

SEXTANT, un langage de modélisation des connaissances pour la navigation textuelle

Couto Javier (1), Minel Jean-Luc (2)

(1) *Universidad de la República – Facultad de Ingeniería – Instituto de Computación*
J. Herrera y Reissig 565
Montevideo
Uruguay,
jcouto@fing.edu.uy

(2) *MoDyCO, UMR 7114 CNRS- Université Paris X Nanterre*
200 Avenue de la République
92001 Nanterre
jean-luc.minel@u-paris10.fr

Nous présentons tout d'abord notre conception de la navigation textuelle conçue comme un processus cognitif qui convoque des connaissances qui sont propres à la finalité de la navigation. Nous formulons l'hypothèse que ces connaissances peuvent être, en partie, modélisées sous une forme déclarative avec le langage SEXTANT que nous décrivons. Enfin, nous présentons deux applications qui utilisent la plate-forme NaviTexte dans laquelle le langage SEXTANT est implémenté.

Mots-clés : Navigation textuelle assistée, langage de modélisation des connaissances

In this paper, we present our approach to text navigation conceived like a cognitive process, which exploits navigation specific knowledge. We draw up the hypothesis that such knowledge can be designed in a declarative way with our language SEXTANT. Finally, two applications are described.

Keywords: Assisted navigation of texts, knowledge management language.

Conceptions de la navigation textuelle

Le terme de navigation textuelle reçoit de multiples interprétations. La plus commune renvoie inévitablement au processus mis en oeuvre par les outils de navigation utilisés pour circuler dans les documents hypertextes, c'est-à-dire la possibilité d'activer un lien pour déplacer le point de lecture ; ce déplacement pouvant être intra ou intertextuel. Plusieurs points sont à souligner dans ce type de navigation hypertextuelle. Tout d'abord, l'activation du lien est « aveugle », plus précisément aucune signalétique (en dehors d'un titre ou de l'adresse Url qui est en général peu significative) ou instructions de navigation ne sont associées au lien. Deuxièmement cette navigation est linéaire, c'est-à-dire qu'une seule voie de navigation est offerte au lecteur quand celui-ci active le lien. Autrement dit, pour chaque nœud source il existe un seul nœud cible. De notre point de vue, cela constitue une contrainte trop restrictive vis-à-vis des fonctionnalités offertes à l'utilisateur. Troisièmement, l'orientation de la navigation n'est pas indiquée explicitement ; le lecteur ne sait pas si le déplacement se fait vers l'amont ou vers l'aval¹ du texte lu, ce qui entraîne entre autre des phénomènes de désorientation cognitive (Edwards D.M., Hardman L. 1989, Cotte 2004). Dans certains systèmes, l'affichage d'une carte représentant l'ensemble du site et la localisation du point de lecture sont utilisés pour résoudre en partie ce problème (Danielson 2002). Enfin et surtout, les liens sont placés dans le corps même du texte, ce qui implique qu'il n'est pas

¹ L'orientation n'a de signification que dans le cas d'une navigation intratextuelle.

possible d'adapter les parcours dans ce texte au lecteur. En d'autres termes aucune information ou connaissances complexes ne peuvent être associées à la navigation.

Notre conception de navigation textuelle se démarque de ce type de navigation car nous considérons que circuler ou naviguer dans un texte est l'expression d'un processus cognitif qui convoque des connaissances qui sont propres à la finalité de la navigation (Minel 2003, Couto & Minel 2004). Ainsi, comme nous l'illustrerons en présentant différentes applications (section 4), un documentaliste qui doit écrire un résumé d'un texte (Endres-Niggemeyer B., Maier E., Sigel A. 1995) ne navigue pas de la même façon qu'un lecteur intéressé par l'évolution des sentiments d'un des personnages d'un roman (Mathieu 2004) ou qu'un linguiste qui explore les annotations placées par un système automatique (Pery-Woodley 2004). Ainsi, le fait qu'un texte soit maintenant numérisé et qu'il soit présenté au lecteur sur un écran peut être considéré, de notre point de vue, comme une nouvelle mutation qui place le lecteur devant de nouvelles possibilités qui restent à explorer :

« Le texte [...] offre en effet une richesse sémiotique particulière, qui fournit de multiples objets d'interprétation et de multiples pistes d'actions [...] les lecteurs n'ont pas la même démarche envers l'objet ni la même définition de cet objet, ils ne « voient » pas la même chose ». (Souchier & al. 2003).

Nous formulons l'hypothèse que la démarche du lecteur peut être assistée par l'exploitation de connaissances qui peuvent être, en partie, modélisées sous une forme déclarative. En conséquence, nous proposons le langage SEXTANT² pour exprimer ces connaissances (section 3).

Du point de vue du lecteur, la navigation textuelle que nous proposons est très différente de la navigation hypertextuelle au sens où nous considérons que le lecteur, qui active lui aussi des connaissances d'interprétation (Kintsch 2003, Baccino 2004) doit pouvoir interagir en choisissant la voie de navigation qui lui semble la plus appropriée pour sa tâche de lecture. Il est néanmoins évident que cette interaction est actuellement très limitée, car la navigation proposée reste dans les limites posées par le concepteur des modules de navigation (section 3). En ce sens, il serait peut-être plus précis de parler de « navigation textuelle assistée ».

Afin de proposer une approche systématisée à la navigation textuelle, quatre éléments sont nécessaires :

- une représentation du texte pouvant décrire différents phénomènes linguistiques ;
- la possibilité de pouvoir isoler les connaissances de visualisation et de navigation ;
- un agent (une personne, une équipe d'experts, un système, etc.) capable d'encoder ces connaissances ;
- un système qui interprète ces connaissances.

Nous n'aborderons pas dans cet article la modélisation des connaissances de visualisation, que l'on peut brièvement définir comme l'ensemble des opérations qui spécifient comment un texte est représenté sur l'écran (voir (Couto 2001, 2006) pour une présentation détaillée). Dans la section suivante nous présentons le modèle de texte que requiert cette conception de la navigation.

Représentation du texte

La représentation du texte, décrite dans un format standard XML, se divise en deux parties ; le *Corps*, où les unités textuelles, significatives pour la tâche sont délimitées, et la *Tête*, où s'expriment les relations non hiérarchiques entre ces mêmes unités.

Le Corps

Dans le *Corps*, l'élément de base de notre modèle est l'*Unité Textuelle* (UT) typée, ce qui permet d'incorporer de nouveaux éléments textuels de manière simple. Ces principes d'annotation sur lequel s'appuie NaviTexte sont classiquement ceux proposés par les standards tels que ceux de la TEI (Text Encoding Initiative).

² Par analogie avec les navigateurs du XVIII^e siècle qui ont parcouru le monde en s'orientant sur les mers avec un sextant.

Concrètement, dans le *Corps*, une unité textuelle (UT) est balisée, avec la balise <Chaine>, et des attributs, en nombre illimité, peuvent lui être attribués. Chaque UT est typée et possède optionnellement un rang. Le type peut aussi bien dénoter la fonction structurelle de l'unité en question, sa caractéristique syntaxique, sa fonction discursive. On peut remarquer que ce type d'annotation laisse une marge de liberté très grande, notamment dans la répartition des valeurs d'annotation entre le type de l'UT et les attributs de cette UT (Couto & al. 2005).

La Tête

Néanmoins ce type de délimitation des unités est insuffisant pour traiter certains phénomènes linguistiques, tel que la discontinuité ou le recouvrement. Plusieurs solutions ont été proposées qui reposent généralement sur les fonctionnalités offertes par X-Link et XPointer. Mais la généricité et la relative complexité de ces approches associées à l'absence d'outils d'éditions sophistiqués rendent leur utilisation plutôt difficile dans le cadre du Traitement Automatique du Langage (TAL). C'est pour répondre à ce besoin, l'annotation des structures complexes rencontrées en TAL, que quatre structures, qui sont déclarées dans la *Tête*, ont été définies (Couto 2006). Ces quatre structures sont nommées *Ensemble*, *Séquence*, *Reference* et *Grappe* et elles permettent de déclarer de nouveaux éléments composés d'unités textuelles du *Corps* du texte. De plus, pour chacune de ces structures, des opérations de visualisation et de navigation prédéfinies sont en cours de développement.

Un *Ensemble* déclare un ensemble non ordonné d'UT pour lesquelles existe, du point de vue de l'annotateur, une relation d'équivalence. Par exemple, des UT avec des étiquettes morphosyntaxiques différentes peuvent exprimer un même thème.

Une *Reference* décrit une relation orientée entre deux UT et une opération de navigation prédéfinie est associée à cet objet. Cette opération va du référé au référent. Typiquement une *Reference* permet de représenter le lien entre une anaphore et son référent discursif. Un autre exemple d'utilisation est la représentation des relations rhétoriques entre deux propositions comme le propose la Rhetorical Structure Theory (Thompson & Mann 1988).

Une *Séquence* permet de décrire des éléments discontinus dans un texte. Plus formellement, une *Séquence* est une suite ordonnée d'éléments à laquelle l'annotateur attribue une cohésion. L'intérêt de ce type de structure peut être illustré sur différents exemples.

Le premier exemple est illustré par le besoin d'annoter un syntagme verbal dont la continuité est par exemple brisée par la négation. Ainsi, il n'est pas possible dans le *Corps* d'indiquer que dans la suite « ne sont pas stockées », le syntagme verbal composé de « sont » et de « stockées » constitue une seule unité, sans y inclure la marque de la négation. En effet, il est tout à fait possible de déclarer les unités indépendamment et de les inclure dans une autre unité, mais ce choix de segmentation ne correspond pas à la description linguistique visée. Or, l'objectif est de visualiser, avec une même couleur de fond par exemple, ces deux unités et de pouvoir déclencher la même opération de navigation à partir d'une de ces unités. La déclaration d'une *Séquence* composée de deux éléments : « sont » et « stockées », offre cette possibilité.

Les cadres thématiques (Porhiel 2003) constituent un deuxième exemple de l'intérêt de cette structure puisqu'elle permet de déclarer les introducteurs de cadre comme appartenant à une même unité.

Le dernier exemple concerne les chaînes de référence lexicales. Une chaîne de référence lexicale est constituée par l'ensemble des syntagmes nominaux qui réfèrent à un même objet. Ainsi, dans un article de presse (Le Figaro, le 16 juillet 2004) sur l'amnistie fiscale³, on trouve pour référer à « La taxe sur les fonds rapatriés en France », dix-sept corrélatifs linguistiques qui réfèrent au même référent dont par exemple « La taxe sur les fonds rapatriés en France », « une taxe sur les fonds placés à l'étranger et rapatriés en France », « une telle mesure », « elle », etc. La déclaration d'une *Séquence* composée de toutes ces unités textuelles, et qui concrétise la chaîne de référence lexicale, permet d'offrir au lecteur un parcours entre ces éléments en utilisant la même opération de navigation décrite précédemment.

3 Ce texte fait partie des textes recueillis et analysés par Lita Lundquist.

Comme l'illustrent ces trois exemples, l'objet *Séquence* qui combine une structure avec une opération de visualisation et une opération de navigation offre les moyens de traiter simplement des phénomènes linguistiques très fréquents.

Le dernier type d'objet, *Grappe* est utilisé pour construire des relations multiples entre des UT. Il correspond exactement à la notion mathématique d'un graphe (Berge 1958) où les nœuds, qui représentent des UT sont liés par des arcs qui représentent les relations entre ces nœuds. Un *Grappe* permet ainsi de représenter un index comme on en trouve par exemple en fin d'ouvrage et permet ainsi de mettre en œuvre une autre conception de la navigation, plus figée que celle que nous proposons, comme celle que propose Nazarenko (2004). Ainsi, chaque entrée générique de l'index est un nœud du *Grappe*, les relations entre les entrées génériques et spécifiques sont représentées par les arcs et feuilles du graphe sont les UT du texte dont la chaîne lexicale a pour valeur une occurrence du terme indexé.

Le langage SEXTANT

Le langage SEXTANT a pour finalité d'offrir des fonctionnalités à la fois suffisamment génériques tout en proposant une sémantique qui se focalise sur l'essentiel du processus de visualisation et de navigation dans les textes, à l'inverse de langages de transformation ou de programmation comme, par exemple, XSLT (EXtensible Stylesheet Language) ou XPATH. Notre langage est donc de type déclaratif et propose des opérations prédéfinies.⁴

Les vues d'un texte

Le fait de pouvoir afficher un texte de manières différentes, et que chaque manière (*vue du texte*) comporte des indications précises sur les différentes options d'affichage (*opérations de visualisation*) et sur les interactions que l'utilisateur peut effectuer (*opérations de navigation*) constitue l'épine dorsale de notre approche. De plus, une vue d'un texte ne montre pas nécessairement tous les constituants d'un texte ; il peut s'agir d'une vue partielle se focalisant sur certains aspects spécifiques ou phénomènes présents dans celui-ci. Cela constitue, en quelque sorte, la vue d'un filtrage du texte.

Afin de présenter une approche systématisée des différentes vues, nous proposons une classification selon leur *type* et leur *contenu*. Les types possibles sont : *linéaire*, *arborescente* et *grappe* tandis que les contenus possibles sont : les *chaînes lexicales* et les *annotations*. Il en résulte qu'il existe six combinaisons possibles.

Certes, d'autres types de vues à ceux présentés ici sont envisageables, comme les vues basées sur la technique « Focus + Context » (Lamping et Rao 1996) (Dieberger et Russell 2002), par exemple ; ou d'autres plutôt *ad-hoc* comme la vue « docball » (Crestani et al. 2002), qui montre la structure hiérarchique d'un document. Néanmoins, le choix des types linéaire, arborescente et grappe correspond à la représentation de texte proposée, et constitue, de notre point de vue, un bon point de départ, pouvant s'enrichir des propositions et des développements postérieurs.

Modules de connaissances et descriptions de vue

Les éléments constitutifs d'une vue sont spécifiés dans une *description de vue*. Plusieurs descriptions de vue peuvent être rassemblées dans une entité cohérente d'après l'encodeur des connaissances, nommée *module de connaissances*. Nous pouvons concevoir la création d'une vue comme l'application d'une description de vue à un texte déterminé. Par analogie, l'application d'un module de connaissances à un texte implique la création d'un ensemble de vues. En conséquence, toute vue est liée à un texte, à une description de vue et, indirectement, à un module de connaissances.

⁴ Voir (Couto 2006) pour une description détaillée du langage de modélisation.

Une description de vue est identifiée dans le module par son nom. Afin de la définir, l'encodeur doit indiquer :

- le type de vue selon les types de vue disponibles : linéaire, arborescente et graphe ;
- le contenu de la vue selon les contenus disponibles : chaînes lexicales et annotations ;
- ses paramètres, selon le type de représentation ;
- ses contraintes de création (*i.e.* des conditions d'appartenance à la vue, à vérifier par les unités textuelles du texte) ;
- un ensemble d'opérations de visualisation ;
- un ensemble d'opérations de navigation ;
- un ensemble d'opérations de coordination.

Le fait de pouvoir créer des vues partielles d'un texte introduit le besoin de contraintes. Il s'agit de conditions sur les UT.

Le langage de conditions

Une partie importante de SEXTANT est le langage de conditions. Par exemple, on utilise une condition pour exprimer des contraintes d'appartenance d'une UT à une vue, pour indiquer les UT sur lesquelles une mise en relief s'applique, ou bien pour préciser la cible et la source dans la description d'une opération de navigation. Le langage de conditions est composé de *conditions simples*, de *conditions d'existence sur les éléments des UT* et de *conditions sur la hiérarchie*.

Les *conditions simples* portent sur les attributs et sur les annotations des UT. Pour ce type de conditions, nous utiliserons une notation proche de la notion de patron. On définit un opérateur *UT* comportant cinq opérands qui correspondent aux propriétés suivantes d'une UT : le *Type*, le *Numéro*, le *Rang*, les *Annotations* et la *chaîne lexicale*. Avec les trois premiers opérands on dénote des contraintes d'égalité, d'inégalité, d'ordre (inférieur et supérieur), de préfixe, de suffixe et de sous-chaîne par rapport à des valeurs. De même pour le cinquième opérande. Le quatrième opérande est utilisé pour indiquer l'existence ou non-existence d'annotations, que ce soit un nom d'annotation, une valeur ou un couple nom d'annotation – valeur.

Pour les *conditions d'existence UT*, un opérateur sans arguments est défini pour chaque élément (*cf.* tableau ci-après).

<p>existeAnnotations : teste si l'ensemble d'annotations d'une UT n'est pas vide ; existeChaîneLexicale : teste si la chaîne lexicale d'une UT est définie ; existeTitre : teste si le titre d'une UT n'est pas vide ; existeParent : teste si une UT a une UT parent ; existeFils : teste si la suite d'UT filles d'une UT n'est pas vide.</p>

Tableau 1 : Opérateurs d'existence sur les éléments des UT.

<p>estParent : teste si une UT est le parent dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ; estFils : teste si une UT est le fils dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ; estFrère : teste si une UT est le frère dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ; estAscendant : teste si une UT est l'ascendant dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ; estDescendant : teste si une UT est le descendant dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ; contientDansTitre : teste si une UT contient dans les UT du titre une UT décrite en utilisant une condition simple ; estDansTitreDe : teste si une UT appartient aux UT du titre d'une UT décrite en utilisant une condition simple ;</p>
--

Tableau 2 : Opérateurs portant sur le rapport hiérarchique des UT.

Pour les conditions où se joue le rapport entre les UT dans la hiérarchie, des opérateurs unaires spécifiques sont définis. Ces opérateurs prennent comme argument une condition simple. Le tableau ci-dessus montre les opérateurs définis pour tester des conditions sur le rapport hiérarchique des UT.

Les conditions peuvent se combiner en utilisant les opérateurs classiques *OU*, *ET* et *NON*, de la logique. Voici un exemple d'expression du langage qui exprime la condition suivante : « Les UT de type *SN* comportant une annotation de nom *Référent discursif*, tel qu'il existe dans les ascendants une UT de type *Paragraphe* qui ne comporte pas une annotation de nom *Étiquette Sémantique* et valeur *Conclusion* »

$UT(\text{Type} = \text{SN}, *, *, \{(R\acute{e}f\acute{e}r\acute{e}nt\ discursif, *)\}, *)$ ET $estDescendant(UT(\text{Type} = \text{Paragraphe}, *, *, \{\neg\exists(\acute{E}tiquette\ S\acute{e}mantique, Conclusion)\}, *))$

Tableau 3 : Exemple d'utilisation du langage de conditions

Les opérations de SEXTANT

Des trois types d'opérations possibles (visualisation, navigation et coordination), nous nous focalisons sur les opérations de navigation. La navigation est conceptualisée comme une opération reliant une UT *source* avec une UT *cible*. La manière dont ces deux UT sont liées est fonction de quatre paramètres :

- la condition à vérifier par l'UT source ;
- la condition à vérifier par l'UT cible ;
- le type d'opération de navigation ;
- le rapport existant entre l'UT source et l'UT cible.

Une opération de navigation est définie comme une opération qui cherche l'UT cible à partir de l'UT source, en vérifiant les différentes conditions et en suivant l'orientation relative au type d'opération. La source est définie en utilisant une condition sur les UT. Implicitement, une opération de navigation est *disponible* pour une UT déterminée si celle-ci vérifie la condition exprimée par la source. La cible est déterminée à partir de deux paramètres: une condition à vérifier pour l'UT cible et le type d'opération de navigation. Une fois la source déterminée, plusieurs UT peuvent vérifier la condition de la cible, et c'est le type d'opération qui indique laquelle choisir d'entre elles. Chaque opération est donc typée avec une valeur qui appartient à l'ensemble {*premier*, *dernier*, *suivant*[*i*], *précédent*[*i*]}, *i* étant un nombre entier positif. Ces valeurs spécifient d'une part l'orientation, c'est-à-dire dans quel sens (avant ou après l'UT source) doit être effectué la recherche de l'UT cible, et d'autre part le référentiel, absolu (*premier*, *dernier*), ou relatif (*suivant*[*i*], *précédent*[*i*]), par rapport à la source. Dans le cas d'un référencement relatif, l'index *i* permet de spécifier le rang de la cible recherchée. Par exemple, le type « *Suivant*[3] » s'interprète comme la recherche, dans les UT vérifiant les conditions spécifiées pour la cible (*i.e.* les cibles potentielles), de la troisième unité textuelle située après l'unité textuelle source.

opérationNavigation	→	OpNav (nomOpérationNavigation , typeOpérationNavigation , source , cible)
nomOpérationNavigation	→	valeur
typeOpérationNavigation	→	premier dernier suivant [valeur] précédent [valeur]
source	→	condition
cible	→	condition

Tableau 4 : Grammaire correspondant aux opérations de navigation

Dans sa première version, la puissance d'expression du langage était limitée par la nécessité d'exprimer de manière absolue les conditions sur les valeurs des attributs des UT. Cette limitation

avait par exemple pour conséquence l'obligation d'écrire une opération de navigation différente pour naviguer entre chaque anaphore et son référent discursif. Dernièrement, nous avons enrichi le langage de conditions par la possibilité d'exprimer des relations entre les valeurs des attributs des UT de la source et de la cible, ce qui entraîne qu'une seule opération de navigation suffit pour traiter la navigation évoquée ci-dessus. Les opérations de coordination, que nous ne détaillerons pas, ont pour finalité de synchroniser les déplacements du point de lecture dans les différentes de vue d'un même texte. Quant aux opérations de visualisation, elles permettent de spécifier les attributs visuels (police, couleur, espacement, etc.) de chaque vue.

Implémentation dans *NaviTexte*

Une première version développée en langage Java nous a permis de vérifier la validité de nos hypothèses. La plate-forme est ainsi composée de différents sous-systèmes. Un premier sous-système se charge de construire à partir d'un texte annoté, la représentation décorée du texte ; un deuxième sous-système gère les interactions avec l'utilisateur en chargeant et en interprétant à la demande les modules de navigation (écrits dans le langage SEXTANT). Le résultat de l'interprétation est un graphe de parcours qui est projeté sur la représentation du texte. La visualisation effective est assurée par un troisième sous-système qui sélectionne dans la base des modèles une ou plusieurs formes sémiotiques⁵ {S_d} spécifiées dans le module. Il faut souligner qu'à tout moment un lecteur peut charger un module spécifique et que l'interprétation de celui-ci s'applique sur le ou les textes en cours. Enfin, des sous-systèmes spécifiques se chargent de mémoriser les actions du lecteur afin de gérer un historique.

Applications

Plusieurs projets, qui s'appuient sur plateforme *NaviTexte*, sont en cours. Ils recouvrent des finalités très différentes, qui vont du parcours de lecture comme procédé de résumé automatique, à l'apprentissage du français comme langue seconde, à l'exploration des sentiments éprouvés par un des personnages dans un roman (Mathieu 2004). Nous présentons ci-dessous les deux premières applications qui sont les plus avancées.

Parcours de lecture comme procédé de résumé automatique

Un grand nombre de systèmes de résumé automatique ont été proposés ces dernières années (Mani 2001, Minel 2003). Tous ces systèmes, fondés sur le principe de l'extraction de phrases, ont été confrontés à deux problèmes intrinsèques au procédé d'extraction. D'une part, à la rupture de la cohésion textuelle, comme par exemple la présence d'anaphores sans leur référent discursif. D'autre part, à l'adaptation du résumé aux besoins spécifiques d'un lecteur. Jusqu'à présent ces problèmes n'ont pas reçu de solutions totalement satisfaisantes. Une autre approche consiste à considérer le processus de résumé comme un cheminement, plus exactement un parcours de lecture, dans le texte source qui soit propre au lecteur. Ainsi plutôt que de construire des fragments textuels, nous proposons des parcours de lecture spécifiques.

Un exemple de parcours de lecture est illustré par la figure 1. Il s'agit d'un texte qui a été préalablement annoté par un système automatique, comme ContextO (Minel & al. 2001) ou Linguastream (Bilhaut 2003). Les annotations sont du type « Annonce Thématique », « Conclusion », « Soulignement Auteur », etc.⁶. On peut voir sur la figure, que le système propose au lecteur quatre parcours de lecture différents, suivant que celui-ci s'intéresse plutôt aux thèmes

⁵ Nous reprenons ainsi la définition d'un texte proposée par Y. Jeanneret (2004) : « un texte est une configuration sémiotique empirique attestée, produite dans une pratique sociale déterminée et fixée sur un support ».

⁶ Voir (Minel & al. 2001) pour plus de détails.

de l'article, à son argumentation ou à ses conclusions. Ainsi, dans la continuité de sa lecture du texte, le lecteur se voit proposer, par une signalétique spécifique, des parcours spécifiques sans rupture de la cohésion textuelle puisqu'il voit à tout instant le texte complet, ce qui lui permet entre autre d'assurer la continuité référentielle (Battistelli & Minel 2006).

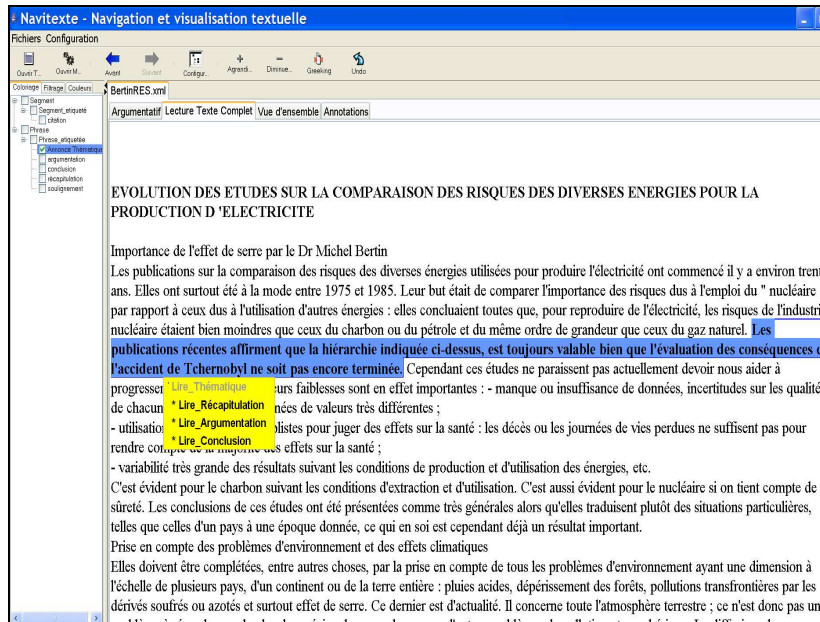


Figure 1 : Proposition de parcours de lecture

Enseignement du français en langue seconde

L'utilisation de la navigation textuelle a des fins pédagogiques est en cours à travers le projet *NaviLire*, fruit d'une collaboration entre Lita Lundquist et nous-mêmes (Couto & al. 2005, Lundquist & al. 2006). En conséquence, nous reprenons brièvement ci-dessous les principaux concepts et résultats exposés dans les deux articles cités.

Par ce procédé, par lequel le lecteur apprend à naviguer dans un texte en suivant ses différentes pistes de cohérence – basées sur la référence, sur la prédication et sur les connecteurs – nous attaquons des problèmes cognitifs cruciaux pour lire, comprendre et interpréter correctement un texte, ainsi que pour apprendre par les textes. Le premier problème consiste à identifier les référents discursifs d'un texte et d'établir les relations correctes entre les SN qui y réfèrent.

Le second problème cognitif consiste à identifier le « où veut en venir l'émetteur » du texte. Cette orientation – expressive, argumentative, et d'autre – a été qualifiée de « programme d'interprétation » (Lundquist 90, 93), étant donné qu'il s'agit d'une orientation marquée dès le début du texte, qui agit tel un « programme » qui fonctionne *du général au particulier*, et qui permet d'identifier des marques suivantes dans le texte, c'est-à-dire *du spécifique au générique*, qui « vont dans le même sens » (voir *macrostructure* et *microstructure*, Kintsch 1998). Cette identification de l'orientation, apportée entre autres par les prédications, est primordiale pour un déchiffrement correct de la cohérence sémantique et pragmatique du texte.

Finalement, les connecteurs soulignent les relations rhétoriques à établir entre des propositions ou autres séquences du texte, ce qui contribue, évidemment, de manière essentielle à établir les relations nécessaires pour construire la représentation mentale correcte du texte, c'est-à-dire, de son contenu et de son acte illocutoire prédominant, tel informer, persuader, convaincre, narrer, décrire, etc.

Dans le cadre du projet *NaviLire*, pour naviguer dans l'objet texte, nous avons isolé des unités textuelles qui permettent de spécifier des opérations de navigation, ce qui équivaut à établir des

liens de cohérence entre des unités de même nature. Comme les éléments textuels appartiennent à des types différents, la navigation permet d'une part de suivre des pistes de cohérence différentes dans un même texte, et d'autre part d'en identifier les réalisations linguistiques dans une langue donnée (ici et pour le moment, le français). Plutôt que de manipuler des structures textuelles hiérarchiques (Couto & Minel 2004), nous distinguons ici des pistes parallèles de marques textuelles qui chacune contribue à un type particulier de cohérence.

Ces types de cohérence sont fondés, *grosso modo*, sur les principes exposés dans les nombreux travaux de Lita Lundquist selon lesquels on peut distinguer dans les textes une cohérence référentielle, une cohérence prédicative et une cohérence pragmatique, fondée respectivement sur les trois actes de langage : la référence, la prédication et l'illocution qui entrent dans l'énonciation de chaque phrase (Searle 1969).

Jusqu'à présent, *NaviLire* a été mis en pratique auprès d'étudiants Danois de Quatrième année d'études dans le cursus de Language and Communication at the Copenhagen Business School. Un « pilote » a permis de réaliser une première expérience auprès de 14 étudiants, divisés en deux groupes, les « Papiristes » qui lisent le texte en utilisant les méthodes traditionnellement utilisées dans ce cursus, et les « NaviListes » qui lisent le même texte avec *NaviLire*. Les premiers résultats (cf. tableau 5), fondés sur les réponses d'un questionnaire composé de 40 questions, dont 35 sur le contenu du texte, (voir Lundquist & al. 2006 pour le détail de l'expérience) montrent que les « NaviListes » ont une performance (mesurée par le nombre de bonnes réponses aux questions) de compréhension du texte qui est supérieure au « Papiristes » pour 14 questions, identique pour 16 autres questions et inférieure pour 5 questions.

	Nombre de questions	Pourcentage
Performances des « NaviListes » supérieures au « Papiristes »	14	40
Performances des « NaviListes » identiques au « Papiristes »	16	45,7
Performances des « NaviListes » inférieures au « Papiristes »	5	14,3
Total	35	100

Tableau 5 : Comparaison des performances entre « NaviListes » et « Papiristes »

Conclusion

Nous avons montré que la navigation dans un texte, processus cognitif complexe, peut être modélisée à l'aide du langage SEXTANT. Les applications créées en utilisant la plate-forme logicielle *NaviTexte*, qui implémente un interpréteur du langage SEXTANT, sont assez hétérogènes, ce qui est pour nous une preuve de la souplesse de *NaviTexte* en tant que plate-forme d'expérimentation. Les premiers résultats obtenus, notamment dans l'application *NaviLire* illustrent le pouvoir d'expression de ce langage ainsi que sa flexibilité. Néanmoins, ces applications nous ont confrontés à de nouveaux besoins. D'une part, nous cherchons à modéliser des phénomènes discursifs qui mettent en jeu des relations d'ordre (par exemple, l'ordre chronologique). D'autre part, nous travaillons actuellement à des extensions du langage et à son implémentation dans *NaviTexte* afin de pouvoir exprimer des opérations de navigation sur ces structures discursives.

Remerciements

NaviLire a reçu un soutien financier en 2005 de l'Ambassade de France au Danemark. *NaviTexte* est soutenu par un programme ECOS-Sud (U05H01).

Bibliographie

- BACCINO, T. (2004), *La lecture électronique*. Presses Universitaires de Grenoble, Grenoble.
- BATTISTELLI, D., MINEL J.-L. (2006), « Les systèmes de résumé automatique : comment assurer une continuité référentielle dans la lecture des textes », in G. Sabah (Ed.) *Compréhension des langues et interaction*, p. 295-330.
- BERGE, C. (1958), *Théorie des Graphes*, Dunod, Paris, France.
- BILHAUT, F. (2003), « The Linguastream Platform », *Proceedings of the 19th Spanish Society for Natural Language Processing Conference (SEPLN)*, Alcalá de Henares, Spain, 339-340.
- COTTE, D. (2004), « Leurres, ruses, désorientation dans les écrits de réseau : la métis à l'écran. », *Communication & langages*, n°139, Avril 2004, p. 63-74.
- COUTO, J. (2001), « ContextO, Los sistemas de exploracion contextual de cara al usuario », Mémoire de Master, Université de la République, Uruguay.
- COUTO, J., (2006), *Modélisation des connaissances pour une navigation textuelle assistée. La plate-forme logicielle NaviTexte*. Thèse en cours, Université Paris-Sorbonne.
- COUTO, J., MINEL, J.-L. (2004), « Outils dynamiques de fouilles textuelles », *Actes de RIAO 2004*, Avignon, p. 420-430.
- COUTO, J., LUNDQUIST, L., MINEL, J.-L. (2005), « Naviguer pour apprendre », *EIAH 2005*, Montpellier, p. 45-56.
- CRESTANI, F., DE LA FUENTE, P., VEGAS, J. (2002), « Experimenting with graphical user interface structured document retrieval », *SIGIR'02*, août 2002, Tampere, Finlande.
- DANIELSON, D.R. (2002), « Web navigation and the behavioral effects of constantly visible site maps ». *Interacting with Computers*, 14(5), p. 601-618.
- DIEBERGER, A., RUSSELL, D.M. (2002), « Exploratory navigation in large multimedia documents using Context Lenses », in *Proceedings of the 35th Hawaii International Conference on System Sciences*, Hawaii.
- EDWARDS, D.M., HARDMAN, L., (1989), « Lost in hyperspace: cognitive mapping and navigation in a hypertext environment », in R. McAleese (Ed.), *Hypertext: Theory and Practice*. Oxford, England: Intellect Books, p. 105-125.
- ENDRES-NIGGEMEYER, B., E. MAIER, A. SIGEL. (1995), « How to implement a naturalistic model of abstracting : four core working steps of an expert abstractor », *Information Processing & Management*, 31(5), p. 631-674.
- KINTSCH, W. (1998), *Comprehension. A Paradigm for Cognition*, Cambridge, Cambridge University Press, 1998/2003.
- JEANNERET, Y. (2004), « Le procès de la numérisation de la culture », *Protée*, Vol 32 : 2 Paris..
- LAMPING, J., RAO, R. (1996), « The Hyperbolic Browser : A Focus + Context technique for visualizing large hierarchies », dans *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publishers, 1999, p. 382-408.
- LUNDQUIST, L. (1990), *L'analyse textuelle. Méthode, exercices*, Copenhagen, Nordisk Forlag.
- LUNDQUIST, L. (1993), « La Cohérence textuelle argumentative, illocution, intention et engagement de consistance », *Revue québécoise de linguistique*, vol. 22-2, 1993, p. 109-138.
- LUNDQUIST, L., MINEL, J.L., COUTO J. (2006), « NaviLire, Teaching French by Navigating in Texts », *accepté à IPMU 2006*, Paris, Juin 2006.
- MANI, I. (2001), *Automatic Summarization*. John Benjamins Publishing Company, Amsterdam.
- MATHIEU, Y. Y. (2004) *Linguistic Knowledge and Automatic Semantic Representation of Emotions and Feelings.* », *International Conference on Information Technology*, ITCC 2004, IEEE Computer Society, p. 314-318.
- MINEL, J.-L., CARTIER, E., CRISPINO, G., DESCLÉS, J.-P., BEN HAZEZ, S., JACKIEWICZ, A., (2001), « Résumé automatique par filtrage sémantique d'informations dans des textes, Présentation de la plate-forme FilText », *Technique et Science Informatiques*, n°3, Hermès, Paris, p. 369-396.
- MINEL, J.-L. (2003), *Filtrage sémantique. Du résumé à la fouille de textes*. Hermès, Paris, France.
- NAZARENKO, A. (2005), « Sur quelle sémantique reposent les méthodes automatiques d'accès au contenu textuel », in A. Condamines (Ed.), *Sémantique et Corpus*. Hermès, France, p. 211-239.
- PERY-WOODLEY, M.-P. (2005), « Discours, corpus, traitements automatiques », in A. Condamines (Ed.), *Sémantique et Corpus*. Hermès, France, p. 177-205.
- PORHIEL, S (2003), « Les introducteurs de cadre thématique », *Cahiers de Lexicologie* 83, 2 : p. 1-36.
- SEARLE, J. (1969), *Speech Acts, An Essay in the Philosophy of Language*, Cambridge, Cambridge University Press..
- SOUCHIER E., JEANNERET Y., LE MAREC J. (2003), *Lire, écrire, récrire : objets signes et pratiques des médias informatisés*, Direction du Livre et de la Lecture, Bibliothèque publique d'information, Paris.
- THOMPSON, S., W. MANN (1988), « Rhetorical structure theory, a framework for the analysis of texts », *IPRA Papers in Pragmatics*, p. 79-105.