EDITE MEDITE : un logiciel de comparaison de versions

Jean-Gabriel Ganascia[1], Irène Fenoglio[2], Jean-Louis Lebrave[2]

[1]LIP6 – Université Pierre et Marie Curie – 8 rue du Capitaine Scott – 75015 Paris – France
[2]ITEM – CNRS – 45 rue d'Ulm – 75005, Paris – France
Jean-Gabriel.Ganascia@lip6.fr

Abstract

MEDITE has been designed to facilitate, with a systematic quantification, the preliminary study required by any textual genetic interpretation and, more generally, to help the philologists. It is to compare two states of literary texts by pointing at the textual transformations between them. At the heart of the program, the main algorithm comprises three steps: the detection of the maximal disjoints common blocs, the identification of the pivots and shifts, and the computation of the deletions, insertions and replacements. Lastly, an interface renders the obtained results easily visible and makes the user able to take notes.

Résumé

MEDITE a été conçu pour faciliter, grâce à une quantification systématique, l'étude préalable à toute interprétation de génétique textuelle et, plus généralement, pour aider les philologues. Il s'agit de comparer deux états de textes littéraires en indiquant les transformations textuelles opérées de l'un à l'autre. Au cœur de ce programme, l'algorithme principal comprend trois phases : la détection des blocs communs maximaux disjoints, l'identification des pivots et des déplacements et le calcul des suppressions, des insertions et des remplacements. Enfin, une interface visualise les résultats obtenus et permet à l'utilisateur de prendre des notes.

Mots-clés : séquences, homologies, genèse textuelle, opérations linguistiques,

1. Introduction

Le travail d'ajustage auquel se livre l'écrivain dans ses repentirs, ses coupes ou ses insertions successifs fait 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 (de Biasi, 2000 ; Hay, 2002 ; Contat et Ferrer 1998 ; Gresillon, 1994 ; Lebrave, 1984 et 1990 ; etc.), 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 et Boucheron, 2002). Notre logiciel mime les opérations exécutées à la main par le philologue 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 caractéri-

ser le style de réécriture de tel ou tel auteur, et d'identifier, pour un même auteur, les différentes phases de réécriture : expansion, resserrement, …

De nombreuses applications sont envisagées. 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. D'autres applications sont envisagées, en particulier la comparaison de variantes pour la littérature médiévale. (Cf. « Éloge de la variante » (Cerquiglini, 1989))

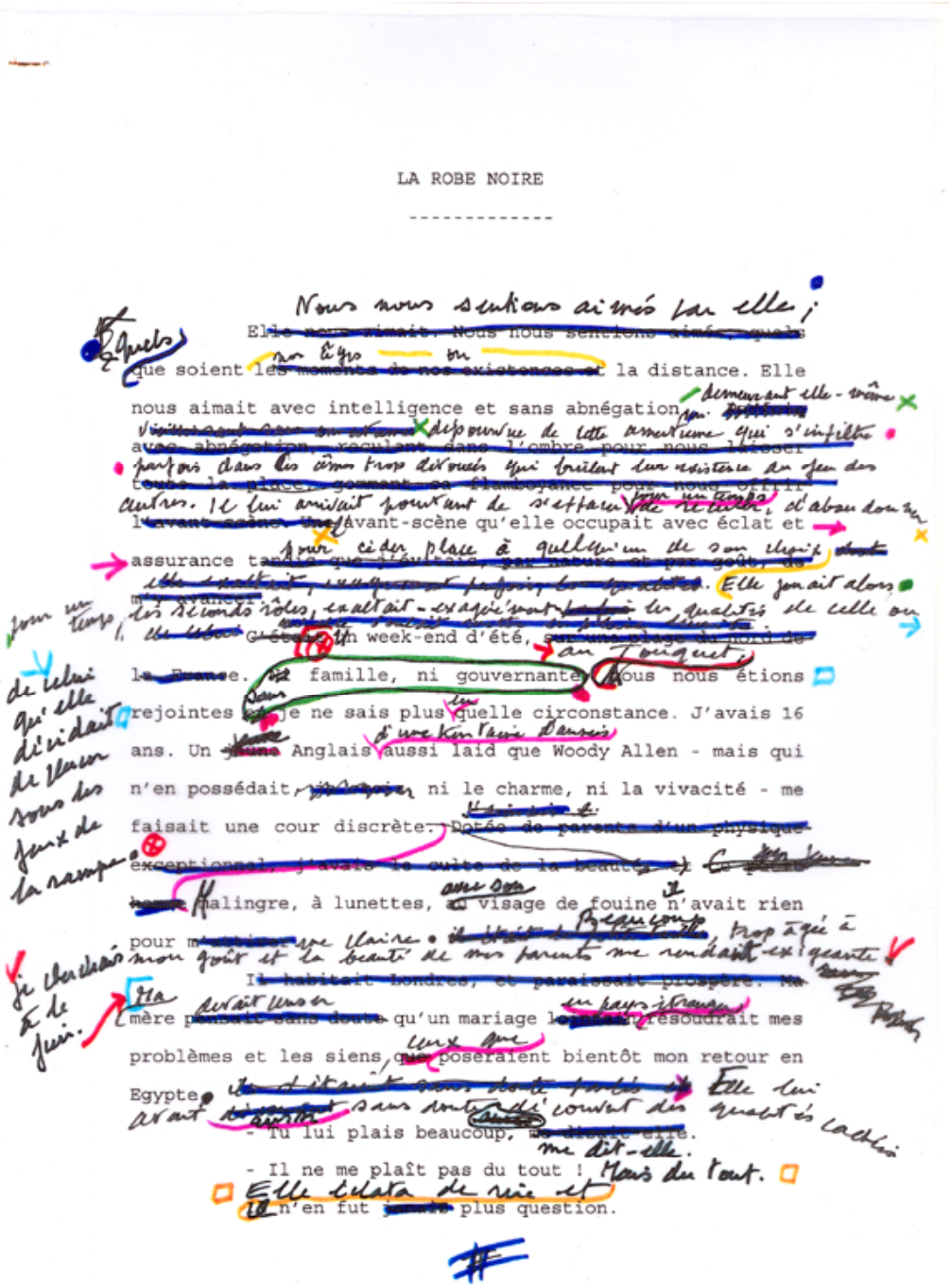

Figure 1. Une page d'un manuscrit de « La robe noire » (Andrée Chedid)

Cet article est consacré à la présentation du logiciel MEDITE, à ses fondements algorithmiques et à son interface de visualisation. Plus précisément, l'article se divise en quatre parties : après avoir précisé le sens d'un certain nombre de termes techniques employés par les généticiens du texte, nous aborderons les fondements algorithmiques du programme, puis, dans une troisième partie, nous présenterons l'interface graphique.

2. Signification de quelques termes techniques

La génétique textuelle étudie les processus d'écriture des textes à partir des traces. Généralement, du moins pour la plupart des auteurs d'avant l'âge informatique, des brouillons rassemblent ces traces sous forme soit intégralement manuscrite, soit partiellement manuscrite et partiellement tapée à la machine, soit totalement tapée à la machine.

À titre d'illustration, la figure 1 contient la photographie d'un brouillon d'auteur. En l'occurrence, le début d'une nouvelle d'Andrée Chedid, « La robe noire » paru dans le recueil Les saisons de passage (Chedid, 1996).

2.1. *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 qui nous intéresse, celui du roman d'Andrée Chedid « La robe noire », l'auteur a recopié cinq fois son texte, donnant naissance à cinq versions.

2.2. *État*

Comme on peut le constater sur la figure 1, chaque brouillon est annoté, raturé, réécrit, ce qui rend la lecture assez confuse. Toutefois, les chercheurs 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. Sur notre exemple, 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 2).

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.

État initial : tapuscrit	État 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	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

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.	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 2. État initial et état final du texte d'Andrée Chedid dans le manuscrit de la figure 1

3. Fondements algorithmiques

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 font 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 et Kruskal, 1983 ; Crochemore et 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 « Étude Diachronique et Interprétative du Travail de l'Écrivain ». 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.

3.1. *Détection des blocs communs maximaux disjoints*

La détection des blocs maximaux fait appel à des algorithmes classiques (Karp, Miller et Rosenberg, 1972 ; Landraud, Avril et Chrétienne, 1989) 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.

Notons que la longueur minimale des blocs est un paramètre qui peut être modifié par l'utilisateur, sans difficulté (Voir section 4.4.3.). Cependant, du fait de l'introduction d'une césure qui supprime les recouvrements, il se peut que des blocs de longueur inférieure à la limite inférieure apparaissent dans les blocs communs. Cela signifie que ces blocs appartiennent à des blocs communs de longueur supérieure à la valeur seuil, mais qu'ils ont été rognés pour éviter des recouvrements.

3.2. *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.

À 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 », ou plus simplement les « pivots » de la comparaison.

3.3. *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.

À 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.

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 est « libéré » par un auteur afin d'être réemployé plus loin dans le même texte, sans commettre de répétition.

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

4. Interface de visualisation

Pour faciliter la lecture des résultats, nous avons programmé une interface de visualisation qui comporte trois fenêtres (voir figure 3) : deux fenêtres dans la partie supérieure, l'une destinée au texte source, l'autre au texte corrigé, et une fenêtre dans la partie inférieure dont le contenu peut varier comme nous allons le voir ici.

4.1. *Partie supérieure : les textes*

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.

Un bouton permet ensuite de lancer la comparaison au moyen de l'algorithme précédemment décrit. Les résultats s'affichent alors 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.

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, il est possible, en cliquant sur un pivot, de faire automatiquement défiler le texte homologue, dans l'autre fenêtre, jusqu'à ce que le pivot correspondant soit mis en regard du premier.

4.2. *Partie inférieure : information d'usage*

Le contenu de la fenêtre inférieure est spécifié au moyen des différents onglets qui apparaissent au bas de l'interface : TRANSFORMATIONS, COMMENTAIRES, LÉGENDE, PARAMÈTRES.

4.2.1. *Transformations*

Par défaut, l'onglet « TRANSFORMATIONS» est activé et la fenêtre contient l'ensemble des transformations qui font passer du texte source au texte corrigé, à savoir l'ensemble des insertions, des suppressions, des remplacements et des déplacements.

4.2.2. *Légende*

L'onglet « LÉGENDE » fait apparaître la légende de l'interface, c'est-à-dire la signification des couleurs, par exemple ici, bleu pour les insertions et les suppressions, vert pour les remplacements et souligné pour les déplacements. Au reste, il est loisible, de modifier manuellement les couleurs des insertions, des suppressions et des déplacements, ainsi que le style des déplacements.

4.2.3. *Paramètres*

Nous avons précédemment mentionné trois paramètres, l'un porte sur la taille minimale des blocs maximaux recensés, les deux autres, sur le lissage au cours du calcul des insertions, des suppressions et des remplacements. Ces trois paramètres sont accessibles dans la fenêtre du bas, à l'aide de l'onglet « PARAMÈTRE». On se trouve alors en mesure de modifier ces paramètres à volonté.

4.2.4. *Commentaires*

Enfin, l'onglet « COMMENTAIRES» fait apparaître une fenêtre vide où il est possible d'insérer des notes réutilisables par la suite. De même, on peut coller des parties du texte, ou des transformations, de façon à préparer un article ou une analyse.

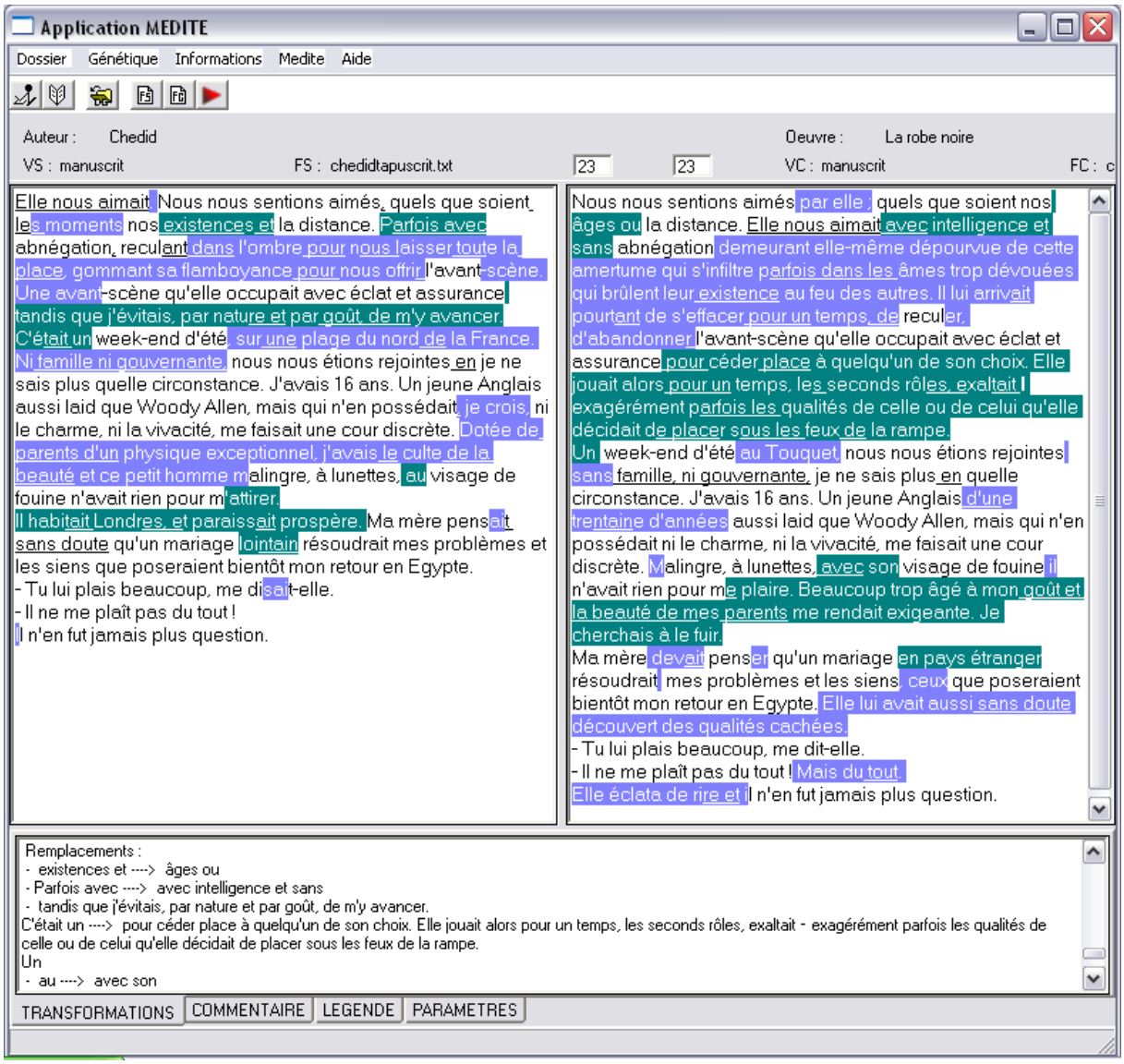

Figure 3. Interface de visualisation de MEDITE

Au terme de cette présentation de l'interface de MEDITE, il faut souligner qu'une fois le travail terminé, l'ensemble des textes, des blocs détectés et des commentaires, sont stockés dans un fichier XML dont le contenu peut être exploité par des procédures d'analyse statistique. De plus, ce fichier peut être rechargé automatiquement dans l'interface, ce qui évite d'avoir à exécuter plusieurs fois l'algorithme de comparaison sur de longs textes comme des romans ou des essais philosophiques.

5. Interprétation génétique

5.1. *Analyse d'un passage : une nouvelle d'Andrée Chedid* La robe noire

Reportons-nous de nouveau à la copie d'écran de la figure trois et observons non plus la forme de l'interface, mais le contenu des deux fenêtres supérieures. La fenêtre gauche contient la tapuscrit du brouillon d'Andrée Chedid donné dans la figure 1 (nous dirons qu'il s'agit là de l'état 1 du texte) et la fenêtre de droite, l'état final de ce brouillon, à l'issue de toutes les campagnes de récriture.

L'application MEDITE permet une visualisation immédiate de tout ce qui se passe entre l'état initial et l'état final du texte de ce brouillon. 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.

Pour ce qui est de 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 à l'état final s'opère une vraie conversion dans la façon dont est vue et présentée la place de la mère. Si l'état initial s'efforce de mettre en valeur une générosité maternelle, la suite des suppressions et remplacements montre clairement que l'état final 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 » ; « avec abnégation » → « 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 place à quelqu'un de son choix », etc.).

5.2. *Analyse d'un dossier complet. Point de vue global sur la genèse d'un texte : une nouvelle de Pascal Quignard,* Bernon l'enfant.

L'analyse par MEDITE des 5 versions de cette nouvelle dans le déploiement de ses différents états (15 états différents au total inégalement répartis selon les versions) fait apparaître 3 grandes campagnes d'écriture à quoi s'ajoute une mise au net (V5).

– V1 à V2 : campagne que l'on pourrait qualifier de « linguistique » : l'auteur joue sur la perfectibilité de l'écriture, le texte est « amélioré » du point de vue des normes linguistiques.

Ainsi les changements suivants : « du visage » → « de son visage » ; « cette promesse ne tombe pas d'oreille d'un sourd » → « cette promesse ne tombe pas dans l'oreille d'un sourd »

– V2 à V3 : campagne d'écriture focalisée sur une thématique, celle de la beauté. Stylistiquement se développe une isotopie du *beau* (voir figure 4).

Sur la copie d'écran présentée ci-dessus (voir figure 4), le relevé des insertions fait clairement apparaître une véritable série lexicale autour de ce paradigme sémantique : insertions de « qui était très beau », « le Beau Palaiseau », « Beau », « Son bel air s'ajoute à sa beauté », « le bel adolescent », « Une fois pesé, radieux, le bel adolescent »…

Des tournures sont transformées de façon esthétisante : « le crâne » est remplacé par « la tête de mort », « l'église » par « la chantrerie de la basilique », « Il » par « Le Palaiseau », …

Enfin la *visibilité* des transformations textuelles par MEDITE fait "entendre" le travail sur les sonorités.

– V3 à V4 : campagne d'écriture thématique à nouveau autour de la passion. L'examen de la liste des transformations montre plusieurs insertions de modalités à caractère affectif et passionnel (« passionnément catholique », « déteste », « hait »).

L'interprétation de ces thèmes fait apparaître une distinction – sinon une opposition – entre « Catholiques » et « Réformés ».

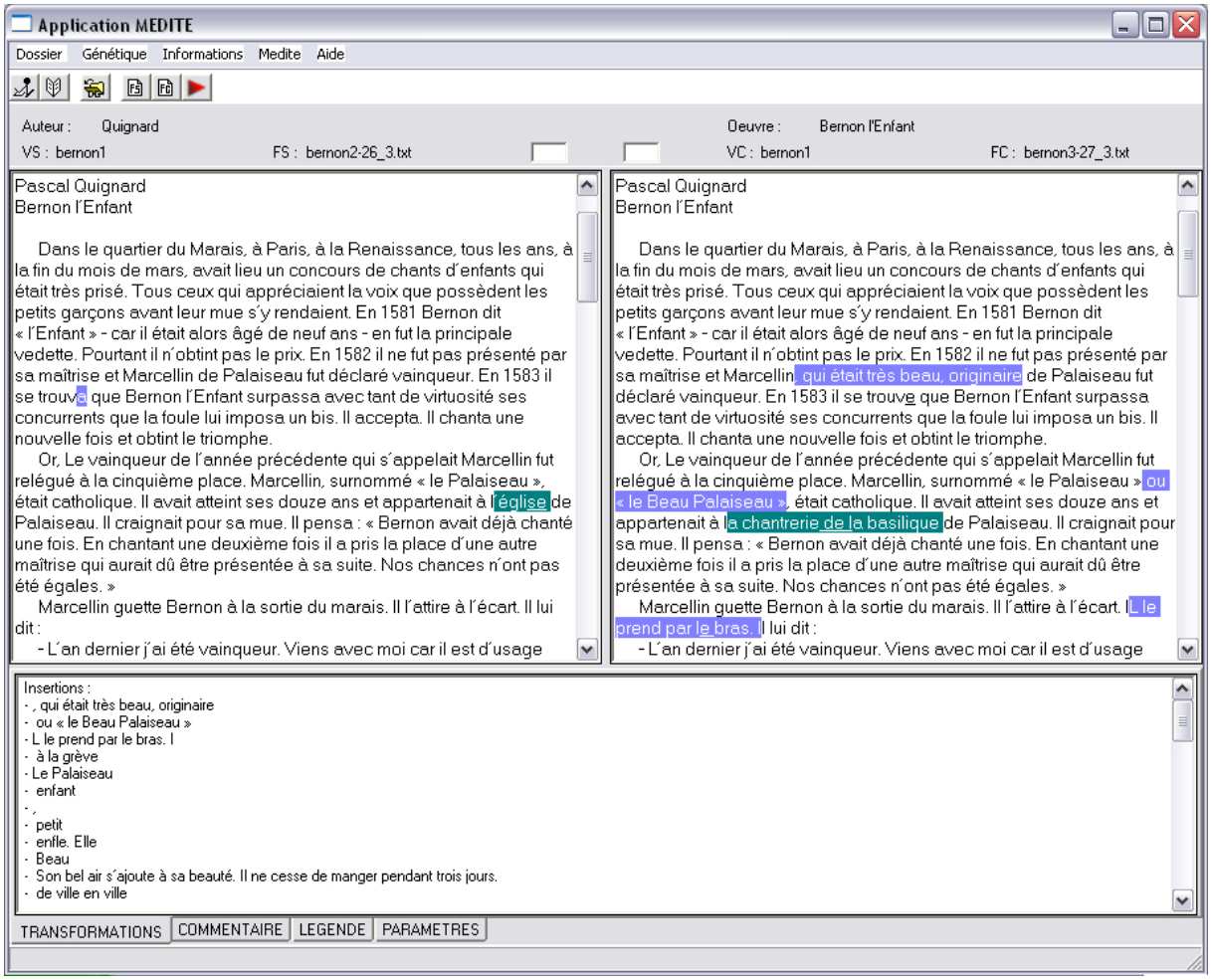

Figure 4. Activation de MEDITE sur les versions 2 et 3 de « Bernon l'enfant »

Cette analyse par MEDITE est tout à fait novatrice pour la génétique des textes. En effet, jusque là, l'étude de la genèse était centrée sur un calibrage qualitatif et plutôt exemplaire, faute de possibilité de relevés exhaustifs des données ; en conséquence, on renonçait soit à l'examen systématique d'un dossier complet dès que celui-ci était trop long, soit à la vision d'ensemble de la genèse. Et, de toute façon, on demeurait dans l'incapacité de fournir des « preuves » fondées sur comptabilisation précise.

Par le biais de MEDITE le généticien dispose d'un matériau *exhaustif*, immédiatement *visible* et surtout dont la comparaison, pièce à pièce, est *directement accessible* et productive.

6. Conclusion

MEDITE est programmé en Python. Il fonctionne actuellement sous les systèmes d'exploitation Windows et LINUX. Une version Mac OS X devrait voir le jour dans les prochains mois. Plusieurs corpus sont actuellement à l'étude avec ce logiciel : Andrée Chedid, *La robe noire*, Louis Althusser *Freud et Lacan*, Marcel Proust *Cahiers*, Pascal Quignard *Bernon l'enfant*. Dès à présent, le logiciel permet d'effectuer automatiquement des études trop fastidieuses pour être réalisées manuellement. Et, même sur des exemples aussi brefs que sur le début de *La robe noire*, ou sur la nouvelle de Pascal Quignard *Bernon l'enfant*, l'observation des transformations explicitées par MEDITE montre à l'évidence la signification du travail de l'auteur ; toutes les réécritures semblent aller dans le même sens : dans le cas du texte

d'Andrée Chedid, la jeune fille est progressivement dessaisie de toutes prérogatives ; au fil des réécritures, elle agit de moins en moins, tandis que la mère, apparemment aimante, la manipule de plus en plus... L'étude de la nouvelle de Pascal Quignard, quant à elle, fait apparaître plusieurs phases distinctes dans le travail de récriture. À ces interprétations sémantiques qui demeurent somme toute assez subjectives, on peut ajouter des études statistiques qui se font sur les transformations elles-mêmes. On devrait donc, grâce au logiciel MEDITE, ouvrir sur une linguistique de l'écrit à même d'aborder quantitativement le travail de réécriture des auteurs. C'est là un premier pas vers de nouvelles applications de l'analyse de données textuelles à la philologie.

Références

de Biasi P.-M. (2000). *La génétique des textes*. Nathan Université.

Cerquiglini B. (1989). *Éloge de la variante. Histoire critique de la philologie*. Seuil.

Chedid A. (1996). La robe noire. In *Les saisons de passage*. Flammarion.

Contat M. et Ferrer D. (Ed.) (1998). *Pourquoi la critique génétique ? Méthodes, théories*, CNRS éditions.

Fenoglio I (2001). Énonciation et genèse dans les autobiographies d'Althusser. *Genesis*, vol. (17) :131-150.

Fenoglio I. et Boucheron S. (Eds) (2002). Processus d'écriture et marques linguistiques. *Langages* vol. (147).

Ganascia J.-G. (2001). Extraction of Recurrent Patterns from Stratified Ordered Trees. In *Actes de la conférence ECML*. Springer.

Grésillon A. (1994). *Éléments de critique génétique*. PUF.

Hay L. (2002). *La littérature des écrivains. Études de critique génétique*. Corti.

Crochemore M. et Rytter W. (1994). Text Algorithms. *Approximate pattern matching* : 237-251.

Karp R.M., Miller R.E. et Rosenberg A.L. (1972). *Rapid Identification of Repeated Patterns in Strings, Trees and Arrays*. In *Proceedings of the 4th Annu. ACM Symp. Theory of Computing* : 125-136.

Landraud A.-M., Avril J.-F. et Chrétienne P. (1989). An algorithm for Finding a Common Structure Shared by a Family of Strings. *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. (11 : 8) : 890-895.

Lebrave J.-L. (1984). Le traitement automatique des brouillons. *Programmation et sciences humaines* (N° spécial), MSH.

Lebrave J.-L. (1990). *Déchiffrer, transcrire, éditer la genèse*. *Proust à la lettre.* Du Lérot : 141-162.

Sankoff D. et Kruskal J.B. (1983). *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading.