Jean-Gabriel Ganascia

EDITE – MEDITE: un passage des versions aux variantes?

Des versions aux variantes

Projet financé par le programme société de l'information du CNRS, EDITE – Etude Diachronique et Interprétative de Textes d'Ecrivains (Ganascia/Fenoglio/Lebrave 2004) –, fait collaborer un laboratoire d'informatique, le LIP6 – Laboratoire d'Informatique de Paris 6 – et un laboratoire de lettres, l'ITEM – Institut des Textes et Manuscrits Modernes. Le nom indique à lui seul les objectifs poursuivis: comparer automatiquement les versions successives des textes pour identifier et caractériser les étapes du processus d'écriture et de réécriture d'œuvres littéraires, musicales, philosophiques ou scientifiques. MEDITE – Machine EDITE – est le nom du logiciel réalisé dans le cadre du projet EDITE pour faciliter et automatiser la comparaison de différents états d'un texte. Il prend en entrée les différentes *versions*, c'est-à-dire différents supports physiques – brouillons d'auteurs ou éditions successives – dont on suppose la chronologie connue. Ces versions donnent souvent naissance à plusieurs états correspondant aux différentes étapes de la correction: mise au net, rature, repentir, ajout etc. Et, l'on transforme chacune de ces versions de façon à obtenir autant de textes linéaires qu'il existe d'états. La machine EDITE opère sur ces textes; elle détermine automatiquement les transformations qui font passer de l'un à l'autre en faisant appel aux opérateurs identifiés par la génétique textuelle – insertion, destruction, remplacement et déplacement. Autrement dit, MEDITE systématise et automatise le travail préalable requis par la critique génétique.

Appliquée à l'étude des versions successives de textes littéraires, la Machine EDITE – MEDITE – donne d'excellents résultats: elle souligne l'existence de phases de réécriture; et, sur certaines, elle met en évidence des «cascades» de transformations relevant toutes d'une même intention ou d'un même champ sémantique. Peut-on employer ces techniques pour étudier les variantes rédigées par des copistes imaginatifs? A supposer qu'il y ait une cohérence dans l'ensemble des transformations qui font passer d'un premier texte à un second, la recension exhaustive de ces transformations et leur juxtaposition devrait y aider. Dans cette éventualité, nous passerions d'une étude des versions à une étude des variantes et nous mettrions les outils développés dans le cadre d'une étude des versions successives de textes d'auteurs modernes au service d'un examen des variantes d'un même texte, du moins c'est ce que l'on médite dans le projet EDITE.

Après avoir rappelé quels sont les principes de la critique textuelle génétique et ce que sont les versions d'un texte littéraire, nous présentons le logiciel MEDITE et nous montrons sur un exemple précis comment il s'applique à l'étude des versions. Nous examinons ensuite les difficultés soulevées par l'utilisation de ce logiciel pour l'étude des variantes de textes médiévaux. Nous décrivons enfin les spécifications de la nouvelle version du logiciel MEDITE consacrée très spécifiquement à l'étude des variantes.

Etude des versions de textes littéraires

Critique textuelle génétique épaulée par ordinateur

La génétique textuelle (de Biasi, 2000; Hay, 2002; Gresillon, 1994; Lebrave 1984, 2000 etc.) étudie les processus d'écriture des textes à partir de traces qui reflètent le travail d'ajustage auquel se livre l'écrivain dans ses repentirs, ses coupes ou ses insertions successifs. Cette discipline a pris place dans le champ de la critique littéraire au cours des années 70. Très exactement c'est en 1968 que l'étude des manuscrits a été reconnue sous la forme de la création d'une équipe CNRS (fondée et dirigée par Louis Hay) chargée de recenser et classer les manuscrits de Heine acquis deux ans plus tôt par la Bibliothèque nationale. L'équipe s'étoffe peu à peu d'autres fonds et après s'être appelée Centre d'histoire et d'Analyse des Manuscrits Modernes (CAM), elle devient, en 1982, un laboratoire: l'Institut des Textes et Manuscrits Modernes (ITEM).

Généralement, du moins pour la plupart des auteurs d'avant l'âge informatique, des brouillons rassemblent les traces d'écriture sous forme soit intégralement manuscrite, soit partiellement manuscrite et partiellement tapée à la machine, soit totalement tapée à la machine. Et, ces brouillons font ici l'objet d'une étude systématique à l'aide d'outils informatiques qui s'inspirent partiellement de ceux développés pour l'étude des macromolécules biologiques. Nous avons réalisé un programme qui repère automatiquement les opérations structurales qui font passer d'un texte à un autre. Ces transformations élémentaires (déplacements, insertions, suppressions et remplacements de blocs de caractères), identifiées depuis longtemps par les spécialistes de la génétique textuelle, peuvent ensuite être associées aux catégories syntaxiques ou sémantiques des mots ou des groupes de mots pour donner naissance à des opérations linguistiques de réécriture (déplacement d'un adverbe, remplacement d'un mot par un hyperonyme ou par un hyponyme, suppression ou ajout d'un adjectif etc.) (Fenoglio and Boucheron, 2002). Notre logiciel mime les opérations exécutées à la main par le généticien qui compare des textes. Il comprend une interface permettant aussi bien de visualiser les modifications faisant passer d'un état du texte à un autre, que de recenser toutes les modifications, ajouts, suppressions, déplacements ou remplacements. L'automatisation autorise à la fois une répétition à l'identique de ces opérations, et une systématisation de la démarche sur des textes longs qu'il eut été très pénible de traiter manuellement. On peut ainsi travailler sur des articles, voire sur des livres entiers, et procéder à des études statistiques afin de caractériser le style de réécriture de tel ou tel auteur, et d'identifier, pour un même auteur, les différentes phases de réécriture: expansion, resserrement, …

Originellement, le projet fut conçu pour la critique génétique: il s'agissait d'aider à comparer des brouillons d'auteurs, afin de saisir la nature du travail de réécriture. Toutefois, d'autres applications sont envisagées. Ainsi, nous l'appliquons actuellement à des articles de journaux gratuits, en vue de les confronter aux dépêches d'agences de presse dont leurs rédacteurs se sont inspirés et d'apprécier la nature exacte du travail journalistique. De même, en nous référant à l'«Eloge de la variante» de Bernard Cerquiglini (Cerquiglini, 1989) nous souhaiterions comparer des variantes de textes de la littérature médiévale pour saisir le sens des écarts. Dans tous les cas, qu'il s'agisse de versions ou de

variantes, il convient d'abord de préciser ce dont on part et ce sur quoi porte exactement la comparaison (chaînes de caractères, mots, syllabes, syntagmes …); c'est la raison pour laquelle nous commençons ici par rappeler exactement ce que sont les versions et les états de textes littéraires.

Versions et états de textes littéraires

Versions

Dans la suite, nous distinguerons les différents supports matériels, c'est-à-dire les brouillons successifs, comme autant de *versions* de l'œuvre. Ainsi, dans le dossier génétique d'une nouvelle d'Andrée Chedid «La robe noire» paru dans le recueil *Les saisons de passage* (1996), l'auteur a recopié cinq fois son texte, donnant naissance à cinq versions. Nous nous référons ici à la première de ces versions qui se présente comme un texte tapé à la machine, puis raturé, annoté et surligné à l'aide de différents feutres de couleurs

États

Chaque brouillon est marginé, biffé, réécrit, ce qui rend sa lecture assez confuse. Avec l'expérience, les spécialistes de génétique textuelle savent identifier, avec une plus ou moins grande certitude, les différents états du texte, c'est-à-dire les différents textes présents sur une même version. Chacun de ces états correspond à une transcription linéaire qui fait abstraction de l'information visuelle et de la spatialisation: inscriptions marginales, couleurs, notes, etc. tout y est réduit à du texte brut. Nous partons d'une des versions, c'est-à-dire d'un des manuscrits d'Andrée CHEDID. Sur cette version, l'état premier est identifié au texte tapé à la machine et les différentes couleurs du manuscrit sont associées aux différentes campagnes de réécriture et de relecture. Pour faciliter la présentation des choses, nous ne considérerons ici que deux états: le tapuscrit et l'état final (voir figure 1). Bien évidemment, ce n'est là qu'un artifice de présentation. Il appartient dans chaque cas au chercheur de définir les différents états qu'il veut considérer.

Une fois ces états identifiés, le logiciel MEDITE va les comparer de façon à retrouver automatiquement les opérations de réécriture, pour en faire l'inventaire. Ceci étant, il convient de bien noter que la mise en œuvre du logiciel MEDITE présuppose qu'un travail préalable ait dégagé, à partir des différentes versions, les différents états du texte sous forme d'autant de transcriptions linéaires de ce même texte.

Etat initial: tapuscrit	Etat final
Elle nous aimait. Nous nous sentions aimés, quels que soient les moments nos existences et la distance. Parfois avec abnégation, reculant dans l'ombre pour nous laisser toute la place, gommant sa flamboyance pour nous offrir l'avant-scène. Une avant-scène qu'elle occupait avec éclat et assurance tandis que j'évitais, par nature et par goût, de m'y avancer. C'était un week-end d'été, sur une plage du nord de la France. Ni famille ni gouvernante, nous nous étions rejointes en je ne sais plus quelle circonstance. J'avais 16 ans. Un jeune Anglais aussi laid que Woody Allen – mais qui n'en possédait, je crois, ni le charme, ni la vivacité – me faisait une cour discrète. Dotée de parents d'un physique exceptionnel, j'avais le culte de la beauté et ce petit homme malingre, à lunettes, au visage de fouine n'avait rien pour m'attirer. Il habitait Londres, et paraissait prospère. Ma mère pensait sans doute qu'un mariage lointain résoudrait mes problèmes et les siens que poseraient bientôt mon retour en Egypte. - Tu lui plais beaucoup, me disait-elle. - Il ne me plaît pas du tout! Il n'en fut jamais plus question.	Nous nous sentions aimés par elle; quels que soient nos âges ou la distance. Elle nous aimait avec intelligence et sans abnégation demeurant elle-même dépourvue de cette amertume qui s'infiltre parfois dans les âmes trop dévouées qui brûlent leur existence au feu des autres. Il lui arrivait pourtant de s'effacer pour un temps, de reculer, d'abandonner l'avant-scène qu'elle occupait avec éclat et assurance pour céder place à quelqu'un de son choix. Elle jouait alors pour un temps, les seconds rôles, exaltait – exagérément parfois les qualités de celle ou de celui qu'elle décidait de placer sous les feux de la rampe. Un week-end d'été au Touquet, nous nous étions rejointes sans famille, ni gouvernante, je ne sais plus en quelle circonstance. J'avais 16 ans. Un jeune Anglais d'une trentaine d'années aussi laid que Woody Allen – mais qui n'en possédait ni le charme, ni la vivacité – me faisait une cour discrète. Malingre, à lunettes, avec son visage de fouine il n'avait rien pour me plaire. Beaucoup trop âgé à mon goût et la beauté de mes parents me rendait exigeante. Je cherchais à le fuir. Ma mère devait penser qu'un mariage en pays étranger résoudrait mes problèmes et les siens, ceux que poseraient bientôt mon retour en Egypte. Elle lui avait aussi sans doute découvert des qualités cachées. - Tu lui plais beaucoup, me dit-elle. - Il ne me plaît pas du tout! Mais du tout. Elle éclata de rire et il n'en fut jamais plus question.

Figure 1: état initial et état final d'une version de «La robe noire» d'Andrée Chedid

Le logiciel MEDITE

Fondements algorithmiques de MEDITE

Comme nous venons de le voir, le programme MEDITE prend en entrée deux états d'un même texte de façon à repérer les transformations qui font passer de l'un à l'autre, ou, plus exactement, l'ensemble minimal de transformations qui fait passer du texte initial au texte «corrigée». Formulé de la sorte, le problème apparaît très proche de celui posé par le calcul des «distances d'édition» (Sankoff/Kruskal, 1983; Crochemore/Rytter, 1994, Ganascia, 2001). Rappelons que la notion de «distance d'édition» se fonde sur des opérateurs de transformation, que l'on appelle en termes techniques des «éditions», car ils modifient des

chaînes de caractères, et sur la minimisation du coût des transformations qui font passer d'une séquence à une autre.

Dans un premier temps, nous avions cru pouvoir réutiliser les distances d'édition, d'où l'acronyme du projet, EDITE, qui fait référence aux dites «éditions» et qui signifie accessoirement «Etude Diachronique et Interprétative du Travail de l'Ecrivain». Or, il s'est rapidement avéré que cette utilisation des distances d'édition n'était pas possible, du moins telle quelle. En effet, il n'existe de procédure efficace de calcul de la distance d'édition que pour des ensembles d'éditions très restreints, comme l'ensemble dit standard qui comprend les trois opérations de *suppression*, d'*insertion* et de *remplacement*. Dans la mesure où la détection des *déplacements* joue un rôle important pour la génétique textuelle, et que l'introduction des *déplacements* dans l'ensemble des éditions change totalement la complexité algorithmique de la procédure de calcul des distances, il est nécessaire de procéder autrement. De plus, la taille des textes (plusieurs centaines de milliers de caractères) interdit l'emploi de procédures d'une complexité polynomiale: il faut se limiter à une complexité linéaire ou, au plus, à une complexité en O(n*lg(n)), n étant la longueur des textes à traiter.

Afin de réduire la complexité et de répondre au mieux au problème posé, nous avons donc conçu un algorithme spécifique qui procède en trois étapes:

1. Détection des blocs communs maximaux disjoints
2. Identification des déplacements et des pivots
3. Calcul des insertions, des suppressions et des remplacements

Détection des blocs communs maximaux disjoints

La détection des blocs maximaux fait appel à des algorithmes classiques (Karp/Miller/ Rosenberg, 1972) de recherche d'homologies dans les séquences. Nous n'insisterons donc pas sur la mise en œuvre de ces algorithmes, sauf à dire qu'il y a parfois des recouvrements entre blocs maximaux. Ainsi, les deux chaînes «Il a avalé» et «Il avala» donnent deux blocs maximaux, |Il a | et | aval| qui se recouvrent partiellement. Pour obtenir des blocs maximaux disjoints, il faut introduire une césure. Or, généralement il y a plusieurs possibilités. Sur notre exemple, il y en a trois: |Il↑ a | et |↑ aval|, |Il ↑a | et | ↑aval| ou |Il a↑ | et | a↑val|, ce qui donne, en soulignant les insertions et les suppressions sur les deux chaînes initiales, les trois solutions suivantes: «Il| a| aval|é» et «Il| aval|a», «Il |a |aval|é» et «Il |aval|a» ou «Il a| a|val|é» et «Il a|val|a». Dans la mesure du possible, il faut éviter la fragmentation des mots, c'est pourquoi nous avons choisi de mettre en priorité la césure sur les signes de ponctuation ou sur les blancs.

Par ailleurs, toujours pour éviter la fragmentation excessive des mots, nous ne mentionnons que les blocs communs d'une taille supérieure à une valeur seuil fixée arbitrairement. Par défaut ce seuil est de 4, ce qui veut dire qu'avant introduction de la césure, les homologies doivent avoir une longueur supérieure à 4 caractères. De la sorte, nous repérons des mots isolés de longueur supérieure à deux caractères, sachant qu'ils sont

entourés de deux frontières de mots (blanc ou signe de ponctuation), ainsi que les préfixes ou les suffixes de plus de trois caractères, ce qui correspond à une syllabe.

Identification des déplacements et des pivots

Parmi l'ensemble des blocs communs maximaux disjoints, certains se retrouvent dans le même ordre dans les deux textes, le texte source et le texte corrigé, tandis que d'autres apparaissent déplacés. Ainsi, si nous avons la séquence de blocs maximaux B_1 B_2 B_3 B_4 B_5 dans le texte source et la séquence B_2 B_3 B_1 B_4 B_5 dans le texte corrigé, on peut inférer que le bloc B_1 a vraisemblablement été déplacé, même si cette appréciation est subjective, car on pourrait tout autant dire que ce sont les blocs B_2 et B_3 qui ont été déplacés. L'algorithme que nous avons mis en œuvre détermine les blocs déplacés en essayant de minimiser l'amplitude des déplacements mesurée en nombre de caractères. Plus exactement, cet algorithme prend en considération la taille des blocs maximaux de façon à minimiser le nombre de déplacements de caractères requis pour passer d'une séquence de blocs maximaux à l'autre.

A l'issue de cette phase, on distingue parmi les blocs maximaux disjoints, des blocs dits «déplacés» et des blocs qui apparaissent dans le même ordre dans le texte source et dans le texte cible. Ces derniers sont appelés les «blocs pivots» de la comparaison.

Calcul des insertions, des suppressions et des remplacements

Une fois déterminés les «blocs pivots» et les «blocs déplacés», il reste à calculer les *suppressions*, les *insertions* et les *remplacements*. Le programme procède comme suit:

- Lorsque deux pivots P et P' sont jointifs dans le texte source, la chaîne qui sépare P et P' dans le texte corrigé correspond à une *insertion*. Notons, pour éviter tout malentendu, que les deux pivots P et P' ne peuvent être jointifs à la fois dans le texte source et dans le texte corrigé, car sinon, P et P' ne serait maximaux ni l'un, ni l'autre.
- Lorsque deux pivots P et P' sont jointifs dans le texte corrigé, la chaîne qui sépare P et P' dans le texte source correspond à une *suppression*.
- Enfin, lorsque deux pivots P et P' ne sont jointifs ni dans le texte source, ni dans le texte corrigé, on dit qu'il y a *remplacement* de la chaîne comprise entre P et P' dans le texte source, par la chaîne comprise entre P et P' dans le texte corrigé.

A ce stade, il convient de préciser qu'un même bloc peut être à la fois déplacé et se trouver dans une insertion, une suppression ou un remplacement. En effet, dans le cas de déplacements de petits blocs situés à l'intérieur d'une insertion, d'une suppression ou d'un remplacement, il est souvent préférable de considérer que c'est l'ensemble qui est inséré, supprimé ou remplacé, de façon à éviter une fragmentation excessive des textes. Nous avons introduit deux paramètres qui permettent de lisser plus ou moins les résultats et d'inclure les blocs déplacés dans les insertions, les suppressions ou les remplacements.

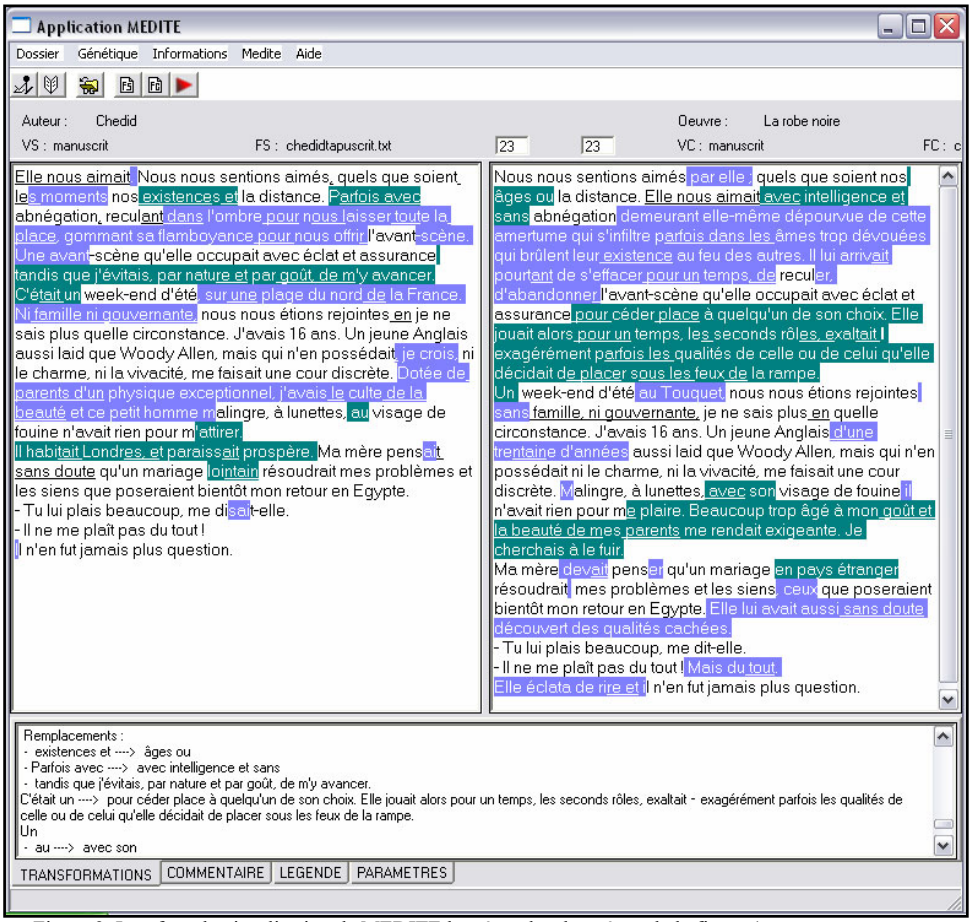

Figure 2. Interface de visualisation de MEDITE lancé sur les deux états de la figure 1

Dans tous les cas, l'information sur le déplacement n'est pas omise. Elle vient se surajouter à d'autres informations. C'est particulièrement important pour repérer qu'un mot ou qu'un radical est «libéré» par un auteur afin d'être réemployé plus loin dans le même texte, sans commettre de répétition.

Notre algorithme a été testé sur de nombreux textes d'écrivains. En confrontant les résultats obtenus avec des interprétations génétiques, on constate que la plupart du temps, il retrouve les déplacements, les insertions, les suppressions et les remplacements déjà identifiés manuellement par les généticiens du texte.

Exemple de fonctionnement de MEDITE

Pour faire fonctionner MEDITE, il faut charger d'abord le texte source dans la fenêtre de gauche et le texte corrigé dans la fenêtre de droite, ce qui se fait, dans l'un et l'autre cas, comme dans un éditeur classique, avec des menus déroulant ou des boutons placés sur le cadre.

Une fois lancée, la comparaison recourt à l'algorithme précédemment décrit. Les résultats s'affichent en couleur: insertions, suppressions et remplacements sont marqués chacun par une couleur spécifique, que l'on peut faire varier à loisir. De plus, les blocs déplacés sont soulignés, ce qui autorise une superposition des deux indications: déplacements d'un côté, insertions, suppressions ou remplacements de l'autre (voir figure 2).

Enfin, comme sur de très longs textes le lecteur est susceptible de se perdre, un compteur indique au-dessus de chacune des deux fenêtres de la partie supérieure, le numéro d'ordre du premier pivot présent dans la fenêtre de visualisation. L'utilisateur a alors tout loisir de faire défiler les textes à l'aide des ascenseurs, pour mettre les pivots de l'un et de l'autre texte en regard. Pour faciliter encore les choses, en cliquant sur un pivot dans l'une ou l'autre fenêtre, le texte homologue défile automatiquement dans l'autre fenêtre, jusqu'à ce que le pivot correspondant soit mis en regard du premier.

Comparaison entre «comparaisons de versions»

MEDITE n'est pas le premier logiciel de comparaison de versions, loin de là. Ainsi, les traitements de textes usuels proposent des outils de comparaison de versions très commodes lorsque l'on souhaite travailler à plusieurs sur un même document. Cependant, ces comparaisons sont inutilisables pour un philologue qui souhaite analyser finement les variations d'un état à un autre et les transcrire sous forme de séquences d'opérations matérielles sur le texte.

Pour se convaincre de l'insuffisance des techniques de comparaison de versions implantées dans le logiciel *Word*, reprenons les deux états donnés dans la figure 1 et reportons nous d'abord à la figure 2, où se trouve effectuée la comparaison entre ces états avec MEDITE, puis à la figure 3, où la même comparaison est faite avec *Word*. Cette dernière illustre les insuffisances dont souffrent les logiciels usuels de comparaison de versions, insuffisances qui interdisent de les employer pour la critique génétique; elles sont au nombre de quatre:

1. Ils font un certain nombre d'erreurs. Par exemple, dans le cas qui nous intéresse, la première phrase «Elle nous aimait» a été supprimée de l'état initial au cours de la réécriture, or elle est marquée comme étant insérée. Une étude attentive montrerait qu'un nombre important d'erreurs de ce type est commis par ces logiciels.
2. Nous souhaitons prendre en considération tous les opérateurs de transformation identifiés par les généticiens du texte, à savoir l'insertion, la suppression, le remplacement et les déplacements. Et, Word ne considère que l'insertion et la suppression.
3. L'interface de visualisation est vraiment très mal commode; il est même quasiment impossible de localiser avec précision les suppressions.
4. Enfin, on aimerait récupérer l'ensemble de ces transformations pour faire des statistiques et étudier de façon systématique les évolutions sémantiques ou syntaxiques d'une version à l'autre. Or, avec la comparaison des versions de Word, c'est impossible.

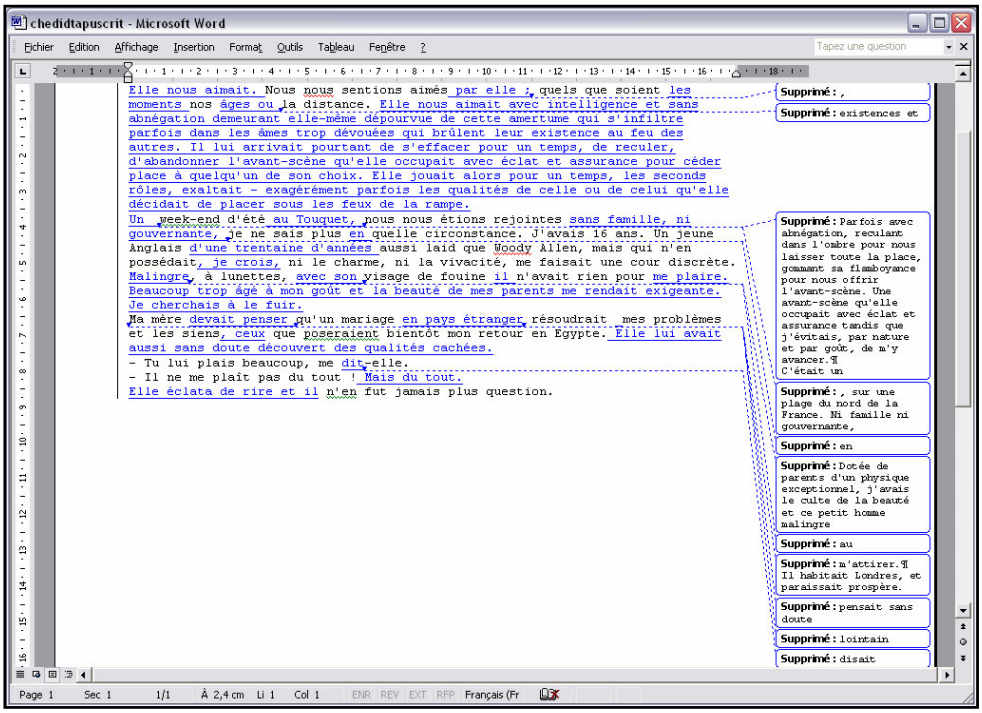

Figure 3. comparaison des deux états du manuscrit d'André Chedid avec Word

Notons que s'il existe des logiciels de comparaison utilisés par les philologues comme TUSTEP (cf. *http://www.uni-tuebingen.de/zdv/tustep/tustep_eng.html*), toutefois, ceux-ci souffrent de limitations semblables à celles que nous avons indiquées: absence de la notion de déplacement, interface mal pratique etc.

Interprétation génétique

Reportons-nous de nouveau à la copie d'écran donnée dans la figure 2. Elle correspond à la comparaison des deux états initiaux et finaux de la version 1 du manuscrit d'Andrée Chedid (voir figure 1). MEDITE permet une visualisation immédiate de tout ce qui se passe entre les états initiaux et finaux de la version 1 du texte. L'exhaustivité des transformations nous est matériellement donnée dans le tableau inférieur. Voilà, immédiatement offert et classé, le matériau minimum nécessaire au généticien du texte.

Ainsi, pour donner une illustration sur ce passage, nous remarquons que les transformations – dont les éléments qui les constituent sont de différentes natures linguistiques – convergent quasiment toutes vers le personnage de la mère «Elle».

Plus exactement, dans ce passage de l'état initial (EI) à l'état final (EF) s'opère une vraie conversion de point de vue; la place de la mère change. Si EI s'efforce de mettre en valeur une générosité maternelle, la suite des suppressions et remplacements montre clairement que EF installe une vision toute différente où la mère s'impose et prend toute la place: le bloc «l'avant-scène qu'elle occupait avec éclat et assurance» reste inchangé alors

que les segments contextuels, à droite comme à gauche, passent d'une minimisation à une affirmation amplifiante de cette place («elle» → «par elle»; «Parfois avec abnégation» → «avec intelligence et sans abnégation»; «reculant dans l'ombre pour nous laisser toute la place» → «il lui arrivait pourtant de s'effacer pour un temps»; insertion de «céder la place à quelqu'un de son choix» etc.)

Application à l'étude des variantes

Jusqu'à présent, nous avons évoqué la comparaison de versions de textes littéraires. Rappelons que, dans cette éventualité, toutes les transformations qu'un auteur imprime à son texte sont signifiantes. Ainsi, les changements de casse, l'insertion de signes de ponctuation, la mise en page, tout doit être mis en évidence. Qui plus s'est, les modifications internes des mots doivent être soulignées. Ainsi, les corrections orthographiques, par exemple la mise au pluriel ou le changement de temps d'un verbe, sont significatives. Enfin, la réutilisation d'un même radical, supprimé à un endroit du texte puis réinséré plus loin dans le même texte, doit aussi être détectée. Toutes ces considérations nous ont conduit à assimiler les textes à des chaînes de caractères que l'on compare, puis à étudier les insertions, les suppressions, les remplacements et les déplacements de sous chaînes à l'intérieur de ces textes.

Lorsque l'on aborde la comparaison de variantes, qu'il s'agisse de variantes de textes littéraires ou de textes plus ordinaires, comme par exemple d'articles de journaux ou de tracts politiques, les questions se posent d'une façon un peu différente. D'un côté, certains changements sont peu, voire pas du tout significatifs; ainsi, la mise en majuscules de certaines lettres, l'introduction de signes de ponctuation, de signes diacritiques ou les changements de mise en page importent peu. D'un autre côté, l'orthographe des mots varie parfois sans que cela change fondamentalement la nature du texte. Un logiciel d'alignement de texte doit être capable de faire fi de ces différences pour mettre en correspondance deux variantes de textes écrites à deux époques distinctes par deux scripteurs qui n'obéissaient pas aux mêmes règles typographiques et orthographiques. En résumé, il faut aligner des textes qui présentent des différences notoires, en faisant abstraction de ces différences.

En d'autres termes, là où la génétique textuelle recommande une attention soutenue à la lettre des textes et à leur mise en page, la comparaison de variantes demande de saisir, si ce n'est l'esprit des textes, du moins leur littéralité, pour les mettre en correspondance, même s'ils présentent un certain nombre de différences orthographiques ou typographiques. À cette fin, on doit établir les équivalences admissibles entre caractères et entre mots. L'équivalence entre caractères gomme des distinctions qui n'ont pas lieu d'être, ce qui facilite l'appariement. À titre d'illustration, il se peut que la casse n'ait pas d'importance ou que la présence d'un signe diacritique doive être ignorée, ou encore que les séparations entre mots, qu'il s'agisse de signes de ponctuation ou de blancs, soient toutes considérées comme identiques. L'équivalence entre mots tient compte des différences orthographiques. Ainsi, «*guer*» peut être équivalent à «*gaire*», «*Jhesu*» à «*Jesu*», «*vierges*» à «*virgines*» etc. Nous voulons modifier MEDITE pour prendre en compte ces équivalences au cours de l'alignement des textes.

Équivalences de caractères

La première modification du programme MEDITE porte sur la prise en compte de l'équivalence de certains caractères. Lorsqu'il s'agit de la casse ou de signes diacritiques ou encore de la cédille, cela ne pose aucun problème; l'algorithme du programme MEDITE peut être réutilisé presque tel quel. Il suffit de préciser quels sont les caractères considérés comme équivalents. C'est ce qui est déjà implanté dans la version 3.2 du logiciel MEDITE où des options offrent à l'utilisateur la possibilité de s'affranchir de la casse, de signes diacritiques ou de la cédille lors de l'appariement entre textes.

Les choses se corsent lorsque l'on désire prendre en considération l'équivalence des séparateurs. Plus précisément, lorsqu'on introduit des signes de ponctuation, les chaînes de séparation entre les mots doivent toutes être considérées comme équivalentes. Ce qui pose problème n'est bien entendu pas l'équivalence entre signes de ponctuations, blancs et retours chariot, puisque nous nous retrouvons là ramené au problème précédent, mais c'est la différence de longueur entre les chaînes de séparation. Ainsi, si l'on introduit une virgule ou un point-virgule entre les mots, cela signifie qu'un blanc de séparation est remplacé par soit deux caractères, une virgule est un blanc, soit trois caractères, un blanc insécable, un point-virgule et en blanc.

Sans entrer dans les détails de l'algorithme mis en oeuvre, nous voyons là que cette transformation requiert une modification substantielle de celui-ci. Ceci étant, nous pouvons toujours peu ou prou faire appel à des algorithmes classiques de détection d'homologies sur les chaînes de caractères. Il suffit d'opérer un prétraitement des textes en remplaçant chaque chaîne de séparateurs contigus par un seul caractère séparateur générique.

Variantes orthographiques

La prise en considération des variantes orthographiques pose des problèmes beaucoup plus délicats. En effet, il faut d'une part connaître les variations possibles et d'autre part tenir compte de l'équivalence des mots qui résulte de ces variations au cours de l'appariement des textes.

Recherche d'homologies sur des chaînes de mots

Pour surmonter la seconde des difficultés, nous avons modifié l'algorithme de recherche d'homologies dans les séquences. Plus exactement, nous avons repris les algorithmes classiques de recherche d'homologies, mais au lieu de rechercher des homologies sur des chaînes de caractères nous recherchons maintenant des homologies sur des chaînes de mots. Le principe au fondement même de l'algorithme de recherche d'homologies n'est pas affecté par cette transformation. En revanche, sa mise en oeuvre informatique est quelque peu changée. Pour comprendre les raisons de ces changements, il convient de rappeler les principes sur lesquels reposent les algorithmes classiques de détection d'homologies: ceux-ci commencent par établir un alphabet à partir du texte. Ensuite, l'algorithme recherche les séquences de deux caractères quelconques de l'alphabet qui se répètent, puis à partir des

couples de chaînes récurrentes de deux caractères, l'algorithme recherche les chaînes de quatre caractères qui se répètent et il recommence de façon récursive avec les chaînes de huit caractères, de seize caractères etc.

Lorsque l'on passe des chaînes de caractères aux chaînes de mots, on peut appliquer exactement le même algorithme si ce n'est que les caractères sont remplacés par les mots. Ainsi, la première étape consiste non plus à construire un alphabet, mais un lexique. Quant aux étapes ultérieures, elles sont identiques, à ceci près que les séquences de caractères sont remplacées par des séquences de mots. Le principe algorithmique n'est pas modifié, mais les dimensions le sont: tandis que le nombre de caractères avoisine la centaine, le nombre de mots qu'il convient de considérer surpasse largement le millier, pour approcher la dizaine de milliers. En conséquence, les structures de données que l'on devra employer pour mettre en oeuvre l'algorithme seront modifiées pour tenir compte de ces nouvelles dimensions. C'est cette nouvelle mise en oeuvre de l'algorithme de détection d'homologies sur les mots que nous sommes en train de programmer.

Équivalence des mots

Etant admis que l'algorithme de détection d'homologies opère sur des chaînes de mots, le problème posé par l'équivalence des mots, qui résulte des variations orthographiques, est identique à celui que posait l'équivalence des caractères sur les chaînes de caractères.

La difficulté ici n'est plus d'ordre algorithmique, mais d'ordre linguistique: il faut être en mesure d'établir une table des variations orthographiques. Or, s'il existe des lexiques et des dictionnaires qui établissent des inventaires de variations orthographiques, il est presque impossible de faire une recension exhaustive de toutes ces variations. Nous ne pouvons donc pas supposer qu'une table des équivalences entre mots soit initialement donnée. A défaut, l'algorithme que nous proposons la construit: nous supposons que les textes adoptent une orthographe régulière, c'est-à-dire que les différentes occurrences d'un mot s'écrivent toutes plus ou moins de la même façon dans un même texte, aux différences flexionnelles près. Quant à la table des équivalences entre mots, elle est établie manuellement, par alignement des lexiques des deux textes en cours de comparaison. Plus exactement, à l'issue de la première étape de construction des lexiques, ceux-ci sont alignés à l'aide d'un certain nombre de règles générales et éventuellement en ayant recours à une intervention des utilisateurs.

Ordre des mots

Le dernier point sur lequel il convient d'insister tient à l'ordre des mots: les variantes ne se restreignent pas à des modifications typographiques ou lexicales; elles font apparaître des modifications d'ordre syntaxique qui portent sur l'ordre des mots. Il convient donc parfois de faire abstraction de déplacements locaux dans la comparaison des textes. L'algorithme mis en oeuvre dans le programme MEDITE autorise de tels appariements de textes. En effet, ce qui distingue MEDITE des autres aligneurs existants, c'est qu'il est capable de détecter les déplacements. Nous avons vu que, dans le cas de la critique textuelle

génétique, cette détection est tout à fait essentielle. Elle rend compte de stratégies de réécriture propres aux auteurs, par exemple d'évitement de la répétition. Lorsque l'on souhaite aligner des textes de la littérature médiévale, cela semble moins pertinent. Toutefois, les déplacements peuvent être mis à profit pour rendre compte de modifications syntaxiques mineures qui affectent l'ordre des mots. Ainsi, on peut distinguer les déplacements à courte distance et des déplacements à longue distance. Les premiers devraient alors être mis en regard de transformations syntaxiques possibles.

Conclusion

Nous avons présenté tout au long de cet article le logiciel MEDITE et ses possibles applications à l'étude des variantes de textes de la littérature médiévale. Ce faisant, nous avons essayé de montrer comment passer d'une étude génétique de textes littéraires épaulée par ordinateur, à l'étude des variantes aidée d'un ordinateur. Nous avons vu que le logiciel MEDITE devait subir des transformations substantielles avant de parvenir à aligner des variantes et à recenser les modifications qui font passer de l'une à l'autre. Parmi ces transformations, certaines ont déjà fait l'objet d'une mise en oeuvre informatique tandis que d'autres sont en cours de programmation. Nous devrions terminer et tester cette nouvelle variante de MEDITE et son implantation informatique d'ici quelques mois. En attendant, cet article dresse l'inventaire des changements qui font passer de la première version du logiciel MEDITE, destinée à la comparaison des versions de textes littéraires modernes, à une variante appropriée à l'étude des variantes de textes médiévaux. C'est donc à une sorte de «MEDITATION» réflexive, c'est-à-dire de simulation manuelle du logiciel MEDITE sur ses propres variantes, à laquelle nous nous sommes livrés ici.

Bibliographie

de Biasi Pierre-Marc (2000): *La génétique des textes.* Nathan Université.

Cerquiglini, Bernard (1989): *Éloge de la variante. Histoire critique de la philologie.* Paris: Seuil.

Chedid, Andrée (1996): «La robe noire». In: *Les saisons de passage.* Flammarion.

Crochemore, Maxime, Wojciech Rytter (1994): *Text Algorithms, «Approximate pattern matching»,* 237-251.

Fenoglio, Irène, Sabine Boucheron (éd.) (2002): Processus d'écriture et marques linguistiques, *Langages* 147, sept. 2002.

Ganascia, Jean-Gabriel, Irène Fenoglio, Jean-Louis Lebrave (2004): *Manuscrits, genèse et documents numérisés. EDITE: une étude informatisée du travail de l'écrivain.* Document numérique, vol. 8/4, 2004, 91-110.

Ganascia, Jean-Gabriel, (2001): Extraction of Recurrent Patterns from Stratified Ordered Trees. In: *Proceedings of ECML 2001* (European Conference on Machine Learning). De Raedt & Flach ed. Springer, 167-178.

Grésillon, Almuth (1994): *Eléments de critique génétique.* PUF.

Hay, Louis (2002): *La littérature des écrivains. Etudes de critique génétique.* Corti.

Karp, Richard, Raymond Miller, Arnold Rosenberg (1972): Rapid Identification of Repeated Patterns in Strings, Trees and Arrays. In: *Proc. 4th Annu. ACM Symp. Theory of Computing*, 125-136.

Lebrave Jean-Louis, (1984): Le traitement automatique des brouillons. In: *«Programmation et sciences humaines».* LISH, CNRS, n. 3, 90-106.

Lebrave Jean-Louis, (2000): La nouvelle philologie et l'édition électronique des dossiers génétiques. In: Budor, D., C. Perrus C. (eds.): *Le texte. Genèse, variantes, édition.* Arzana n° 5, Presses de la Sorbonne nouvelle, 121-151.

Sankoff, David, Joseph Kruskal (1983): *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison.*