Articulation des traitements en TAL

Principes méthodologiques et mise en œuvre dans la plate-forme LinguaStream

Antoine Widlöcher — Frédérik Bilhaut

GREYC, Université de Caen, CNRS UMR 6072
Boulevard du Maréchal-Juin, B.P. 5186, F-14032 Caen Cedex
Antoine.Widlocher@info.unicaen.fr, Frederik.Bilhaut@info.unicaen.fr

RÉSUMÉ. *Différents travaux en TAL font apparaître la nécessité d'articuler, au sein de chaînes complexes, divers traitements mettant en jeu une pluralité d'objets linguistiques et de méthodes d'analyse. La plate-forme LinguaStream propose une architecture et un ensemble d'outils visant à faciliter la mise en œuvre de tels « assemblages » : modèles génériques d'analyse (grammaires, transducteurs, projection de lexiques...), ponts vers des modules externes (analyseurs morphologiques ou syntaxiques...), outils lexicométriques, outil de visualisation des annotations, etc. La conception de LinguaStream et son exploitation dans différents projets, nous ont conduits à formuler différents principes méthodologiques que nous tentons ici d'expliciter. Nous décrivons ensuite succinctement trois applications développées avec LinguaStream, selon cette méthodologie : analyse de cadres de discours, recherche d'information géographique et segmentation thématique par des méthodes de fouille de texte.*

ABSTRACT. *Current advances in NLP show the need for assembling into complex processing streams, sets of treatments involving a variety of lingustic objects and analysis methods. To achieve this goal, the Linguastream platform offers both a software architecture and a large set of tools: generic analysis models (grammar-based analysers, transducers, projection of lexicons...), bridges toward external modules (taggers, syntactic analysers...), lexicometric procedures, devices for visualising text annotations, etc. The design of LinguaStream, and its use in different projects, rely on a set of methodological principles discussed in this paper. Then, three applications exploiting the platform and this methodology are shortly presented: a lingusitic analyser of discourse frames, a geographical search engine, and a thematic segmenter using text mining techniques.*

MOTS-CLÉS : *articulation de traitements sur corpus, méthdologie du TAL, plate-forme de TAL.*

KEYWORDS: *articulating corpus processing tasks, NLP methodology, NLP platform.*

1. Introduction, questionnements et positionnements

Le développement du travail sur corpus, qu'il s'agisse de mettre en œuvre des systèmes à visée applicative ou d'étudier expérimentalement des modèles linguistiques, rend nécessaire l'utilisation d'outils appropriés de TAL, dispositifs expérimentaux pouvant soutenir et simplifier les cycles de conception−expérimentation−évaluation.

De tels outils devront d'une part autoriser l'assemblage d'un ensemble varié de traitements relatifs à différents « paliers » et facettes de la langue (analyses morphologique, phonologique, syntaxique, sémantique, discursive...), éventuellement complétés, en aval, par des applications exploitant les structures et informations extraites du texte (outils d'ingénierie des connaissances, par exemple). D'autre part, des ensembles variés de méthodes d'analyse et de ressources linguistiques et logicielles sont aujourd'hui disponibles au sein de la communauté, qu'il serait évidemment fécond de pouvoir assembler dans l'optique de tâches nouvelles et de complexité croissante.

De ce double mouvement naît l'exigence de « plates-formes de TAL », dont la communauté scientifique propose déjà une assez grande variété. Sans prétendre à l'exhaustivité, un rapide aperçu de quelques-unes de ces plates-formes laisse d'emblée apparaître un ensemble de tendances, de priorités, de choix possibles dans leur réalisation.

Notons tout d'abord l'existence de « collections d'outils » fonctionnant souvent sur le principe du *repository* et visant à capitaliser et centraliser des outils *a priori* disparates. Nous pensons en particulier ici au projet OpenNLP[1]. Si de telles initiatives présentent un intérêt évident pour fédérer un certain nombre d'efforts, elles laissent cependant finalement ouverte la délicate question de l'articulation des traitements.

L'architecture UIMA (Ferrucci et Lally, 2004) vise précisément à garantir la cohérence informationnelle des chaînes de traitements, en proposant une *infrastructure commune* et *ouverte* assurant l'homogénéité des enchaînements, des annotations et de l'accès à ces annotations. Toutefois, la visée très générale de cet environnement en fait une architecture abstraite de relativement bas niveau ne proposant, en tant que telle, aucun module d'analyse de TAL immédiatement utilisable. La mise en œuvre des traitements en vue d'une tâche donnée reste ainsi à la charge du concepteur, qui doit se munir de composants d'analyse développés par lui-même ou par des tiers, ces derniers restant à l'heure actuelle relativement peu nombreux et très spécifiques[2].

Adoptant une philosophie significativement différente, un certain nombre de plates-formes, que nous qualifierons d'*intégrées*, proposent au contraire un ensemble plus ou moins riche de composants, accompagnés d'un environnement permettant de les assembler en chaînes complexes. La plus connue d'entre elles est probablement GATE (Cunningham *et al.*, 2002), autour de laquelle se sont développés différents projets, notamment à visées applicatives (extraction d'information, résumé, constitu-

1. `http://opennlp.sourceforge.net`
2. Voir par exemple les composants disponibles sur `http://uima.lti.cs.cmu.edu/UCR`.

tion d'ontologies...), ainsi que, corrélativement, différents modules de traitement linguistique ou de gestion des connaissances. D'autres plates-formes ont également vu le jour, souvent guidées par des objectifs ou des dispositifs particuliers. Citons, par exemple, Tilt (Guimier de Neef *et al.*, 2002), plate-forme « industrielle » orientée notamment vers le multilinguisme et le « contrôle multicritère » des analyses ; Ogmios (Hamon *et al.*, 2007), conçue en vue d'applications de type extraction d'information, avec un souci de performance et de robustesse vis-à-vis de l'hétérogénéité des documents accessibles sur le Web ; ou encore ASV-Toolbox[3] qui accorde une place importante à la fouille de textes.

Enfin, d'autres plates-formes se distinguent par le choix de ce que nous nommerons ici un *modèle d'analyse* particulier, c'est-à-dire un *métamodèle*, formalisé, permettant de spécifier une certaine classe de modèles linguistiques et de les projeter sur corpus. Nous pensons ici particulièrement à la famille des plates-formes mettant en œuvre des modèles à base d'automates, enchaînés en « cascades » pour constituer des traitements complexes, tels que Intex/Nooj (Muller *et al.*, 2004) et Unitex (Paumier, 2003). On mentionnera également ContextO (Minel *et al.*, 2001), qui propose un environnement logiciel consacré au modèle de l'exploration contextuelle (Desclés *et al.*, 1997), intégrant des prétraitements et une gestion de l'architecture des documents traités, avec une application au résumé et au filtrage de textes (Minel, 2002).

De ce rapide panorama il ressort distinctement que la conception d'une « plate-forme de TAL » peut répondre à des objectifs assez divers et dépend de contraintes, tant informatiques que linguistiques, complexes, se traduisant finalement par des choix et des équilibres particuliers.

C'est dans ce contexte général que nous développons depuis 2001, au sein du GREYC, la plate-forme LinguaStream[4] (Bilhaut et Widlöcher, 2006). Par rapport aux différentes tendances qui viennent d'être évoquées, LinguaStream peut être très rapidement caractérisée comme une plate-forme intégrée, reposant sur une infrastructure ouverte facilitant l'intégration de composants exogènes, proposant différents (méta-)modèles d'analyse, et attachant une attention particulière aux problèmes nés de l'hétérogénéité des modèles et des composants.

La conception de LinguaStream, et son utilisation dans diverses applications, nous ont conduits à appréhender progressivement un ensemble de problèmes méthodologiques que nous envisageons dans le présent article, en nous focalisant en particulier sur les aspects suivants :

– différentes questions se posent tout d'abord en matière d'application de *principes de génie logiciel* : comment assurer la coopération entre modules *a priori* hétérogènes, en particulier dans une optique d'ouverture de la plate-forme ? Quelle architecture mettre en œuvre (*pipeline*, agents, données partagées...) ? Comment décrire

3. http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/
4. http://www.linguastream.org

un assemblage de traitements (enchaînement de commandes, scripts, représentation visuelle...) ?

– différentes *questions plus immédiatement linguistiques* se posent également : quelle palette de traitements doit être proposée par la plate-forme ? S'agira-t-il de modules de type « *black box* » ? de modules dédiés à des tâches particulières ? de métamodèles d'analyse ? Et, dans ce dernier cas, s'agira-t-il d'un modèle unique ou d'un ensemble de modèles ?

– les traitements de TAL se traduisent d'autre part généralement par des annotations attachées au texte et d'autres questions sont soulevées par le choix des *modèles d'annotation et des formats d'échanges* : quel format retenir ? Quel en est le pouvoir expressif ? Comment les informations sont-elles codées et attachées au texte ? Comment maîtriser la complexité inhérente à l'empilement des traitements ?

– enfin, il est également nécessaire de s'interroger sur les *cycles d'expérimentation* en eux-mêmes : quels outils sont proposés à l'utilisateur pour prendre connaissance des analyses réalisées, exploiter l'information produite, en évaluer la pertinence et la qualité ? Quels dispositifs expérimentaux simplifieront les allers-retours entre édition des règles linguistiques, projection sur corpus et évaluation ?

Ces différentes problématiques seront envisagées dans la section 2. La section 3 présentera trois expériences menées avec LinguaStream et illustrant certains de ses principes fondateurs. Nous conclurons par un premier bilan et quelques perspectives.

2. La plate-forme LinguaStream : principes et mise en œuvre

Des différentes problématiques que nous venons d'évoquer naissent non seulement des exigences en matière de plates-formes techniques, mais aussi le besoin d'outils méthodologiques aidant à maîtriser les systèmes complexes qui en résultent. Nous tenterons ici de traiter ces deux questions en parallèle : pour chaque axe de notre problématique nous dissocierons ce qui relève des principes généraux de ce qui tient aux modalités de leur réalisation dans la plate-forme LinguaStream. Il est évident que l'ensemble de ces résultats est le fruit d'une seule et même démarche. Néanmoins nous revendiquons volontiers la portée plus générale des propositions méthodologiques, que nous avons cherché à exprimer de façon aussi indépendante que possible de leur mise en œuvre logicielle. Cette dernière ne se veut en effet ni définitive ni exclusive, les mêmes principes pouvant être pris en charge par d'autres implémentations.

D'une manière générale, LinguaStream propose une architecture et un ensemble d'outils favorisant une approche expérimentale du TAL. Reposant sur le principe d'enrichissement incrémental des documents électroniques et visant l'articulation de traitements sur corpus, elle facilite la conception et l'évaluation de chaînes de traitements complexes, par assemblage visuel de modules d'analyse de types et de niveaux variés : morphologique, syntaxique, sémantique, discursif, etc. Chaque palier de la chaîne de traitement se traduit par la découverte et le marquage de nouvelles informations, sur lesquelles pourront s'appuyer les analyseurs subséquents. Un environnement de dé-

veloppement intégré (*cf.* figure 1) permet de construire visuellement et de tester les dispositifs ainsi construits.

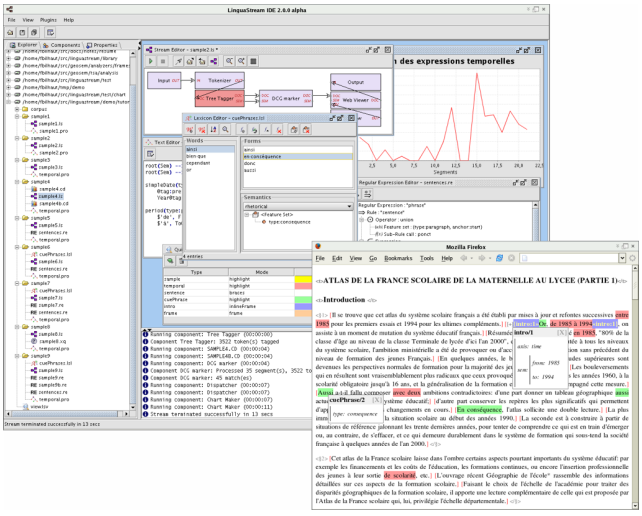

Figure 1. *Environnement intégré de LinguaStream et visualisation sous Firefox*

2.1. *Articulation de traitements*

2.1.1. *Principes*

Nous avons rappelé la nécessité, aujourd'hui communément admise, d'organiser l'articulation des traitements pour faire face à la diversité d'objets et d'objectifs auxquels le TAL est confronté, articulation devant prévaloir sur la conception d'outils *ad hoc* dédiés à des objets et à des tâches particulières, toujours difficiles à capitaliser et à réutiliser dans d'autres contextes. De toute évidence, pour que les résultats d'un champ particulier servent efficacement les besoins des recherches connexes, préoccupation légitime en toute démarche scientifique, il est nécessaire de rendre les traitements interopérables.

Mais l'interopérabilité considérée sur un plan purement logiciel, c'est-à-dire reposant sur des formats d'échange et des architectures unifiées, n'est en la matière pas suffisante : il nous paraît plus important encore de considérer la question de l'*interdépendance des analyses*, elle-même liée aux relations entre différentes structures linguistiques, et fondamentalement indépendante des modalités d'interaction entre les modules logiciels qui les réalisent. Par exemple, s'il est communément admis que les analyses morphologique et syntaxique se nourrissent mutuellement, différentes possibilités sont envisageables pour les articuler sur un plan logiciel : séquentiellement (comme le fait par exemple l'analyseur Syntex (Bourigault et Fabre, 2000)), itérati-

vement (à la manière de (Vergne, 2001)), à l'aide d'un moteur d'inférences ou de résolution de contraintes, etc. Cette question est bien sûr omniprésente en TAL, et ce à tous les niveaux linguistiques. On peut également mentionner le cas de l'analyse des chaînes de coréférence qui apporte des indices pour la délimitation de segments thématiques, qui peut elle-même, inversement, soutenir une telle analyse.

Pour autant, toute entreprise de modélisation repose sur le principe de « décomposition des problèmes », fût-elle arbitraire, indispensable à l'appréhension des problèmes complexes. En conséquence, du point de vue de la conception de modèles opérationnels, il paraît extrêmement utile de pouvoir décrire séparément des structures linguistiques distinctes mais interdépendantes, sans pour autant devoir faire d'hypothèse sur les modalités logicielles qui assureront la cohabitation entre ces modèles. Ainsi, par exemple, à supposer que les formalismes nécessaires soient disponibles, on souhaitera pouvoir concevoir la description de structures syntaxiques sur la base d'indices morphologiques, et inversement, sans devoir faire d'hypothèse sur les modalités effectives d'ordonnancement des calculs nécessaires à leur projection sur un corpus. Charge à l'outil informatique de mettre en œuvre (ou même de choisir par lui-même) un procédé logiciel susceptible de réaliser au mieux cette tâche. Cela rend absolument nécessaire l'existence de formalismes orientés vers les objets linguistiques à décrire plutôt que vers les algorithmes et l'infrastructure logicielle.

De toute évidence, même s'il n'influence pas ou que très peu le processus de modélisation, toute plate-forme concrète reposera nécessairement sur un certain paradigme d'interaction entre modules logiciels (ou, au mieux, sur une palette de paradigmes), selon les choix de ses concepteurs. Comme pour beaucoup d'outils similaires (notamment GATE et UIMA), nous avons sur ce point retenu un modèle séquentiel, répondant à l'essentiel des besoins que nous avons rencontrés, et qui sera décrit ci-après. Mais nous avons cependant veillé à ce que ce choix influence aussi peu que possible les moyens offerts à l'utilisateur pour décrire les objets linguistiques qui l'intéressent, d'abord pour les raisons que nous venons d'évoquer, mais aussi pour préserver la possibilité de remettre en cause le modèle séquentiel, sans compromettre les applications existantes de la plate-forme.

2.1.2. *Mise en œuvre*

LinguaStream permet l'assemblage des différentes étapes d'un processus d'analyse sous la forme de chaînes de traitements, correspondant à des graphes acycliques où les nœuds sont constitués des composants choisis pour réaliser chaque sous-tâche. Les arcs correspondent bien sûr aux flux informationnels qui relient ces composants, et sont plus précisément attachés aux différents points d'entrée ou de sortie de chaque composant (la figure 2 montre un exemple de chaîne de traitement, divers exemples étant par ailleurs présentés dans la dernière partie de cet article). Ce modèle, plus puissant que les *pipelines* purement linéaires de GATE, se rapproche davantage du système de flux d'UIMA. LinguaStream s'en distingue cependant en permettant la mise en place de chaînes sans aucune programmation, par le biais de l'interface (*cf.* ci-dessous) ou par l'écriture de descripteurs XML.

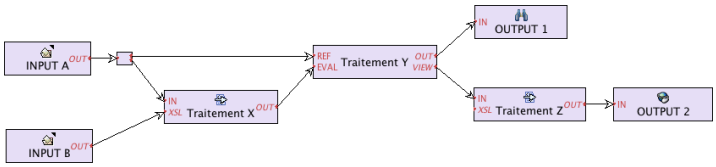

Figure 2. *Exemple de chaîne de traitement LinguaStream*

Le comportement de chaque composant est déterminé par un ensemble de paramètres, qui régissent la façon dont les flux d'entrée seront traités et dont seront produits les flux de sortie. Selon les cas, ces paramètres peuvent constituer de simples réglages (pour des composants de type « *black box* ») ou aller jusqu'à la spécification complète de la tâche à réaliser (pour des composants de type « *glass box* »)[5], notamment par le biais des *modèles d'analyse* qui seront présentés ci-après.

La séquence des actions à réaliser lors de l'exécution d'une chaîne de traitements est déterminée par un simple tri topologique des nœuds du graphe. Une chaîne de traitement est donc toujours exécutée de façon linéaire, sans qu'aucune structure de contrôle ne permette de définir dynamiquement un « chemin à suivre » (pas de retour en arrière, ni de point de choix, etc.). Ce choix s'est opéré dans l'objectif de préserver la lisibilité et la déclarativité des chaînes de traitements, qui n'ont pas vocation, de notre point de vue, à devenir des organigrammes algorithmiques. Ainsi n'autorisons-nous pas le contrôle de flux tel qu'offert, par exemple, par l'API d'UIMA. Certaines évolutions du modèle d'articulation des traitements sont néanmoins envisagées, notamment dans le but d'intégrer une notion de récursivité dans les chaînes elles-mêmes.

LinguaStream dispose d'un environnement graphique permettant d'élaborer et de tester des chaînes de traitement, par assemblage visuel des composants, ces derniers étant sélectionnés dans une palette et reliés par des flèches figurant les flux informationnels. Un éditeur de propriétés permet de définir le paramétrage de chaque composant. La plate-forme offre par ailleurs une interface de programmation Java permettant d'exploiter très facilement des chaînes de traitements dans des applications tierces. Étant représentées au format XML, elles sont également facilement exploitables elles-mêmes en tant que données.

2.2. *Approche par composants*

2.2.1. *Principes*

Comme dans beaucoup de domaines, les principes du génie logiciel s'appliquent de façon intéressante au TAL, comme l'a notamment montré (Cunningham, 2000) à

5. Nous employons ici les notions de *black* et de *glass box* au sens habituel que leur confère le génie logiciel.

travers le modèle CREOLE (qui fonde la plate-forme GATE). En particulier, la décomposition d'analyses linguistiques complexes en sous-tâches plus élémentaires permet d'assurer la *modularité* des processus de traitement, ce qui favorise tout d'abord la *réutilisabilité* des composants dans des contextes différents.

D'autre part, la décomposition en sous-processus permet de tirer pleinement profit de l'*équivalence fonctionnelle* entre des analyseurs dédiés à un même type d'objet linguistique. Pour une chaîne donnée, on pourra substituer à un composant tout autre composant *fonctionnellement équivalent*. Ceci rend notamment possible la mise en comparaison des traitements, en soumettant ces derniers à des contextes rigoureusement identiques, condition *sine qua non* d'une comparaison pertinente.

Cette exigence impose la capacité à définir des modèles d'annotation susceptibles de rapporter la diversité des analyses fonctionnellement équivalentes à des annotations respectueuses d'un modèle commun. Nous n'ignorons rien des nombreuses discussions qui animent la communauté scientifique quand il s'agit d'établir, par exemple, un jeu d'étiquettes morphologiques ou syntaxiques, ni des nombreux et délicats problèmes dont ces discussions résultent. Il nous est néanmoins apparu possible en pratique, pour des tâches bien définies, de définir un modèle d'annotation suffisamment générique pour rendre compte des résultats d'analyseurs variés dédiés à un même type d'analyse, tout en laissant cependant à chacun d'entre eux la possibilité d'enrichir les annotations réalisées selon ce modèle.

2.2.2. *Mise en œuvre*

La systématisation de cette approche nous a conduits, au fil des développements et usages de la plate-forme, à constituer un ensemble conséquent et éprouvé de composants hautement réutilisables[6]. La plate-forme permet en outre de constituer des « macrocomposants » à partir de chaînes complètes, ce qui permet de les réutiliser facilement au sein de chaînes d'ordre supérieur, et contribue à la logique générale de séparation des problèmes.

Les composants dédiés au TAL peuvent être plus ou moins opaques, et permettent pour certains d'exploiter des systèmes de traitement externes à la plate-forme : étiqueteur TreeTagger (Schmid, 1994), analyseur syntaxique Syntex (Bourigault et Fabre, 2000), ou encore bibliothèque de fouille de données MVMiner (Rioult et Crémilleux, 2004). On trouvera en outre des composants réalisant des traitements XML (XSLT, XQuery...), ou encore documentaires (extraction/génération PDF...).

Soulignons que la plate-forme nous a permis d'expérimenter en situation le principe de substituabilité de composants fonctionnellement équivalents, dont il est permis de douter *a priori*. Nous avons notamment mené deux expériences spécifiques dans ce domaine. La première concerne la substitution de différents étiqueteurs morpho-syntaxiques, sur la base d'un jeu d'étiquettes unifié pour le français. Ce dernier, défini

6. Environ une cinquantaine dans la plate-forme « standard », auxquels s'ajoutent les composants disponibles par ailleurs sous la forme de modules externes (« *plugins* »).

par nos soins, s'est révélé amplement suffisant pour de très nombreuses applications, en dépit des quelques approximations rendues nécessaires par l'intersection des jeux d'étiquettes d'origine. Dans une seconde expérience, relative à l'analyse de discours, nous avons pu substituer, dans une même chaîne de traitements, deux versions d'un même analyseur de portée des introducteurs de cadres temporels (*cf.* section 3), le premier prototype étant réalisé, de façon procédurale, à l'aide d'un langage de script, et la seconde version faisant usage du formalisme déclaratif CDML évoqué ci-après.

2.3. *Annotation par enrichissement incrémental*

2.3.1. *Principes*

Un problème majeur pour les plates-formes de TAL concerne la manière dont peuvent communiquer différents modules de traitement intégrés à une même chaîne. Pour y répondre, on pourrait, à première vue, remarquer que certaines procédures tendent à se répéter d'une application à l'autre. Ainsi, par exemple, une tâche classique d'extraction d'information sera souvent constituée, dans un ordre assez stable, d'une étape de découpage en mots, suivie d'une analyse morphosyntaxique, débouchant finalement sur l'analyse de patrons syntagmatiques. Dès lors, on pourrait être tenté de prendre acte de ces ordres classiques en définissant des interfaces dédiées à la communication entre des composants logiciels ainsi traditionnellement liés.

Il est cependant important de tenir compte du fait que ces ordres traditionnels ne reflètent qu'une partie (certes importante) des enchaînements de traitements possibles. En particulier, il est difficile de déterminer *a priori* quelles étapes d'analyses pourront être conduites à exploiter des informations issues de traitements préalables. Si nous considérons, par exemple, le cas d'une chaîne dédiée à l'analyse de structures discursives, et composée de modules opérant à différents niveaux de granularité (niveaux lexical, syntagmatique, discursif...), il est tout à fait possible que l'analyse discursive proprement dite nécessite le recours à des indications morphologiques issues de traitements réalisés très tôt, en amont de la chaîne. Chaque traitement doit alors pouvoir accéder à toute information produite en amont, sans présupposé d'échelle ou d'enchaînement classique.

Il nous semble dès lors essentiel de mettre en avant l'idée d'un *enrichissement incrémental* du texte sur lequel opèrent les différents traitements. Dans cette optique, chaque composant de l'analyse produit des objets et des annotations qui viennent enrichir la matière textuelle à laquelle il s'applique, matière transformée au sein de laquelle les traitements subséquents trouveront, en plus des objets initiaux, les marques laissées par les différentes phases d'analyse. Chaque phase revient ainsi à exprimer des contraintes sur les objets linguistiques disponibles, c'est-à-dire non pas seulement les objets initialement présents, en surface, dans le corpus (caractères, mots, ponctuations...), mais également tous les objets ayant été mis en évidence au fil de l'analyse (noms, verbes, propositions, phrases, relations syntaxiques, relations de discours, segments thématiques...). Sans communiquer directement entre eux, les différents agents

de traitement communiquent ainsi par le biais de leur environnement commun, le corpus, au sein duquel ils déposent l'information résultant de leurs opérations.

Dans la continuité des arguments avancés dans la section 2.1, on voit se dessiner ici un principe permettant de s'abstraire d'un mode particulier d'articulation des traitements, au profit d'une approche de type « tableau noir » : chaque traitement identifie ou produit des objets textuels et/ou produit des annotations qui sont mis à disposition des autres analyseurs. Dès lors, le modèle séquentiel que nous avons choisi pour LinguaStream n'est qu'une alternative parmi d'autres pour ordonnancer les traitements.

Dans cette démarche, puisqu'il est impossible de déterminer *a priori* quelles analyses devront s'appuyer sur telle ou telle information, il est nécessaire d'assurer leur compatibilité avec tout traitement ultérieur potentiel. Réciproquement, toute analyse nécessitant le recours à des informations préalablement identifiées est par principe indifférente à la manière particulière dont ces informations ont été produites. Il est donc essentiel d'assurer l'*unicité du modèle de représentation des objets et des annotations*. Il ne s'agit naturellement pas de restreindre la nature des informations pouvant être produites par chaque analyseur, mais d'homogénéiser leur format de représentation.

Sur ce point, on distingue généralement le *marquage* (ou *ancrage*) des structures proprement dites, de la *représentation symbolique* (ou *caractérisation*) qui leur est associée, l'ensemble constituant leur *annotation*. Concernant l'ancrage, on devra permettre de manipuler aussi bien des unités que des relations, en autorisant différents modes d'enchaînement, tels qu'emboîtements et chevauchements. Concernant les informations associées aux unités marquées, on distinguera clairement le *métamodèle* d'annotation (structures de traits par exemple), qui pourra être commun, des *modèles* spécifiques qu'il permet d'exprimer (étiquettes morphosyntaxiques, indications en sémantique temporelle...), qui seront propres à un type de traitement.

Il nous paraît essentiel par ailleurs de bien distinguer ce qui relève de la structure propre du document à traiter (structure dite « logique », mise en forme, etc.) de ce qui relève des objets linguistiques annotés. On distinguera, par exemple, les paragraphes pris comme objets linguistiques (et ce dans un cadre théorique donné) de leur matérialisation éventuelle dans un document structuré (HTML, TEI...). Des règles de correspondance devront cependant pouvoir entrer en jeu, règles qui seront nécessairement différentes pour chaque format de document à traiter. Ceci permet de maintenir une indépendance entre ces deux plans tout en se donnant les moyens de passer de l'un à l'autre le cas échéant, et ce par l'intermédiaire de règles adéquates.

2.3.2. *Mise en œuvre*

Conformément aux principes que nous venons d'évoquer, le modèle d'annotation mis en œuvre par la plate-forme vise à garantir une grande flexibilité et une indépendance totale vis-à-vis des différents types de documents à annoter et des différents types d'analyses linguistiques qui seront représentés. Un document est tout d'abord conçu comme une séquence d'objets indivisibles, correspondant aux unités définies par le système d'encodage utilisé pour représenter numériquement ce document sur

son support (généralement Unicode), que nous qualifions de *primitives*. Les *unités secondaires* correspondent, quant à elles, à des ensembles d'au moins deux jalons insérés au sein de la séquence d'unités primitives qui forment le document, constituant un ou plusieurs groupes d'unités primitives. Bien entendu, ces nouvelles unités correspondront généralement à celles d'un certain modèle linguistique (syllabes, mots, syntagmes, propositions, paragraphes, etc.), mais notre modèle documentaire ne fait aucune hypothèse quant à leur nature effective : tout ensemble d'unités primitives peut former une unité secondaire.

Les unités secondaires sont elles-mêmes regroupées en « couches » indépendantes, plusieurs marquages pouvant cohabiter sans contrainte particulière. Différentes unités secondaires pourront également se chevaucher, et restent indépendantes de la structure logique propre au document traité. En somme, les différents marquages d'un même document peuvent donc être considérés comme appartenant à des couches distinctes et indépendantes, reposant sur une couche d'arrière-plan constituée par la structure logique du document.

Le modèle permet bien sûr d'associer des informations aux unités secondaires, celles-ci étant uniformément représentées sous forme de structures de traits. Il s'agit d'un modèle de données simple mais souple, très communément utilisé en linguistique formelle et en traitement automatique des langues. Plus expressif que le modèle attributs/valeurs de GATE, il est également utilisé dans UIMA. Il a pour avantages principaux d'autoriser la représentation d'informations relativement complexes, éventuellement récursives, et de se prêter facilement à l'expression de contraintes *via* des mécanismes tels que l'unification. Toutes les annotations produites par les composants d'une chaîne de traitement sont ainsi représentées sous cette forme, et inversement, tous les formalismes proposés par la plate-forme permettent d'exploiter ces structures.

En plus des marquages et des annotations que nous venons d'évoquer, le modèle d'annotation prévoit la représentation de *relations* entre les unités marquées. Là encore, aucune hypothèse n'est faite quant à la nature de ces relations (ni des unités mises en relation), qui pourront aussi bien représenter des liens d'ordre syntaxique, rhétorique ou thématique que, par exemple, des rapports d'alignement. Une relation est constituée d'un ou plusieurs arcs, dont chacun connecte deux unités secondaires. Tout comme les marquages eux-mêmes, les relations sont regroupées en classes, identifiées par un type arbitraire. Il est également possible de leur associer des représentations symboliques sous forme de structures de traits.

Comme toutes les autres données manipulées par la plate-forme, les annotations des documents (marquages, structures de traits et relations) sont représentées en XML, de façon à bénéficier des apports bien connus de ce format standard, notamment en termes d'interopérabilité avec d'autres sources de données et de facilité de manipulation grâce aux nombreux outils qui lui sont associés. Toutefois, le codage XML d'annotations conformes au modèle abstrait que nous venons de décrire implique certains choix de conception, que nous ne décrirons pas ici mais qui sont détaillés dans (Bilhaut, 2006). Nous retiendrons ici seulement deux éléments principaux.

Tout d'abord, parmi les différentes approches envisageables pour procéder à l'annotation de documents, au sein desquelles on distingue généralement les approches « déportées » (ou « *stand-off* ») et « embarquées » (ou « *inline* »), nous avons adopté une approche hybride, qui conduit à la modification du document annoté par insertion de balises autour des unités annotées (*ancrage*), tout en stockant les informations associées (caractérisations et relations) dans des documents distincts. Cette solution présente l'avantage de résister aux modifications du document tout en insérant le minimum d'informations dans le document lui-même, et en séparant clairement les marquages des informations associées. En outre, les problèmes de chevauchement et de concurrence entre annotations, qui orientent traditionnellement vers les approches *stand-off*, sont ici pris en charge par ce modèle hybride, grâce notamment à l'indépendance entre les « couches » d'annotation et à l'utilisation de marques XML atomiques pour la délimitation des unités.

Le second élément concerne le respect de la forme initiale du document traité : la plate-forme est capable d'annoter tout type de document XML, indépendamment de son schéma, et ce en préservant totalement son intégrité. Ainsi, au fil des modifications induites par les différentes étapes d'une chaîne de traitement, aucune information du document original ne sera perdue, et les informations ajoutées par la plate-forme seront transparentes pour les applications qui ne sont pas expressément conçues pour les exploiter. Ce procédé permet notamment de préserver la mise en forme initiale du document, et d'observer ainsi les annotations produites *in situ*.

Mentionnons enfin que, pour faciliter le développement de nouveaux modules de traitement conformes à ce modèle, la plate-forme propose une interface de programmation (API), également décrite dans (Bilhaut, 2006), qui masque toute la complexité inhérente aux modalités de marquage XML, en ne laissant apparaître que les concepts du modèle abstrait, soit, principalement, les unités secondaires, les relations, et les représentations symboliques associées.

2.4. *Modèles d'analyse*

2.4.1. *Principes*

Par *modèle d'analyse*, nous désignons un formalisme de représentation de règles linguistiques, auquel on peut associer au moins un algorithme de projection sur corpus, c'est-à-dire de détection et d'annotation des objets vérifiant les modèles spécifiés. Pour l'analyse d'un objet linguistique particulier, on exprime un certain nombre de contraintes devant être satisfaites par des structures textuelles pour qu'elles puissent être regardées comme instances de la classe d'objets décrite. Le choix de ces outils formels, face à la diversité des modélisations de problèmes particuliers, conduit à se poser la question de l'expressivité des modèles d'analyse retenus et de l'éventail nécessaire à la prise en compte de la variété des objets linguistiques. Sur ce point, force est de constater que la portée d'un même modèle d'analyse se restreint à des classes de problèmes souvent bien délimitées. Par exemple, si le modèle des automates s'avère

adapté pour la description de phénomènes de niveau morphologique ou syntagmatique, il est en revanche difficile à manipuler à d'autres niveaux, tels que le niveau discursif.

Il nous semble donc pertinent de prendre en compte et d'exploiter la *complémentarité des modèles d'analyse*, en renonçant à privilégier un hypothétique modèle « omnipotent » capable d'exprimer efficacement tout type de contrainte sur tout type d'objet linguistique. Nous faisons en effet l'hypothèse qu'un traitement complexe doit adopter successivement plusieurs regards sur le même matériau linguistique, auxquels répondront des formalismes distincts. La décomposition d'une tâche complexe en problèmes plus élémentaires autorise la sélection, pour un sous-problème donné, d'un modèle d'analyse adapté à sa résolution. On pourra ainsi combiner, au sein d'un même traitement, des expressions régulières au niveau morphologique, une grammaire locale au niveau syntagmatique, un automate au niveau de la proposition et une grammaire de contraintes au niveau discursif.

Il devient alors nécessaire d'évaluer la « compatibilité » entre chaque modèle d'analyse et chaque classe de problèmes, en identifiant celles pour lesquelles il offre un environnement de description et de résolution adapté. Si cette problématique ouvre un champ de recherche immense, qu'il est exclu d'explorer ici, nous pouvons cependant en illustrer le principe par deux exemples. Pour la description de phénomènes locaux répondant à des motifs assez figés, comportant peu d'éléments optionnels, peu d'alternatives de construction, éventuellement récursifs, l'utilisation de grammaires d'unification s'avère, par exemple, tout à fait adaptée. Dans ce cadre, la compatibilité entre l'objet linguistique et le modèle d'analyse utilisable pour rendre compte de sa structure tient principalement au fait que celle-ci repose sur un enchaînement séquentiel d'éléments adjacents (nous parlerons alors de *séquentialité*) dont l'ordre (nous parlerons alors de *linéarité*) est également évidemment imposé. Si nous nous tournons à présent vers des structures textuelles macroscopiques de niveau discursif, il apparaît qu'elles s'articulent souvent autour de faisceaux d'indices de natures variées, dont l'enchaînement séquentiel et même l'ordre ne constituent pas toujours des informations pertinentes. Pour la description de tels objets, dans une perspective à la fois non séquentielle et non linéaire, le recours à une modélisation par contraintes constitue souvent un bon choix. Si certains modèles d'analyse s'avèrent ainsi plus adaptés à certains types de problèmes, il est cependant essentiel de ne pas les y restreindre. Ainsi, par exemple, l'utilisation d'une grammaire de contraintes non séquentielle et non linéaire, indispensable au niveau discursif, pourra également accroître la robustesse d'un analyseur d'objets locaux (par exemple syntaxiques) en permettant notamment de relâcher certaines contraintes d'ordre.

Il apparaît par ailleurs que le choix des modèles d'analyse mis à disposition doit être guidé, non pas seulement par leur variété, en termes de paradigme, mais également par la capacité de chacun d'eux à s'exprimer sous une forme adaptée à la démarche scientifique et expérimentale. De ce point de vue, il est tout d'abord essentiel de garantir l'*expressivité* du formalisme de représentation des règles d'analyse. Il importe en effet que les contraintes que l'on souhaite exprimer soient pour ainsi dire repré-

sentées telles quelles, et non intégrées à l'appareil procédural informatique qui pourra les prendre en charge. Ce point constitue un élément décisif pouvant seul réellement garantir la capitalisation du savoir linguistique mis en œuvre, sous une forme expressive, modifiable et échangeable, nécessaire à la démarche scientifique. À la *black box* entremêlant de manière opaque le modèle linguistique utilisé et l'algorithme utilisé pour projeter les règles, il convient de substituer dès que possible une *glass box* laissant apparaître ce modèle sous sa forme épurée. D'autre part, il est essentiel de tenir compte du fait qu'un même modèle linguistique pourra souvent être projeté sur corpus de multiples manières. Pour un jeu de règles et contraintes laissé inchangé, on devra dès lors pouvoir modifier l'algorithme d'application de ces règles et contraintes au texte, et changer d'appareil procédural. Aussi, nous devrons toujours, autant que possible, privilégier les *formalismes déclaratifs* et définir des moyens d'exprimer des règles et des contraintes d'une manière à la fois fidèle au paradigme considéré (grammaires de contraintes, transducteur...) et indépendante de l'algorithme de projection ou de résolution.

2.4.2. *Mise en œuvre*

Fidèle à ces observations, LinguaStream permet effectivement d'articuler des traitements réalisés à l'aide de différents formalismes adaptés à différentes tâches, mais partageant un modèle d'annotation uniforme garantissant leur interopérabilité. Les principaux modèles d'analyse disponibles dans la plate-forme sont les suivants :

– un système appelé EDCG (pour *Extended-DCG*), permettant d'écrire des grammaires locales d'unification en se basant sur la syntaxe DCG (*Definite Clause Grammars*) de Prolog et la notation GULP (Covington, 1994). Une telle grammaire peut être décrite dans le plus pur style déclaratif, bien que les spécificités du langage logique restent accessibles aux utilisateurs expérimentés ;

– un système, nommé MRE (pour *Macro-Regular-Expressions*), permettant de décrire des patrons par automates, s'appliquant aussi bien aux formes de surface qu'aux annotations préalablement calculées. Ce modèle d'analyse est assez similaire à celui d'outils tels que Intex ou Unitex, mais il bénéficie ici de la notion de perspective d'analyse, que nous décrirons plus loin, et qui lui permet de porter de manière transparente sur toute unité textuelle préalablement analysée ;

– un langage d'expression de contraintes au niveau discursif. Le formalisme CDML (pour *Constraint-based Discourse Modeling Language*) (Widlöcher et Enjalbert, 2007 ; Widlöcher, 2006), que nous évoquerons ci-après, permet la modélisation par contraintes de structures textuelles (en particulier au niveau discours), sur la base de critères potentiellement non séquentiels (pouvant être distants) et non linéaires (dont l'ordre peut être ignoré). Les contraintes sont exprimées à l'aide d'un ensemble de fonctions primitives (présence/absence, position...), et peuvent porter sur les annotations produites en amont et sur des relations entre ces dernières ;

– un système basé sur un moteur d'inférences de type CLIPS, permettant une modélisation par des règles agissant sur une base de connaissances qui reflète, sous la forme de « faits », le corpus d'entrée tel qu'observé dans une perspective d'analyse

donnée (*cf. infra*) ;

– autres : un système d'annotation à partir de lexiques sémantiques, un outil de « tokenisation » basé sur des expressions régulières (au niveau caractère), un système permettant de délimiter des objets linguistiques à partir du balisage XML du document, etc.[7]

Notons qu'à notre connaissance, aucune plate-forme de TAL n'offre une telle variété de modèles d'analyse, à la fois interopérables et reposant sur des formalismes purement déclaratifs. Même si des architectures comme GATE, UIMA ou OpenNLP pourraient en principe héberger des modèles d'analyse variés, il ne semble pas exister à l'heure actuelle d'ensemble cohérent et comparable à celui que nous venons de présenter. D'autre part, le degré de déclarativité des formalismes proposés est parfois discutable : on notera par exemple que le formalisme JAPE mis en œuvre par GATE est moins éloigné d'un langage de programmation que d'un formalisme de représentation de modèles linguistiques opérationnels. Les outils qui s'appuient sur un formalisme réellement déclaratif (comme ContextO ou Unitex), quant à eux, sont justement dédiés à un modèle d'analyse particulier et ne permettent donc pas, en tant que tels, d'exploiter la complémentarité que nous défendons ici, sans être intégrés dans une plate-forme plus générale.

Soulignons enfin que le recours aux modèles d'analyse proposés n'est imposé d'aucune manière et qu'il est tout à fait possible d'en proposer de nouveaux, ou d'intégrer des systèmes existants[8], en leur donnant accès au modèle commun d'annotation.

2.5. *Variabilité des perspectives sur le texte*

2.5.1. *Principes*

Nous avons vu plus haut que le principe d'enrichissement incrémental du texte invite à considérer ce dernier comme un assemblage toujours inachevé d'objets résultant d'analyses et d'interprétations conduites sur différents plans. Il en résulte que chaque analyse se rapporte nécessairement au texte à la lumière d'une *perspective* particulière par laquelle elle accorde plus ou moins d'intérêt à telle ou telle information dont le texte s'est vu enrichi. Nous désignerons ici par *perspective d'analyse* la manière dont une analyse se rapporte à la donnée textuelle, manière dont elle la rend pour ainsi dire « observable ».

Cette notion nous paraît extrêmement importante dans la mesure où elle conditionne très fortement le processus de projection d'un modèle opérationnel sur corpus. Or, elle n'est que rarement rendue visible, ce qui peut masquer des propriétés importantes des procédés d'analyse, et limite grandement leur reproductibilité. Il nous paraît au contraire indispensable de chercher à la rendre aussi *explicite* que possible,

7. Si ces derniers ne sont pas tout à fait comparables aux précédents, ils n'en demeurent pas moins conformes à la définition des modèles d'analyse donnée précédemment.

8. L'idée d'établir un pont vers Unitex a par exemple été évoquée.

de façon à augmenter la transparence des modèles d'analyse. Il semble en outre extrêmement utile de la rendre *modulable*, de façon à pouvoir définir, pour chaque procédé d'analyse, une perspective adéquate.

Nous avons en particulier identifié trois propriétés que nous considérons comme des caractéristiques essentielles d'une perspective sur le texte pouvant être adoptée par un procédé de TAL quelconque :

– l'adoption d'une perspective particulière se traduit en premier lieu par la sélection de certains objets particuliers, dotés de certaines propriétés pouvant être de nature formelle ou symbolique, afin d'exprimer des modèles portant exclusivement sur les éléments pertinents dans un cadre théorique donné. À l'inverse, on pourra préférer exclure de l'analyse certains types d'éléments désignés, tels, par exemple, que parenthèses ou digressions, au contenu considéré comme « parasite ». Cette sélection consiste, en d'autres termes, en un *filtrage* du texte ;

– d'autre part, la perspective que l'on peut adopter sur un texte se traduit par le choix des environnements, ou unités macroscopiques, au sein desquels les structures décrites doivent être recherchées (fût-ce le texte dans son ensemble). On pourra par exemple s'intéresser à des phénomènes linguistiques se produisant exclusivement *au sein de* titres, ou *au sein de* paragraphes possédant une propriété particulière. La définition d'un tel *domaine d'analyse* peut également consister à fixer une limite maximale au-delà de laquelle les motifs décrits n'auront pas à être recherchés ;

– enfin, de nombreux formalismes imposent la définition d'un grain d'analyse minimal, c'est-à-dire l'existence d'une *unité textuelle minimale* à laquelle s'appliquent les patrons décrivant les structures. Le choix de cette unité participe clairement à la détermination d'un point de vue sur le texte. Or, si le caractère ou le mot est souvent considéré comme unité minimale « naturelle », il est extrêmement utile de ne pas s'y limiter, en permettant à une analyse de spécifier les *atomes* sur lesquels elle s'appuiera. Toute unité textuelle disponible devrait pouvoir jouer ce rôle. Remarquons que l'adoption implicite d'un grain minimal tel que le mot augmente considérablement la complexité des descriptions d'objets de haut niveau de composition.

La définition explicite de la perspective que l'on adopte permet d'approcher autant que possible l'expression d'un modèle opératoire dans un formalisme donné, du modèle théorique lui-même, en formulant distinctement la perspective sous-jacente à ce modèle.

Au-delà de cette conformité, la variabilité des perspectives permet d'accroître la portée des différents modèles d'analyse. En effet, chaque modèle n'étant plus, dès lors, inféodé à une vision particulière du texte, la classe de problèmes qu'il permet d'appréhender s'étend significativement.

Le notion de perspective renvoie également à la possibilité plus générale de disposer d'une *représentation abstraite* du texte et des annotations. Nous avons dit, en effet, que chaque palier de l'analyse peut accéder simultanément et de manière homogène au texte original et aux annotations produites par tous les paliers antérieurs. Cependant, les analyses de haut niveau, en termes d'échelle et/ou d'interprétation, re-

quièrent souvent principalement l'exploitation d'informations résultant elles-mêmes d'analyses préalables, la forme concrète des éléments analysés pouvant pour sa part souvent être ignorée, de même que le résultat d'analyses intermédiaires éventuelles. Plus généralement, l'idée d'abstraction renvoie à la possibilité d'observer un élément *en tant qu'*instance d'une classe particulière, indépendamment de la manière dont sa présence a été révélée. Cette *abstraction des formes de surface et des analyses intermédiaires*, au profit du résultat de leur interprétation, offre l'avantage de ramener une diversité difficile à appréhender exhaustivement à un ensemble de représentations respectant un modèle formel fixe, contrôlables, sur lesquelles différentes opérations pourront être effectuées.

2.5.2. *Mise en œuvre*

LinguaStream met en œuvre ce principe de « perspective d'analyse » de façon concrète et explicite, et tous les modèles d'analyse décrits dans la section précédente en bénéficient. Cette fonctionnalité constitue une des originalités importantes de la plate-forme par comparaison aux logiciels analogues, qui se révèle extrêmement utile dans de nombreuses situations. Elle se matérialise par un certain nombre de paramètres qui permettent de spécifier comment les marquages déjà présents dans le document à traiter doivent être interprétés : classes d'unités secondaires à considérer comme jetons ; classes d'unités définissant le domaine d'analyse ; filtres à appliquer sur les annotations antérieures.

La possibilité de définir *localement* quelles classes d'unités doivent être considérées comme jetons garantit la variabilité du grain d'analyse au cours du traitement. Elle concerne les nombreux modèles qui imposent la définition d'un grain minimal, dit jeton ou *token*. C'est par exemple le cas de toute grammaire ou transducteur : ces formalismes supposent l'existence d'une unité textuelle (comme le caractère ou le mot) à laquelle s'appliquent les patrons. Quand la définition de ce grain minimal est nécessaire au fonctionnement d'un composant, la plate-forme permet de spécifier localement le ou les types d'unités à considérer comme jetons. Toute unité préalablement délimitée peut jouer ce rôle : il pourra s'agir du découpage habituel en mots, ou de tout autre type d'unité : syntagmes, phrases, segments discursifs, etc. Le grain minimal peut donc être différent pour chaque palier d'une chaîne, ce qui augmente considérablement la portée des différents modèles d'analyse disponibles dans la plate-forme. Il sera par exemple possible d'écrire une grammaire de texte à l'aide de grammaires d'unification (formalisme EDCG), pourtant traditionnellement réservées à la description des motifs assez locaux.

Ce principe est mis en œuvre de la manière suivante. Au niveau du modèle documentaire, il n'existe pas de jeton dans le sens où toutes les annotations sont équivalentes. Par défaut, un marquage sera considéré comme une séquence d'objets textuels composée du jalon de départ, puis des différentes unités contenues (texte ou autres marquages), et enfin du jalon de fin. En revanche, lorsqu'une unité appartient à une classe localement considérée comme « jeton », elle ne constitue plus une séquence mais une unité atomique au contenu purement textuel.

Considérons, par exemple, un marquage délimitant les mots et un autre délimitant les phrases. Dans ce cas, on sera généralement amené à considérer les unités de type « mot » comme jetons et les unités de type « phrase » comme succession de jalons. Dans ce cas, le document sera perçu comme une séquence d'objets du type :

$$\ldots \quad \textit{début}_{phrase} \quad \textit{jeton}_{mot} \quad \textit{jeton}_{mot} \quad \ldots \quad \textit{jeton}_{mot} \quad \textit{fin}_{phrase} \quad \ldots$$

Mais il est également possible de spécifier le type « phrase » comme désignant des jetons. Dans ce cas, les phrases seront considérées comme des unités atomiques, et le marquage des mots (du moins ceux qui sont effectivement compris dans un marquage de type « phrase ») ne sera plus accessible :

$$\ldots \quad \textit{jeton}_{phrase} \quad \textit{jeton}_{phrase} \quad \textit{jeton}_{phrase} \quad \ldots$$

La seconde composante d'une perspective d'analyse concerne le filtrage. Précisons que, là encore, il n'est question que du point de vue adopté par un module sur le texte : les annotations filtrées ne sont en aucun cas supprimées du document, mais seulement masquées à un certain point de la chaîne de traitement.

La troisième et dernière composante d'une perspective d'analyse permet de restreindre le champ d'application des règles à certaines classes d'unités, champ qualifié plus haut de « domaine d'analyse ». Cela peut tout d'abord être utile pour éviter que les unités reconnues ne chevauchent d'autres unités déjà marquées (par exemple, des syntagmes ne sauraient chevaucher des phrases). Ou, plus simplement, on pourra souhaiter éviter le marquage de certaines zones du document qui n'ont pas lieu d'être analysées (par exemple les zones de métadonnées). Mais cela est également important du point de vue des performances : quand l'algorithme du modèle d'analyse employé n'est pas déterministe, on s'efforcera de limiter le domaine d'analyse au grain le plus fin possible de façon à éviter des calculs inutiles.

2.6. *Cycle d'expérimentation*

2.6.1. *Principes*

La démarche expérimentale impose, par ailleurs, la possibilité de procéder de manière itérative en alternant phases de modélisation, d'édition de règles, de projection sur corpus, d'évaluation, d'ajustement du modèle, etc.

La nécessité d'opérer des allers-retours entre édition et applications des règles devra en particulier tirer bénéfice du privilège accordé aux représentations déclaratives qui permettent de minimiser la distance entre un modèle et son opérationalisation.

Une place importante doit par ailleurs être accordée à l'évaluation du résultat de la projection d'un modèle sur corpus. En particulier, la visualisation de ce résultat occupe une place importante. Sur ce point, il est indispensable de tenir compte de la variété des objets (unités, relations, variabilité du grain...) et de multiplier les vues

possibles : corpus annoté, vue de type concordancier, visualisation des relations, accès aux représentations symboliques...

De même, l'évaluation suppose souvent l'accès à différentes méthodes de comptage dont les résultats doivent également pouvoir être appréhendés de manières variées, rendant explicites les propriétés importantes des résultats obtenus.

Ajoutons enfin que la démarche d'évaluation suppose souvent la confrontation du résultat d'une analyse computationnelle avec une annotation de référence, cette confrontation devant elle-même être rendue observable par différents artéfacts.

2.6.2. *Mise en œuvre*

La plate-forme offre de multiples outils permettant de procéder à la visualisation et à l'évaluation des résultats produits par une chaîne de traitement, reposant dès que possible sur les standards XML et les outils associés.

Le mode de visualisation le plus communément utilisé consiste à rendre visibles les marquages et annotations au sein du document lui-même. Un éditeur permet de déterminer les modalités d'affichage de chaque marquage, sous la forme d'un « descripteur de vue ». Au sein du document « décoré », un clic sur un segment annoté permet de faire apparaître la structure de traits associée. Il est également possible de masquer l'ensemble des annotations produites par LinguaStream de façon à visualiser le document dans son état initial. Outre la visualisation au sein même du document, la plate-forme propose d'autres modes qui permettent, cette fois sans préserver la forme initiale du document, de disposer d'une vue spécifique. Un premier mode de visualisation reprend la vue classique de type « concordancier », qui permet d'observer les contextes gauche et droit d'un ou plusieurs types d'expressions analysées. Un autre mode produit une vue dite « macroconcordancier », qui offre une vision sélective adaptée à des objets de grain plus important. Elle peut, par exemple, être utile pour visualiser des objets de type cadre de discours, que l'on souhaite pouvoir observer isolément, tout en bénéficiant d'une vue synthétique sur leur contenu et/ou leur contexte. Un autre mode de visualisation permet l'observation des relations résultant de l'analyse, et notamment l'observation des relations portant sur des objets de granularité élevée, ainsi que l'accès aux annotations associées à ces relations.

Outre ces vues « textuelles », la plate-forme permet également de générer différents types de graphiques relatifs aux annotations présentes dans un document. Une interface spécifique permet de créer aisément divers types de graphiques à partir d'un document annoté : proportions relatives de marquages possédant certaines propriétés (par filtrage sur les structures de traits), quantités de marquages par segment (le grain étant paramétrable), graphe représentant une valeur numérique présente sur un trait donné, etc.

Parmi les outils susceptibles d'intervenir dans la création de systèmes expérimentaux, il convient également de mentionner ceux qui permettent de manipuler les chaînes de traitement elles-mêmes. Un module permet ainsi d'appliquer une chaîne de traitement à un jeu de documents, et de collecter les résultats d'analyse obtenus.

Un autre système permet quant à lui d'automatiser différentes tâches réalisables par la plate-forme sous la forme de scripts. Un tel script pourra, par exemple, lancer une requête sur un moteur de recherche en fonction de mots-clefs demandés à l'utilisateur, puis appliquer une chaîne de traitement à chacun des documents retournés.

3. Exemples d'applications

Envisageons à présent trois applications mises en œuvre à l'aide de la plate-forme. Nous chercherons principalement, dans cet exposé, à mettre en lumière quelques-uns des principes et dispositifs présentés dans les sections précédentes.

3.1. *Cadres de discours*

Notre premier exemple se situe dans la perspective générale de l'étude sur corpus de phénomènes linguistiques, en l'occurrence le phénomène de l'encadrement du discours, au sens de (Charolles, 1997). Rappelons les principes généraux de la théorie de Charolles. Le terme de cadre de discours désigne, selon cet auteur, des segments textuels homogènes du point de vue d'un critère d'interprétation fixé par une expression en position détachée en début de phrase, dite introducteur de cadre. La figure 3 montre un exemple de cadre (tel qu'annoté par la plate-forme par le traitement indiqué ci-après) : les éléments présents dans le cadre (délimité par des pointillés) doivent, pour être interprétés, être rapportés au critère sémantique fourni par l'expression temporelle en position d'introducteur : « Depuis le milieu des années 1980 ». On parlera ici de cadre temporel[9], type auquel nous nous limiterons dans cet exposé.

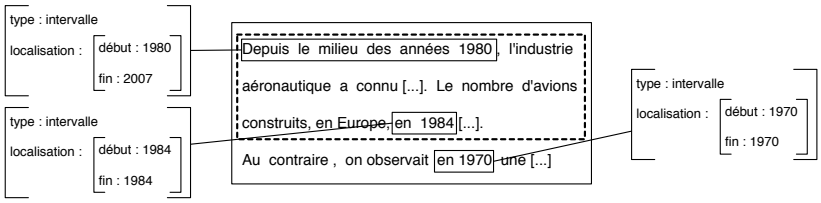

Figure 3. *Exemple de cadre de discours temporel*

L'opérationalisation en TAL de ce modèle psycholinguistique impose la résolution de deux problèmes principaux : d'une part la détection des introducteurs, et d'autre part l'évaluation de leur portée, c'est-à-dire la détermination de la borne droite du cadre introduit[10].

9. D'autres types de cadres, par exemple spatiaux (*cf. infra*), sont décrits dans la littérature.
10. Pour des détails concernant le modèle linguistique et l'expérimentation sur corpus on pourra se reporter à (Bilhaut *et al.*, 2003) et (Ferrari *et al.*, 2005).

Le problème de la détection des introducteurs (ici temporels) se décline lui-même en deux sous-problèmes : l'analyse des expressions temporelles, et celle des introducteurs s'appuyant sur elles. L'analyse sémantique des expressions temporelles fait l'objet d'une grammaire EDCG implémentant une grammaire sémantique (Allen, 1995) selon une méthodologie bien établie. Cette grammaire, précédée d'un étiquetage morphosyntaxique (par le module Tree-Tagger en l'occurrence), reconnaît les expressions pertinentes (syntagmes prépositionnels) et leur associe une représentation sémantique sous forme de structures de traits (*cf.* figure 3).

Sur cette base, la détection des introducteurs peut être mise en place à l'aide de critères principalement positionnels. Ces règles sont exprimables à l'aide du formalisme des « macro expressions régulières » (MRE), comme illustré ci-dessous.

```
{type : phrase, anchor : start}
<introducteur>
{type : connecteur}? {type : temporel} /as $t
</introducteur> /sem {axe : temps, valeur : $t} ","
```

Nous exploitons ici trois marquages préalables : les expressions temporelles, la délimitation des phrases, et la projection d'un lexique de connecteurs de discours (« mais », « cependant », etc.). Les contraintes sur les structures de traits produites en amont, ainsi que sur certaines formes de surface (la virgule en fin de motif) permettent de délimiter l'introducteur. Nous recherchons une zone de texte précédée d'un début de phrase (`{type : phrase, anchor : start}`) composée d'un éventuel connecteur de discours (`{type : connecteur}?`) et d'une expression temporelle (`{type : temporel}`), le tout suivi d'une virgule. La structure de traits associée à l'expression temporelle est mémorisée dans la variable `$t`. Les autres éléments de l'expression sont relatifs à l'annotation produite : seule la partie entre les balises `<introducteur>` et `</introducteur>` (indiquant le type du marquage) sera intégrée comme formant l'introducteur, les autres éléments constituant les contextes gauche et droit attendus. À cette expression est associée une structure de traits (dite « sémantique » et spécifiée sous le symbole `/sem`) mentionnant la nature de l'introducteur (`{axe : temps}`) et la sémantique de l'expression temporelle (variable `$t`).

À quelques détails de forme près, le lecteur aura reconnu un format, relativement classique en TAL (ou en compilation) de reconnaissance/annotation par expressions régulières, mais avec une spécificité notable : le fait que l'expression intègre des éléments *à différents niveaux de grain* : ponctuation, mot, syntagme, et borne de phrase. Cet exemple illustre ainsi clairement l'un des principes majeurs de la plate-forme : non seulement des structures de différentes granularités peuvent coexister, mais il est également possible de les exploiter au sein d'un même modèle d'analyse, et au cours d'une même phase de traitement.

La détermination de la portée de l'introducteur s'avère beaucoup plus complexe dans la mesure où les critères formels de clôture des cadres sont difficiles à établir linguistiquement. Un certain nombre d'indices ont toutefois pu être dégagés dans le cas précis des cadres temporels (Bilhaut *et al.*, 2003). La méthode utilisée ici s'appuie

sur des critères énonciatifs tels que la cohésion des temps verbaux et sur des calculs sémantiques de cohérence entre l'introducteur et les autres expressions temporelles. La nature de ces contraintes diffère radicalement des précédentes. D'une part, nous pouvons désormais nous abstraire de la séquentialité du texte : contrairement à une approche par expressions régulières, nous pouvons ici ignorer un certain nombre d'éléments du flot textuel. D'autre part, s'il existe bien des contraintes interprétatives entre l'introducteur et certains éléments de la zone introduite, ces contraintes n'imposent pas un ordre strict entre ces éléments. Pour l'expression de telles contraintes à la fois non linéaires et non séquentielles, nous disposons du formalisme CDML (Widlöcher et Enjalbert, 2007 ; Widlöcher, 2006), et pouvons formuler la « grammaire » suivante.

```
Unit cadre {type :"cadre"} :
@mru :['paragraphe']
@tokens :['parenthèse','apposition']
        start(pattern : {type :"introducteur"})
        homogeneity(comparator : portée)

Comparator portée ({type :"verbe"} as $v1, {type :"verbe"} as $v2) :
        $v1/temps = $v2/temps

Comparator portée ({type :"introducteur"} as $i, {type :"tempo"} as $t) :
        (($i/debut >= $t/debut) and ($i/debut <= $t/fin))
        or
        (($i/debut <= $t/debut) and ($i/fin >= $t/debut))
```

Nous recherchons ici une unité textuelle commençant par un élément identifié comme « introducteur » dont tous le verbes sont au même temps, et au sein de laquelle les expressions temporelles portent sur un intervalle compatible avec l'introducteur. Nous précisons par ailleurs que le domaine d'analyse est le paragraphe (paramètre `@mru` pour *Maximal Relevant Unit*). Si aucun autre critère de fermeture n'est rencontré, le cadre se prolongera ainsi, au plus, jusqu'à le fin du paragraphe. D'autre part, les parenthèses et les appositions sont rendues atomiques (paramètre `@tokens`) : les éventuels verbes et expressions temporelles contenus, qui pourraient parasiter le test de compatibilité sémantique, seront ainsi simplement ignorés.

Finalement, la figure 4 illustre les différentes étapes de l'analyse des cadres temporels, en précisant, pour chacune d'elles, le « grain » auquel elle opère et le modèle d'analyse utilisé. Il est ainsi possible, à l'aide des principes méthodologiques promus par la plate-forme, et en nous appuyant sur la complémentarité des modèles d'analyse, de mettre en place un analyseur de cadres temporels, certes encore imparfait, mais ne faisant usage que de formalismes purement déclaratifs propices à la capitalisation de l'expertise linguistique.

3.2. *Une application à la recherche d'information géographique : le moteur de recherche GeoSem*

L'information documentaire possède souvent une « dimension géographique ». Les entités mentionnées (fleuves, villes...), les faits relatés (événements politiques...)

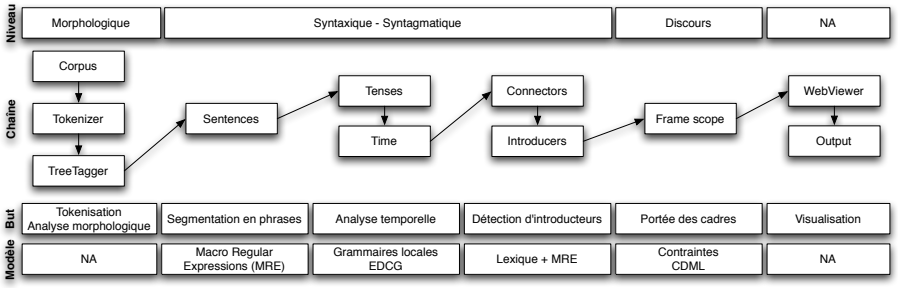

Figure 4. *La chaîne d'analyse des cadres temporels*

ou les observations décrites (de nature économique, par exemple) y sont alors liés, d'une manière ou d'une autre, à une localisation dans un *espace géographique*. La notion de *recherche d'information géographiquement informée* (*geographically-aware*) (Vaid *et al.*, 2005) en découle comme domaine de recherche spécifique consacré à cette dimension. De plus, le « discours géographique » possède souvent une dimension chronologique, si bien que la localisation *spatiale* y croise la localisation temporelle, autre problématique bien identifiée en TAL (Setzer, 2001). L'extrait suivant[11], illustre ce mode spécifique de structuration et de « référencement » de l'information.

> De 1965 à 1985, le nombre de collégiens et de lycéens a augmenté de 70 %, mais selon des rythmes et avec des intensités différents selon les académies et les départements. Faible **dans le Sud-Ouest et le Massif central**, modérée **en Bretagne et à Paris**, l'augmentation a été considérable **dans le Centre-Ouest, en Alsace, dans la région Rhône-Alpes et dans les départements de la grande banlieue parisienne** où les effectifs ont souvent plus que doublé

Les expériences et réalisations mentionnées ici, menées dans le cadre du projet GeoSem (Enjalbert et Gaio, 2006) et présentées dans (Bilhaut *et al.*, 2007), s'inscrivent dans ce double contexte de recherche d'informations spatialement et temporellement qualifiées. Le contexte applicatif de GeoSem est celui de « l'Intelligence Territoriale »[12]. Il s'agit d'offrir à l'utilisateur des outils lui permettant d'analyser des situations locales, d'effectuer les diagnostics appropriés et finalement de prendre des décisions pertinentes concernant les politiques d'aménagement du territoire. Le corpus, composé de textes expositifs présentant des analyses socio-économiques dans

11. Issu de Hérin, R., Rouault, R., Veshambre, V. *Atlas de la France scolaire. De la maternelle au lycée*, Collection Dynamiques du territoire, Reclus, 1994.

12. Des développements inspirés par le même projet initial, concernant la valorisation d'un patrimoine régional, sont menés à Pau au LIUPPA (Marquesuzaà *et al.*, 2005), avec une implémentation utilisant également LinguaStream.

lesquelles la variabilité spatiale et temporelle des phénomènes observés est un élément essentiel, comprend une forte proportion de textes longs ou très longs (rapports de plusieurs dizaines de pages), qui nous oriente vers une problématique d'*extraction de passages* plutôt que de recherche documentaire au sens traditionnel.

Dans ce cadre, une requête naturelle portera sur un triple critère *Espace - Temps - Phénomène* : « Quelles informations puis-je avoir sur tel phénomène socio-économique, dans tel espace et dans telle période ? », et nous voulons retourner à l'utilisateur des *passages* des documents, afin de permettre une *navigation* intra-documentaire.

Développé dans ce contexte, le moteur de recherche GeoSem (Bilhaut *et al.*, 2007) suit les principes usuels en matière de RI, distinguant une phase d'indexation « *off-line* » des ressources documentaires, et une phase d'interrogation « *on-line* » par un utilisateur, grâce à une interface appropriée. Mais il intègre aussi un certain nombre de spécificités :

– indexation multidimensionnelle : les dimensions spatiales et temporelles ne peuvent être mises sur le même plan que les informations textuelles « ordinaires ». Les documents reçoivent donc une indexation selon trois *dimensions* ou *axes sémantiques* : Espace - Temps - Phénomène ;

– indexation sémantique : une analyse sémantique est réalisée et produit une annotation des expressions de localisation temporelle et spatiale. Pour exploiter ces annotations et les comparer à la requête d'un utilisateur, des *comparateurs sémantiques* spécifiques à chaque axe sont définis ;

– indexation de passages : nous n'indexons pas les documents mais des passages de textes, identifiés par des *analyseurs discursifs*, intégrant une analyse de cadres de discours (*cf. supra*) et une méthode « de type $tf \cdot idf$ ».

La figure 5 présente l'interface d'interrogation avec les trois champs de saisie correspondant aux trois dimensions sémantiques de l'indexation, et le retour de passages sélectionnés. La requête de l'utilisateur, formulée en langue naturelle, fait l'objet d'une analyse linguistique, par l'analyseur qui traite également les ressources documentaires. En outre l'interface utilisateur inclut un dispositif de visualisation sur une carte des localisations portées par le texte et la requête.

La mise en place du moteur de recherche repose fortement sur la plate-forme LinguaStream, avec laquelle ont été développés les analyseurs prenant en charge à la fois l'analyse *off-line* du corpus et l'analyse *on-line* des requêtes. La mise en place de ces analyseurs tire avantage de différents principes évoqués ci-dessus. Les différentes analyses impliquées imposent tout d'abord, comme dans l'exemple précédent, des traitements multiéchelles simplifiés par la plate-forme : analyses syntagmatiques temporelles et spatiales, délimitation de cadres de discours, analyse thématique... La complémentarité des modèles d'analyse est également exploitée ici. Aux modèles d'analyse MRE et EDCG s'ajoute ici une analyse distributionnelle de type $tf \cdot idf$, disponible sous la forme d'un module spécifique. Les mécanismes d'abstraction des formes de surface et des objets intermédiaires interviennent également, de manière cruciale pour

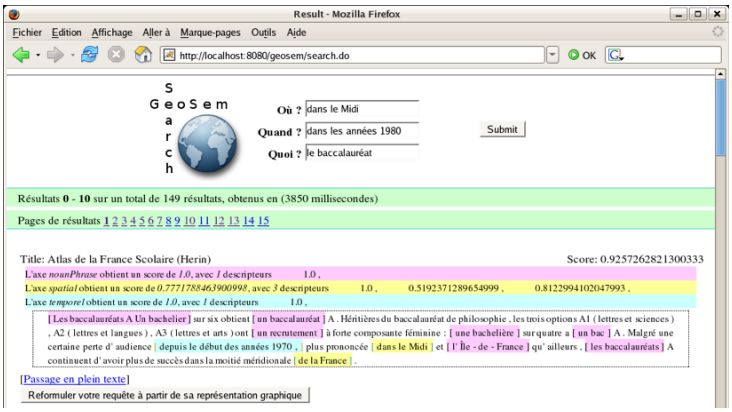

Figure 5. *Interface du moteur de recherche avec affichage d'un passage résultat*

maîtriser une telle suite de traitements. De même, l'approche par composant prend ici tout son sens. Remarquons en particulier que la chaîne des cadres temporels présentée dans la section précédente peut être intégrée ici en tant que macrocomposant. On trouvera en outre dans cette application une bonne illustration des bénéfices tirés de l'uniformité du modèle d'annotation, la phase de production de l'index (sur la base des analyses sémantiques réalisées) étant en effet tout à fait indifférente à la manière dont les informations pertinentes ont été extraites. Il en résulte, fait notable, que l'architecture du moteur de recherche est en elle-même parfaitement générique et n'est aucunement inféodée à l'information géographique, précisément parce qu'elle s'appuie sur un modèle d'annotation « universel ». Pour adapter le moteur à d'autres axes sémantiques, il suffira en effet de l'alimenter avec d'autres analyseurs.

3.3. *Fouille de texte*

Nous terminerons par une application développée dans le cadre du DÉfi Fouille de Textes (DEFT) 2006[13] et présentée en détail dans (Widlöcher *et al.*, 2006). La tâche proposée concernait la segmentation thématique de textes. Trois corpus distincts étaient proposés, avec, pour chacun, une notion spécifique d'« unité thématique » : un corpus politique dont les unités étaient de l'ordre du paragraphe ; un corpus juridique, recueil de lois de l'Union européenne agrégées sans marques de séparation, dont chaque loi devait constituer une unité ; un ouvrage scientifique, dont il fallait retrouver la structure logique, les titres de chapitre et section ayant été retirés. La méthode que nous avons souhaité expérimenter et que nous avons mise en œuvre combine des méthodes de TAL « linguistique » et de fouille de données. Plus précisément, elle

13. `http://www.lri.fr/ia/fdt/DEFT06`

associe la détection de marqueurs discursifs génériques à une technique de fouille séquentielle de texte, les différents traitements étant intégrés au sein de LinguaStream.

Du point de vue linguistique, nous nous appuyons sur des indices de cohésion discursive, répartis en différentes catégories (rupture, amorçage, prolongement, clôture, etc.). En découle un repérage de différentes unités telles que connecteurs de discours (« donc », « ensuite », etc.), marques indicatives (« le problème est que »), marqueurs anaphoriques, zones de cohésion lexicale, introducteurs de cadres de discours, temps verbaux, etc. Ces différents marqueurs sont repérés par des chaînes spécifiques de composants, comme dans le cas des applications présentées précédemment.

Une méthode de fouille de données, par classification supervisée, est alors appliquée pour découvrir les configurations d'indices linguistiques caractérisant les phrases marquant une rupture thématique. La méthode repose sur la notion de *règle séquentielle*, constituée d'une prémisse représentant une succession d'attributs et d'une conclusion qui est elle-même aussi un attribut. L'extracteur WinMiner (Méger et Rigotti, 2004) fournit des règles séquentielles caractéristiques du corpus étudié, en ce sens qu'elles y constituent un ensemble de *motifs fréquents*. Ces règles, constituées sur un corpus d'apprentissage, sont ensuite projetées sur un corpus de test. Pour déterminer le type d'une phrase (*rupture* ou non), un classifieur opérant par vote entre les différentes règles extraites a été réalisé.

La mise en œuvre proposée prend la forme d'une chaîne LinguaStream intégrant à la fois des modules linguistiques et des composants « externes » de fouille de données. Une des caractéristiques de l'architecture de la plate-fome est en effet de permettre d'établir aisément ce type de « pont » vers des applications externes. D'autres avantages des principes méthodologiques évoqués ci-dessus se trouvent ici encore mis en lumière. La modularité des chaînes de traitement a rendu possible un développement rapide dans le cadre fortement collaboratif de ce travail, LinguaStream permettant l'assemblage des composants réalisés ou paramétrés par les différentes sous-équipes. Pour la partie linguistique de ce travail, la complémentarité de modèles d'analyse et leur respect du paradigme déclaratif ont permis une expression rapide des contraintes linguistiques retenues. Enfin, insistons sur la capitalisation du savoir et des expériences permise par la plate-forme. L'assemblage des composants au sein d'une chaîne et le paramétrage de chacun constitue une « mémoire » du dispositif expérimental qui garantit la reproductibilité des expériences, ce qui s'avère particulièrement pertinent dans le cadre d'une campagne d'évaluation et de comparaison telle que DEFT.

4. Conclusion

Outre les applications que nous venons de mentionner, la plate-forme LinguaStream est et a été utilisée dans divers projets de recherche du GREYC ou d'autres laboratoires. Ces projets concernent aussi bien des travaux applicatifs (recherche et extraction d'information, veille, résumé...) qu'expérimentaux, en TAL ou en linguistique (extraction de ressources, validation d'hypothèses sur corpus, constitution d'ob-

servables, analyses distributionnelles...). Dans chacun de ces cas, elle s'avère précieuse pour la mise au point rapide de dispositifs expérimentaux ou de maquettes, en simplifiant significativement le cycle conception - expérimentation - évaluation. Elle est également utilisée depuis plusieurs années dans l'enseignement du TAL. Par l'expérimentation, les étudiants peuvent ainsi se familiariser avec différentes méthodes, en bénéficiant d'un ensemble de traitements linguistiques et documentaires, qui leur permet de concentrer leur effort sur un problème ou une tâche spécifique.

Les performances observées à l'utilisation de LinguaStream sont extrêmement disparates et dépendent fortement des applications. Un certain nombre de choix concernant son architecture générale ont certes des conséquences en termes de performances (comme la sérialisation XML des flux de données, ou le recours par plusieurs modèles d'analyse à des algorithmes non déterministes). Néanmoins, la complexité algorithmique d'une chaîne de traitement dépend quasi exclusivement de celle des modules qui la composent et des règles appliquées. Ainsi ne peut-on pas évaluer en toute généralité les performances de la plate-forme en tant que telle, mais seulement celles de chaque chaîne ou de chaque composant. Nous retrouvons ici un avantage notable de sa flexibilité, qui permet, selon les cas, de sacrifier les performances à l'expressivité des formalismes (utilisation plutôt expérimentale), ou inversement, de privilégier des modèles plus limités pour obtenir de meilleures performances (traitements en masse). Par ailleurs, il est important de noter que l'ensemble de l'architecture a été conçu pour traiter des données de taille arbitraire, sans aucun effet de plafond qui serait lié, par exemple, aux ressources en mémoire. LinguaStream permet donc de traiter de grandes quantités de données, par des procédés potentiellement complexes, pourvu que le temps nécessaire soit disponible.

Parmi les perspectives de développement les plus immédiates, on notera l'intégration à la plate-forme des outils et formalismes du Web sémantique tels que promus par le W3C. Un travail en cours permettra prochainement d'intégrer des ontologies lexicalisées comme ressources, en amont des chaînes de traitement, et de produire, en aval, des graphes sémantiques au format RDF à partir des annotations produites sur corpus. Un autre travail actuellement en cours permettra d'automatiser la mise en place d'applications Web à partir de chaînes de traitement, à des fins pédagogiques et d'expérimentation. Le travail d'intégration d'outils d'apprentissage pour la fouille de textes, engagé à l'occasion des campagnes DEFT depuis 2006 et unissant différentes compétences du GREYC, sera également poursuivi. À plus long terme, le modèle séquentiel pourrait également être reconsidéré, en envisageant d'intégrer les notions de récursivité et de parallélisme au sein même des chaînes, voire même de substituer au modèle séquentiel une approche à base d'agents communiquant par le texte. Enfin, ajoutons que les questions de robustesse, de performance et de passage à l'échelle feront également l'objet d'une attention particulière, notamment dans le cadre d'un projet, en cours, de valorisation industrielle.

5. Bibliographie

Allen J., *Natural Language Understanding*, Benjamin/Cummings Pub. Co., Menlo Park, 1995.

Bilhaut F., Analyse automatique de structures thématiques discursives – Application à la recherche d'information, PhD thesis, Université de Caen, 2006.

Bilhaut F., Dumoncel F., Enjalbert P., Hernandez N., « Indexation sémantique et recherche d'information interactive », *Actes de la 4e conférence en recherche d'information et applications (CORIA)*, p. 65-76, 2007.

Bilhaut F., Ho Dac L.-M., Borillo A., Charnois T., Enjalbert P., Le Draoulec A., Mathet Y., Miguet H., Péry-woodley M.-P., Sarda L., « Indexation discursive pour la navigation intra-documentaire : cadres temporels et spatiaux dans l'information géographique », *Actes de la 10e conférence sur le traitement automatique des langues naturelles (TALN)*, Batz-sur-Mer, France, p. 315-320, 2003.

Bilhaut F., Widlöcher A., « LinguaStream : An Integrated Environment for Computational Linguistics Experimentation », *Proceedings of the 11th Conference of the European Chapter of the Association of Computational Linguistics (EACL'06) (Companion Volume)*, Trento, Italy, p. 95-98, 2006.

Bourigault D., Fabre C., « Approche linguistique pour l'analyse syntaxique de corpus », *Cahiers de grammaire*, vol. 25, p. 131-151, 2000.

Charolles M., « L'encadrement du dicours - Univers, champs, domaines et espaces », *Cahiers de recherche linguistique*, 1997.

Covington M. A., « GULP 3.1 : An Extension of Prolog for Unification-Based Grammar », *Artificial Intelligence*, 1994.

Cunningham H., Software Architecture for Language Engineering, PhD thesis, University of Sheffield, 2000.

Cunningham H., Maynard D., Bontcheva K., Tablan V., « GATE : A Framework and Graphical Development Environment for Robust NLP Tools and Applications », *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

Desclés J., E. C., Jackiewicz A., Minel J., « Textual Processing and Contextual Exploration Method », *Proceedings of CONTEXT 97*, Universidade Federal do Rio de Janeiro, Brazil, p. 189-197, 1997.

Enjalbert P., Gaio M., « Traitements sémantiques pour l'information géographique, textes et cartes », *Revue internationale de géomatique/European journal of GIS and Spatial Analysis*, 2006.

Ferrari S., Bilhaut F., Widlöcher A., Laignelet M., « Une plate-forme logicielle et une démarche pour la validation de ressources linguistiques sur corpus : application à l'évaluation de la détection automatique de cadres temporels », *Actes des 4e journées de linguistique de corpus (JLC)*, Lorient, France, 2005.

Ferrucci D., Lally A., « UIMA : an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment », *Natural Language Engineering*, vol. 10, p. 327-348, 2004.

Guimier de Neef E., Boualem M., Chardenon C., Filoche P., Vinesse J., « Natural language processing software tools and linguistic data developed by France Télécom R&D », *Proceedings of Indo European Conference on Multilingual Technologies*, Pune, India, 2002.

Hamon T., Derivière J., Nazarenko A., « OGMI OS : une plate-forme d'annotation linguistique de collection de documents issus du Web », *Actes de la 14e conférence sur le traitement automatique des langues naturelles (TALN) - session poster*, p. 103-112, 2007.

Marquesuzaà C., Etcheverry P., Lesbegueries J., « Exploiting Geospatial Markers to Explore and Resocialize Localized Documents », *Proceedings of the 1st Conference on GeoSpatial Semantics (GEOS)*, Mexico City, 2005.

Méger M., Rigotti C., « Constraint-based mining of episode rules and optimal window size », *In Proceedings of the ECML/PKDD'04 International Conference*, 2004.

Minel J., *Filtrage sémantique. Du résumé automatique à la fouille de textes*, Hemès, Lavoisier, 2002.

Minel J.-L., Desclés J.-P., Cartier E., Crispino G., Hazez S. B., Jackiewicz A., « Résumé automatique par filtrage sémantique d'informations dans des textes. Présentation de la plateforme Filtext », *Technique et science informatique*, 2001.

Muller C., Royaute J., Silberztein M. (eds), *INTEX pour la Linguistique et le traitement automatique des langues*, Presses Universitaires de Franche-Comté, 2004.

Paumier S., De la reconnaissance de formes linguistiques à l'analyse syntaxique, volume 2, Manuel d'Unitex, PhD thesis, Université de Marne la Vallée, 2003.

Rioult F., Crémilleux B., « Représentation condensée en présence de valeurs manquantes », *XXIIe congrès Inforsid*, Biarritz, France, p. 301-317, 2004.

Schmid H., « Probabilistic Part-of-Speech Tagging Using Decision Trees », *Proceedings of the Conference on New Methods in Language Processing*, Manchester, UK, 1994.

Setzer A., Temporal Information in Newswire Articles : an Annotation Scheme and Corpus Study, PhD thesis, Université de Sheffield, Royaume-Uni, 2001.

Vaid S., Jones C., Joho H., Sanderson M., « Spatio-Textual Indexing for Geographical Search on the Web », *Proceedings of the 9th International Symposium on Spatial and Temporal Databases*, Angra dos Reis, Brazil, p. 218-235, 2005.

Vergne J., « Analyse syntaxique automatique de langues : du combinatoire au calculatoire », *Actes de la 8e conférence sur le traitement automatique des langues naturelles (TALN) - conférence invitée*, 2001.

Widlöcher A., « Analyse par contraintes de l'organisation du discours », *in* P. Mertens, C. Fairon, A. Dister, P. Watrin (eds), *Actes de la 13e conférence sur le traitement automatique des langues naturelles (TALN)*, Presses Universitaires de Louvain, Leuven, Belgique, p. 367-376, avril, 2006.

Widlöcher A., Bilhaut F., Hernandez N., Rioult F., Charnois T., Ferrari S., Enjalbert P., « Une approche hybride de la segmentation thématique : collaboration du traitement automatique des langues et de la fouille de texte », *Actes du défi fouille de texte (DEFT), semaine du document numérique (SDN)*, Fribourg, Suisse, septembre, 2006.

Widlöcher A., Enjalbert P., « Constraint-based Analysis of Discourse Structure », *Proceedings of the 4th International Workshop on Constraints and Language Processing (CSLP)*, Roskilde University, Denmark, p. 78-92, 2007.