés, en passant des mots graphiques aux caractères, on aura donc gagné un ordre de grandeur en capacité de traitement. Ce gain est dû uniquement aux progrès algorithmiques tels que décrits dans cet article, et non aux progrès des machines qui, elles, accélèrent uniquement l'exécution de ces traitements.

4. Convergence avec la topologie textuelle

Examinons à présent en quoi les notions présentées ci-dessus s'inscrivent dans le cadre de la topologie textuelle.

4.1 Du concept de voisinage

Reprenons le concept topologique de voisinage (voir [Bar65]), illustré dans [MB08] comme étant un nombre fini de d'éléments x_i entourant un point x. Dans le cas qui nous intéresse, x est un élément appartenant au texte vu comme « un ensemble E d'unités linguistiques qui ne sont pas indépendantes les unes des autres, muni d'une structure ou, plus exactement, de plusieurs structures imbriquées dont l'union constitue cet ensemble », d'après [MB08]. Cette définition du texte sied parfaitement à notre propos et nous nous basons sur celle-ci pour présenter l'utilisation du concept de voisinage dans nos travaux.

Un concept unificateur

Les unités linguistiques élémentaires en lesquelles un texte peut être découpé sont généralement les mots ou syntagmes en ADT. Ces unités peuvent s'assembler en *chunks*, groupes (verbaux, nominaux, …), propositions, phrases, paragraphes jusqu'à former un texte. Ces assemblages ont lieu par l'intermédiaire de règles syntaxiques, sémantiques ou pragmatiques qui forment les structures imbriquées mentionnées ci-dessus. Dès lors que le texte est envisagé d'un point de vue opérationnel, les structures considérées se limitent bien souvent aux résultantes d'une analyse syntaxique dans le domaine du Traitement Automatique du Langage Naturel (TALN). En ADT, l'approche est encore plus radicale

puisque, la plupart du temps, seules les unités lexicales sont conservées en entrée des traitements.

Notre approche, telle que présentée depuis le début de cet article, prend le pari inverse des traitements utilisés en ADT en utilisant la structure intrinsèque au texte. Toutefois cette utilisation de la structure est minimale. En effet, les unités linguistiques élémentaires sont ici les caractères, unités effectivement minimales ou atomiques du texte en tant que manifestation écrite de la langue. De fait, la structure associée à cet ensemble de caractères est définie par la relation de séquentialité qui les lie. Ainsi cette simplification nous amène à considérer le texte sous la seule dimension de la séquence de caractères, ce qui permet d'éviter de traiter les autres dimensions syntaxico-sémantiques. Cette réduction permet d'obtenir un algorithme d'alignement tel que présenté dans la section 3.

Comme le soulignent Mellet & Barthélemy, « chaque point x se trouve muni d'une famille de voisinages ». Au vu des considérations des deux paragraphes précédents, nous pouvons même affirmer que chaque point x se trouve muni de plusieurs familles de voisinages ; chaque famille étant relative à un niveau linguistique. Cette unification des niveaux linguistiques sous le concept de voisinage est rendue possible par le fait qu'ils reposent sur la séquentialité du texte. Ces considérations théoriques ont une application très pratique. Si chaque niveau linguistique n'est qu'une déclinaison du concept de voisinage, alors une méthode permettant de traiter un niveau devrait être capable de traiter les autres. En particulier, un algorithme permettant de traiter un niveau serait également capable de traiter les autres niveaux modulo une éventuelle adaptation aux symboles ou codes représentant les unités en entrée.

C'est ce dont est capable l'algorithme de MÉDITE. En effet, nous avons montré qu'il était capable d'aligner les textes au niveau des caractères. Ceci le rend ainsi capable d'aligner les textes au niveau des mots, phrases ou paragraphes directement et sans modification de l'algorithme. De plus, si en entrée de l'algorithme on fournit non plus le texte mais une séquence de codes représentant, par exemple, les parties du discours associées au texte, on obtient alors un alignement relatif aux parties du discours. Cette capacité d'adaptation, propre aux outils de l'algorithmique textuelle, est rendue possible dans le cas présent par la séquentialité du texte.

Un concept opérationnalisé

Limitons-nous à présent à une seule famille de voisinages entourant un point x, et considérons pour cela uniquement le niveau pour lequel MÉDITE a été conçu, le niveau graphique du texte, soit la séquence de caractères.

Chaque caractère x possède une famille de voisinages. La taille de ces voisinages varie en fonction du nombre de caractères précédant et suivant x. Ainsi le plus petit voisinage de x est ce caractère lui-même, et le plus grand l'ensemble du texte. Entre ces deux extremums, il existe un nombre fini de voisinages. Si les voisinages sont de type $]\ x-h_1,\ x+h_2\ [$ avec $h_1 = h_2$, alors le nombre de voisinages de x est linéaire par rapport à la taille du texte. Si $h_1 \neq h_2$ alors le nombre de voisinages de x est quadratique par rapport à la taille du texte. Donc, si on considère l'ensemble des caractères du texte, le nombre de voisinages est au minimum quadratique. Si maintenant on restreint cet ensemble en contraignant les voisinages à être des segments répétés entre deux textes que l'on souhaite comparer, on

réduit considérablement la cardinalité de l'ensemble. Néanmoins, comme l'exemple de la section 3.2 le montre, on est toujours face à une explosion combinatoire.

Notre solution consiste à introduire une contrainte de non-chevauchement. A partir de l'ensemble des voisinages qui sont également des segments répétés, on en extrait un sous-ensemble dont les éléments soient non-chevauchants. Cette nouvelle contrainte de non-chevauchement implique que pour chaque caractère du texte, on conservera au plus un seul voisinage auquel il appartient. Ce voisinage du caractère correspond au segment répété auquel il appartient. De plus, comme l'exemple de la section 3.2 le montre, les segments répétés de taille minimale sont les segments de taille 1. Or ceux-ci ne présentent que peu d'intérêt. On introduit donc un seuil représentant la taille minimale pour les segments répétés. Dans la pratique, nous forçons les segments répétés à avoir une taille d'au moins 3. Les segments ne respectant pas ces trois critères (répétition, non-chevauchement et taille minimale) ne seront pas considérés comme des segments répétés mais comme des insertions, suppressions ou remplacements. Finalement, comme les segments répétés sont non-chevauchants, leur nombre est linéaire par rapport à la taille des textes. Ceci a pour effet de supprimer la combinatoire qui avait surgi avec notre souhait de traiter automatiquement les textes.

Cet algorithme repose fondamentalement sur l'utilisation du concept de voisinage. Du fait de sa faible spécification, l'opérationnalisation de ce concept entraîne une explosion combinatoire. Néanmoins, notre algorithme est capable de faire face à cette combinatoire en s'orientant dans l'espace des voisinages au moyen du principe d'optimisation que nous avons fixé. De manière imagée, l'algorithme sélectionne la focale d'observation adéquate du texte, soit le voisinage « le plus intéressant » auquel associer chaque caractère. En outre, cela correspond à un point de vue topologique sur la méthode de segmentation présentée dans la section 3.

On remarquera que cet algorithme formalise un procédé effectué de façon manuelle par le linguiste. En effet, lors de l'étude d'un texte ou d'un corpus, c'est le linguiste qui restreint manuellement, grâce à sa connaissance et sa pratique de la langue, le voisinage d'une unité linguistique autour duquel rechercher un ensemble de corrélations. Dans notre cas, on recherche automatiquement un ensemble de voisinages « intéressants » relativement au critère fixé, soit l'étude des segments répétés.

4.2 Linéarités

Dans la section précédente, nous avons constaté que c'est la linéarité, la séquentialité du texte qui fondait l'utilisation des outils de l'algorithmique textuelle que nous avons mis en œuvre. Nous avons présenté les fondements algorithmiques de notre approche et en quoi ceux-ci s'inscrivaient dans un cadre topologique d'un point de vue local. Examinons à présent en quoi notre approche s'inscrit toujours dans ce cadre topologique, mais d'un point de vue global cette fois.

Des segments répétés à l'alignement

A partir d'une paire de textes en entrée, MÉDITE fait émerger une segmentation de ceux-ci suivant les segments répétés. Examinons à présent en quoi ceci a pour effet de quasiment produire un alignement entre ces deux textes.

Nous avons déjà mentionné le fait que les segments répétés pouvaient être soit des segments invariants entre les deux textes, soit des segments déplacés d'une position dans le premier texte vers une position différente dans le second. Il convient donc de décider, pour chaque segment répété, de son type. Nous utilisons pour cela une heuristique simple en cherchant à minimiser la taille totale des segments déplacés. Les segments invariants sont des segments colinéaires (c'est-à-dire dans le même ordre dans les deux textes), alors que les segments déplacés ne le sont pas (puisque par définition leur position dans les deux textes varie) et croisent les invariants. Un algorithme de recherche d'une plus longue sous-séquence croissante [JV92] permet de décider du type de chaque segment répété. En effet, en appliquant cet algorithme à notre séquence de segments répétés, on obtient la séquence des segments invariants. Par complémentarité, les autres segments sont typés comme segments déplacés.

Une fois que les segments invariants et déplacés ont été définis, les autres types de segments sont déduits de façon triviale. Les suppressions sont des segments non répétés présents uniquement dans le premier texte. De même, les insertions sont des segments non répétés présents uniquement dans le second texte. Si une suppression et une insertion sont situées entre deux mêmes paires de segments invariants, alors on peut les apparier sous la forme d'un remplacement. Finalement, on obtient bien tous les éléments d'un alignement avec déplacements.

Linéarité en diachronie et réticularité

Observons à présent qu'un alignement entre deux textes n'est rien d'autre qu'un réseau, contraint et simple, mais néanmoins un réseau de correspondances entre les deux textes. Les invariants et les déplacements sont des liaisons entre des paires de segments répétés. Les remplacements sont des liaisons entre segments différents. Les segments supprimés et insérés sont liés à un élément vide. Ce dernier permet de spécifier dans l'autre texte la présence d'une altération diachronique en vis-à-vis. Ainsi, notre approche utilisant une algorithmique basée uniquement sur la topologie locale des formes graphiques se révèle produire une structuration réticulaire du corpus en entrée.

Une fois cet alignement obtenu automatiquement, l'informaticien cède la place au linguiste et au travail interprétatif. En effet, le linguiste ou le généticien du texte dispose alors d'un objet, l'alignement, lui permettant d'étudier l'évolution diachronique de la linéarité du texte à travers ses différents états.

Les insertions ne sont rien d'autre que des ruptures de linéarité entre deux segments du texte auparavant joints. Les suppressions, au contraire, joignent deux segments qui étaient disjoints. A l'inverse, remplacer un segment de texte par un autre ne modifie pas la linéarité du texte, bien que les syntagmes soient modifiés. Encore une fois, nous sommes ici au cœur de l'étude des voisinages qui intéresse la topologie textuelle. Par contre, les voisinages évoqués le sont cette fois suivant un point de vue global. C'est-à-dire que les deux textes ont déjà été alignés et que les voisinages ne sont plus au même niveau que dans les sections précédentes. On a cette fois affaire à l'étude des voisinages entre segments (répétés ou non) en tant que constituants élémentaires de l'alignement ; le niveau graphique a laissé place au niveau des segments. On notera que ce niveau des segments reste relativement imprécis puisqu'il englobe plusieurs niveaux linguistiques. En effet, les segments constituants

l'alignement peuvent être un suffixe d'un mot, un ou plusieurs syntagmes, ou encore plusieurs paragraphes.

Le généticien peut alors étudier dans quelle mesure ces modifications de la linéarité du texte font sens. Par exemple, la modification de tel suffixe corrigera une faute d'orthographe. Le remplacement d'une virgule et d'une minuscule par un point et une majuscule scindera une phrase en deux ; la répétition de tels phénomènes pointera une campagne de réécriture stylistique axée sur la simplification des phrases. On pourra se référer, à [BGF07] ou [Mah07] pour des exemples d'études génétiques assistées par MÉDITE.

Le cas des déplacements est plus complexe. Il participe d'un double mouvement, de rupture de linéarité d'une part, et de création de liaison réticulaire d'autre part. Déplacer un segment de texte consiste à supprimer ce segment dans le premier état, puis à l'insérer dans le second. On a donc bien une double rupture de linéarité. Mais un déplacement est plus qu'une suppression suivie d'une insertion, plus que la somme des parties ; d'où son identification en tant qu'opération particulière par la génétique textuelle. En effet, en appariant une suppression et une insertion en un déplacement, on crée ainsi une liaison réticulaire non seulement entre deux états du texte mais également entre deux passages distants du texte. Par la même occasion on passe du local, les ruptures de linéarité, au global, la création des liens réticulaires.

Il est très intéressant de remarquer qu'habituellement de telles liaisons sont obtenues en traitant des corpus au moyen d'un concordancier ou de méthodes tabulaires. On applique ensuite des méthodes de statistique lexicale, de calcul de distance intertextuelle ou d'Analyse Factorielle des Correspondances [LS94, BLM03]. Ces méthodes statistiques établissent un réseau de correspondances au sein du corpus en ne tenant pas compte de la séquentialité. En revanche notre méthode intrinsèquement basée sur la séquentialité du texte permet également d'établir le réseau que représente l'alignement avec déplacements.

5. Conclusion

Nous avons présenté nos travaux concernant l'alignement automatique en vue de l'aide à la critique génétique. Ceux-ci utilisent largement les concepts définis par la topologie et la topographie textuelles tels que les segments répétés, le voisinage et la linéarité. En effet, notre approche des études textuelles se veut résolument en phase avec le courant topologique de l'ADT.

En guise de perspectives, nous mentionnerons le fait que l'algorithmique textuelle possède bien d'autres outils pouvant se révéler intéressants pour l'ADT. Par exemple, la question de la recherche de motifs dans un texte est une question largement traitée et de nombreuses méthodes ont été proposées. La plupart d'entre elles sont basées sur des principes topologiques similaires à ceux exposés dans cet article. Il nous semble fort intéressant d'étudier en quoi ces outils pourraient être utiles dans le cadre de l'ADT.

Références

[Bar65] Barbut, M. (1965). « Topologie générale et algèbre de Kuratowski », *in Mathématiques et Sciences Humaines (12), pages 11-27*.

[BLM03] Barthélémy J.-P., Luong X. et Mellet S. (2003). « Prenons nos distances pour comparer des textes, les analyser et les représenter », *in Corpus, Numéro 2, La distance intertextuelle*.

[Bia00] de Biasi P.-M. (2000). « La génétique des textes ». *Nathan Université*.

[BG07] Bourdaillet J. et Ganascia J.-G. (2007). « Alignements monolingues avec déplacements », *in Actes de la 14e conférence sur le Traitement Automatique des Langues Naturelles, TALN 2007*.

[BGF07] Bourdaillet J., Ganascia J.-G. and Fenoglio I. (2007). « Machine Assisted Study of Writers' Rewriting Processes », *in Proceedings of the 4th International Workshop on Natural Language Processing and Cognitive Science (NLPCS)*.

[Bru08] Brunet É. (2008). « Fréquences et séquences. Mise en œuvre dans Hyperbase », *in Lexicometrica 7 « Topographie et topologie textuelles »*.

[CHL01] Crochemore M., Hancart C. et Lecroq T. (2001). « Algorithmique du texte », *Vuibert*.

[GB06] Ganascia J.-G et Bourdaillet J. (2006). « Alignements unilingues avec MEDITE », *in Actes des 8èmes Journées d'Analyse des Données Textuelles, JADT 2006*.

[Gus97] Gusfield D. (1997). « Algorithms on Strings, Trees and Sequences: Computer Science and Computer Biology », *Cambridge University Press*.

[HNS97] Habert B., Nazarenko A. et Salem A. (1997). « Les linguistiques de corpus », *Paris, Armand Colin.*

[JV92] Jacobson G. et Vo K.-P. (1992). « Heaviest Increasing/Common Subsequence Problems », *in Proceedings of the Tthird Annual Symposium on Combinatorial Pattern Matching, CPM 1992.*

[LS83] Lafon P., Salem A. (1983). « L'inventaire des segments répétés d'un texte », *in Mots 6, pages 161-177*.

[LS94] Lebart L., Salem A. (1994). « Statistique textuelle », *Paris, Dunod.*

[Mah07] Mahrer R. (2007). « La Génétique Assistée par Ordinateur : MEDITE au banc d'essai ou Du tout neuf pour le Tout-Vieux », *in Genesis 27, pages 168-172*.

[MB08] Mellet S. et Barthélemy J.-P. (2008). « La topologie textuelle : légitimation d'une notion émergente », *in Lexicometrica 7 « Topographie et topologie textuelles »*.

[Ukk95] Ukkonen E. (1995). « On-Line Construction of Suffix Trees », *in Algorithmica 14(3), pages 249-260.*

[Vér00] Véronis J. (éditeur). « Parallel Text Processing, Alignment and Use of Translation Corpora ». *Kluwer Academic Publisher*.