TALN 2011, Montpellier, 27 juin – 1er juillet 2011

Stratégie d'exploration de corpus multi-annotés avec GlozzQL

Yann Mathet[1] Antoine Widlöcher[1]
(1) GREYC, UMR CNRS 6072, Université de Caen, 14032 Caen Cedex
{prénom.nom}@unicaen.fr

Résumé. La multiplication des travaux sur corpus, en linguistique computationnelle et en TAL, conduit à la multiplication des campagnes d'annotation et des corpus multi-annotés, porteurs d'informations relatives à des phénomènes variés, envisagés par des annotateurs multiples, parfois automatiques. Pour mieux comprendre les phénomènes que ces campagnes prennent pour objets, ou pour contrôler les données en vue de l'établissement d'un corpus de référence, il est nécessaire de disposer d'outils permettant d'explorer les annotations. Nous présentons une stratégie possible et son opérationalisation dans la plate-forme Glozz par le langage GlozzQL.

Abstract. More and more works in compuational linguistics and NLP rely on corpora. They lead to an increasing number of annotation campaigns and multi-annotated corpora, providing informations on various linguistic phenomena, annotated by several annotators or computational processes. In order to understand these linguistic phenomena, or to control annotated data, tools dedicated to annotated data mining are needed. We present here an exploration strategy and its implementation within the Glozz platform, GlozzQL.

Mots-clés, Keywords: Corpus, Annotation, Exploration, GlozzQL.

1 Besoins en exploration de corpus

De nombreux travaux sur corpus, en linguistique computationnelle et en TAL, donnent lieu à des campagnes d'annotation et à la production de corpus multi-annotés. Pour mieux comprendre les phénomènes que ces campagnes prennent pour objets, pour assister le processus d'annotation et pour contrôler son résultat, il est nécessaire de disposer d'outils et de méthodes permettant d'interroger les annotations. Avant d'introduire le langage GlozzQL, qui en permet l'exploration, nous commencerons par indiquer les caractéristiques des données auxquelles il s'applique et présenter le méta-modèle sur lequel il repose.

1.1 Corpus multi-annotés

Après avoir insisté sur le fait que nous ne visons pas la mise en place d'un environnement dédié à l'étude d'une configuration linguistique particulière, mais la définition de stratégies permettant l'exploration combinée de phénomènes hétérogènes, nous pouvons en particulier mettre en avant les propriétés suivantes. **Données multi-annotateurs :** Les données explorées sont le fruit d'annotations réalisées par plusieurs annotateurs. Elles peuvent porter sur les mêmes textes et sur les mêmes objets et on souhaitera souvent pouvoir les confronter. **Variété catégorielle :** Ces corpus annotés sont porteurs d'informations linguistiques et infra-linguistiques (par exemple typo-dispositionnelles) hétérogènes correspondant à différents points de vue sur le matériau textuel. Ainsi, par exemple, on pourra disposer d'un étiquetage morpho-syntaxique, d'indications de dépendance résultant d'une analyse syntaxique, du repérage d'occurrences d'éléments de lexiques... **Configurations structurellement variées :** Les différents phénomènes annotables relèvent de modes de structuration fortement hétérogènes. Certaines approches identifient ainsi, par exemple, des segments textuels (unités morphologiques, entités nommées, segments thématiques...), là où d'autres se concentrent sur l'identification de relations entre des objets plus ou moins distants (relations syntaxiques, relations sémantiques de cohérence...) ou sur la reconnaissance de dispositifs plus complexes (chaînes de coréférence, structures énumératives...). **Granularité variable :** Ces différentes structures opèrent à des échelles pouvant varier fortement. Si le mot constitue ainsi souvent l'échelle élémentaire, les annotations syntaxiques prendront place à l'échelle de la phrase et les annotations discursives (par exemple thématiques) pourront déborder les échelles phrastiques. **Topologies variées :** La distribution des différents phénomènes répond à des « topologies » variées : parfois juxtaposés, les objets pourront aussi être imbriqués ou se chevaucher.

1.2 Exploration de corpus annotés

En quoi l'exploration de corpus annotés peut-elle alors consister ? Du point de vue de sa finalité et de ses usages, disons qu'elle vise souvent à nous assister dans la recherche d'une meilleure compréhension de phénomènes étudiés. Dans d'autres cas, elle permettra, une fois le phénomène bien connu et annoté, de naviguer entre ses occurrences, par exemple pour analyser leur population (analyse de fréquence, de corrélations...) ou vérifier que les occurrences ont été annotées conformément au modèle d'annotation fixé. Dans la plupart des cas, l'exploration doit nous permettre de valider certaines hypothèses relatives à des phénomènes, ou nous mettre sur la voie de leur formulation. On pourra alors distinguer des tâches d'exploration, en fonction du degré de supervision dont elles font l'objet, de la pure « découverte » à la validation d'une hypothèse déjà finement formulée. Au-delà de la diversité des usages, nous proposons de définir l'exploration de corpus comme la confrontation entre les données d'un corpus et un modèle (éventuellement partiel) de phénomène exploitant des informations disponibles (données textuelles brutes, annotations préalables, informations typo-dispositionnelles...), modèle visant à isoler les propriétés récurrentes des configurations visées et pouvant être aussi peu contraignant que possible dans le cas de l'exploration ouverte (fouille de texte, certaines approches statistiques...) ou au contraire très contraint dans le cas de la validation d'un modèle théorique ou opératoire bien établi.

Explorer un corpus c'est alors exprimer un modèle d'un phénomène observé, modèle faisant référence aux données disponibles. L'expression de ce modèle devra s'appuyer sur un formalisme permettant de le décrire, en fixant les contraintes que devront vérifier des configurations textuelles pour être regardées comme instances dudit modèle. La question se pose dès lors de savoir quel formalisme permettra de procéder à cette description. Pour répondre à cette question, il est nécessaire d'interroger l'adéquation entre tel ou tel formalisme de représentation et telle ou telle famille de configurations linguistiques. Pour une méthode d'exploration de corpus dédiée à un phénomène particulier, on pourra s'appuyer sur un formalisme permettant la représentation de configurations structurellement homogènes (motifs séquentiels, arbres syntaxiques...). Pour l'exploration combinée de structures hétérogènes, on s'appuiera idéalement sur un formalisme suffisamment versatile autorisant la représentation, dans un langage unifié, d'objets structurellement très différents. Ajoutons que le choix du formalisme d'interrogation doit également être guidé, en plus de son expressivité, par sa déclarativité et son aptitude à masquer l'appareil procédural qui prendra en charge l'analyse effective des données.

Se pose également la question de la place que doit occuper l'exploration de corpus au sein du dispositif expérimental. Si un usage classique des outils d'exploration est d'étudier un corpus après annotation de ce dernier, leur utilisation au sein même du processus d'annotation peut aussi se révéler pertinente. En effet, l'annotation manuelle de corpus consiste souvent à opérer de façon incrémentale en alternant phases d'observation et d'annotation.

Enfin, il est important de préciser que le système d'exploration de corpus devra être couplé à un mécanisme de retour au texte permettant, étant donnée une liste d'occurrences correspondant aux contraintes exprimées, de les consulter *in situ*.

1.3 État de l'art

Guidée par l'expression d'un modèle, l'exploration de corpus pourra s'appuyer sur des outils dédiés à une famille très spécifique de configurations linguistiques. De ce point de vue, tout formalisme d'expression de règles et de contraintes consacré à un domaine d'étude particulier ou à un paradigme formel particulier, couplé à un analyseur automatique[1], pourrait être utilisé pour une exploration ainsi orientée. Il permet en effet par nature de mettre en relation des informations préexistantes (par exemple morphologiques) pour découvrir des configurations d'ordre supérieur (par exemple syntaxiques) et pourra donc être exploité pour interroger le corpus. Toutefois, de tels dispositifs ont principalement pour objectif d'annoter automatiquement un corpus et ne visent pas prioritairement à produire des représentations synthétiques (liste d'occurrences, concordanciers...). De plus, souvent dédiés à un type d'objet, il ne permettent pas d'interroger conjointement des éléments hétérogènes, pour explorer leurs éventuelles corrélations avec les structures visées. Enfin, et plus généralement, ces structures visées ne pourront appartenir qu'à des catégories restreintes : un formalisme dédié à l'analyse syntaxique ne permettra souvent pas d'explorer des structures thématiques ou des chaînes de coréférence.

Le besoin d'interroger des données relatives à des phénomènes linguistiques variés et structurellement hétérogènes

[1] Évoquons par exemple le cas d'Unitex (Paumier, 2003), pour les approches orientées-automates.

pourra trouver une réponse partielle dans l'utilisation d'environnements intégrés d'expérimentation et d'annotation automatique tels que GATE (Cunningham *et al.*, 2002), LinguaStream (Widlöcher & Bilhaut, 2008), ou encore UIMA, dans l'utilisation qui en est faite en TAL, par exemple par (Hernandez *et al.*, 2010). De tels environnements reposent en effet sur des modèles d'annotation suffisamment abstraits pour qu'il soit possible de représenter des objets linguistiques variés et de les interroger de manière unifiée, conformément à notre programme. De plus, ces environnements proposent des formalismes de représentation de règles variés (règles séquentielles, automates, grammaires de contraintes...) permettant d'encoder des modèles linguistiques très hétérogènes, afin de les projeter sur corpus. S'il est souvent possible, en aval d'une chaîne de traitement, de visualiser et d'explorer les données annotées, et si les formalismes proposés peuvent être mis au service d'un travail d'exploration, il convient cependant de noter que ces outils sont prioritairement destinés à l'annotation automatique et qu'ils exigent un travail de mise en œuvre trop important pour la seule exploration de données pré-constituées.

D'autres travaux s'orientent davantage vers la mise en place d'environnements d'exploration de corpus multi-annotés. Si CorpusReader (Loiseau, 2008) prend place sur ce terrain de l'exploration de corpus, il convient de noter qu'il se situe en amont de l'exploration proprement dite, davantage chargé de la constitution des observables que de leur interrogation. Il vise en effet principalement à intégrer des annotations issues de plusieurs niveaux de description linguistique et d'analyseurs variés, en préalable de l'étude des corrélations entre ces niveaux. Du reste, la préparation des données pour laquelle il a été initialement pensé l'oriente de manière privilégiée vers des études statistiques, au demeurant non intégrées à l'outil lui-même. La plate-forme Annis (Chiarcos *et al.*, 2008) offre l'environnement d'exploration de corpus multi-annotés le plus proche de nos préoccupations. Elle s'appuie sur le modèle générique Paula qui permet d'exploiter des annotations provenant d'outils variés, procédant à des niveaux d'analyse éventuellement différents, et est directement opérationnelle avec nombre de documents déjà annotés. Elle dispose d'un langage d'interrogation unifié, AnnisQL, qui permet d'adresser des requêtes à ces annotations hétérogènes. Cependant, ces travaux privilégient l'interopérabilité et l'interrogation d'annotations préexistantes, là où nous souhaitons permettre d'interroger les données pendant le processus d'annotation, pour le guider.

2 Cadre théorique et opérationnel

La mulitplicitié des phénomènes linguistiques observables conduit assez naturellement à la multiplicité des modèles, des théories, formalismes et formats permettant d'en rendre compte. Or nous souhaitons pouvoir étudier de manière unifiée ces phénomènes hétérogènes, pouvoir explorer leurs combinaisons et interactions.

Afin d'uniformiser la représentation des phénomènes observables et annotables, nous nous appuyons sur le méta-modèle URS (pour Unité-Relation-Schéma), issu de (Widlöcher, 2008), qui propose un cadre théorique abstrait, non dédié à telle ou telle théorie ou à tel ou tel type d'objet, et permettant conséquemment la description de modèles linguistiques variés. Dans le cadre fixé par le méta-modèle, des modèles spécifiques, relatifs à telle ou telle théorie, peuvent être exprimés, en déclarant les classes d'objets linguistiques identifiables et en explicitant la manière sont leurs instances doivent être caractérisées.

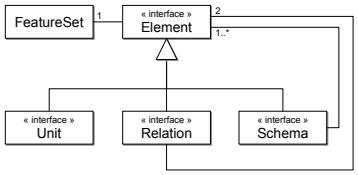

FIG. 1 – Diagramme de classes UML du méta-modèle URS

Représenté par la figure 1, ce méta-modèle repose sur l'articulation de trois catégories abstraites d'**éléments**, qui représentent l'ensemble des objets manipulables. Ces *éléments* peuvent être plus spécifiquement des *unités*, des *relations* ou des *schémas*. Tous sont caractérisés par un *type*, qui explicite typiquement leur catégorie linguistique, et par un ensemble de *traits* qui représentent leurs propriétés. Les catégories manipulables, les traits disponibles pour un type donné et les valeurs qu'ils peuvent prendre dépendent d'un modèle d'annotation spécifique défini pour une campagne donnée. La notion d'**unité** renvoie à des segments textuels, de taille quelconque : mots, syntagmes, phrases, segments thématiques... Une **relation** désigne un lien, orienté ou non, entre deux éléments. Si les relations

entre unités sont les plus utilisées, notons que l'expressivité du méta-modèle autorise également la définition de relations portant sur des schémas ou des relations. Relations syntaxiques de dépendance et relations sémantiques de cohérence consituent des exemples bien connus de telles relations. Les **schémas** peuvent pour leur part être utilisés pour représenter des configurations complexes, éventuellement récursives, impliquant une combinaison quelconque d'élements de plus bas niveau (unités, relations ou schémas imbriqués). Une chaîne de coréférence pourra par exemple être représentée par un ensemble de relations binaires groupées au sein d'un schéma dont les traits caractériseront la référence commune. Les structures énumératives constituent un autre exemple de structure complexe exigeant une telle expressivité. Il est important de remarquer que ce méta-modèle abstrait peut être utilisé pour représenter une grande diversité de structures, que celle-ci soient du reste strictement linguistiques ou non. Ainsi, la structure d'un document (sections, titres...) et les indications typographiques (graisse, italique, listes...), éléments dont on sait l'importance pour l'exploration de certains phénomènes de langue, peuvent être encodées sans sortir de ce cadre. Représentées de manière homogène, toutes les informations disponibles pourront être explorées de manière unifiée.

La plate-forme d'annotation et d'exploration de corpus Glozz (Widlöcher & Mathet, 2009) utilise ce méta-modèle. La méthode d'exploration présentée ci-après y a été implémentée.

3 Paradigme des contraintes et GlozzQL[2]

Le paradigme des contraintes offre un cadre adapté à la prise en considération de l'hétérogénéité des données évoquées en section 1.1[3]. En effet, ce paradigme permet en particulier de représenter de manière uniforme des catégories de structures irréductibles les unes aux autres (unités, relations et schémas), pouvant opérer à des échelles très variées, en s'abstrayant si nécessaire de leur inscription textuelle effective, de l'ordre des éléments du texte et des distances qui les séparent. Le Langage GlozzQL repose sur ce paradigme[4].

3.1 GlozzQL, un langage de requêtes incrémental et à fort pouvoir expressif

GlozzQL ayant vocation à être utilisé par un public large, il repose sur des concepts simples et une interface graphique pour la construction des requêtes, des assemblages incrémentaux permettant de parvenir progressivement à l'expressivité nécessaire. Il repose sur deux concepts interdépendants, la notion de contrainte ($Constraint$), et la notion d'annotation contrainte ($ConstrainedAnnotation$).

Constraint : Une contrainte exprime une condition que doit satisfaire une annotation pour être sélectionnée. 20 types de contraintes sont prédéfinis, couvrant un large spectre de requêtes. Certaines contraintes peuvent s'appliquer spécifiquement à un type d'élément particulier (unité, relation, schéma), d'autres sont universelles, et enfin 3 contraintes portent sur d'autres contraintes afin de les modifier ou de les combiner (Not, Or, And).

Parmi les contraintes universelles, mentionnons notamment $Feature$ qui permet de contraindre des valeurs de traits, $Author$ qui permet de contraindre l'identifiant de l'annotateur dans le cas de corpus multi-annotés, et quelques autres qui ont trait à la position relative d'entités par rapport à d'autres ($Distance$, $After$, $Before$).

Parmi les requêtes spécifiques aux unités, $TextContains$ permet de spécifier une séquence de texte que l'entité doit contenir, $RegExp$ définit une expression régulière à laquelle le texte contenu dans l'unité doit se conformer, tandis que $Covers$ et $CoveredBy$ sont relatives à d'autres unités, et imposent respectivement le fait que l'unité contrainte recouvre un certain type d'unité, et la réciproque. Par exemple, $C1 = textContains(\text{"}isotope\text{"})$ est une contrainte portant sur une unité et imposant que cette dernière contienne le texte "isotope".

Concernant les schémas, la contrainte $Contains$ spécifie que ce dernier doit contenir un type d'entité particulier, par exemple une unité contenant le texte "isotope". Cette contrainte est doublement paramétrable, d'une part en autorisant ou non la recherche en profondeur, d'autre part en contraignant le nombre d'entités concernées.

Enfin, deux contraintes sont spécifiques aux relations, $Start$ et $Target$, qui permettent respectivement de spécifier

[2] Manuel et vidéos de démonstration de GlozzQL sont disponibles sur `http://www.glozz.org`.

[3] Ce point a été amplement discuté dans (Widlöcher, 2008).

[4] Selon une autre stratégie d'exploration, assez opportuniste, il est également possible, depuis Glozz, d'exporter simplement les données annotées vers un SGBDR et de bénéficier de l'expressivité et de la puissance de SQL pour les interroger.

la source ou la cible d'une relation de façon similaire à la contrainte $Contains$ des schémas.

Constrained Annotation : De façon corollaire, on appelle "annotation contrainte" un ensemble d'annotations satisfaisant une contrainte donnée. Par exemple, $Unit1 : C1$ définit l'ensemble des unités qui satisfont la contrainte $C1$, c'est-à-dire qui contiennent le texte "isotope".

Construction incrémentale et interdépendante des Constraints et des ConstrainedAnnotations : Une annotation contrainte dépend, par définition, d'une contrainte. La réciproque peut être vraie, dans la mesure où, comme nous l'avons vu, certaines contraintes s'appuient sur des annotations contraintes. C'est par exemple le cas de $C2 = Contains(Unit1)$, qui est une contrainte exprimant le fait, pour un schéma, de contenir une unité de $Unit1$, c'est-à-dire contenant le texte "isotope". Pour obtenir l'ensemble des schémas correspondants, il suffit de définir l'annotation contrainte correspondante $Schema1 : C2$. De la sorte, il est possible de construire de façon incrémentale des contraintes et des annotations contraintes de plus en plus riches. L'avantange est double : d'une part, chaque étape du processus de construction de requêtes est élémentaire ; d'autre part, chacune de ces étapes constitue elle-même un résultat partiel éventuellement intéressant et exploitable en soi (en cherchant tous les schémas contenant une unité contenant le texte "isotope", on a aussi et d'abord trouvé l'ensemble des unités contenant "isotope").

3.2 Exemple réel de constitution de requêtes

Observons un exemple issu d'une campagne d'annotation dans laquelle des schémas appelés SE (structures énumératives) peuvent contenir, entre autres, des unités appelées "amorces". Nous souhaitons accéder d'une part à tous les schémas comportant une amorce, et d'autre part à tous ceux qui n'en comportent pas.

Constraints

Constrain...	Content	Scope
C1	TypeName = amorce	Any
C2	Contains(Unit1), Level=1 min=1 max=No...	Rel/Schema
C3	Not(C2)	Rel/Schema
C4	TypeName = SE	Any
C5	And(C2,C4)	Rel/Schema
C6	And(C3,C4)	Rel/Schema

Constrained Annotations

Annotation	Constraint	Matches
Unit1	C1-->TypeName = amorce	6
Schema1	C5-->And(C2,C4)	6
Schema2	C6-->And(C3,C4)	1

FIG. 2 – Exemple de construction de requêtes GlozzQL

Pour ce faire, comme illustré par la figure 2, nous créons $Unit1$ comme l'ensemble des unités de type amorce, qui s'appuie sur la contrainte $C1$. Nous créons ensuite la contrainte $C2$ imposant qu'un schéma contienne un élément de $Unit1$, et sa contraposée $C3$. Comme nous souhaitons obtenir uniquement les schémas de type SE, nous créons la contrainte idoine $C4$. Il reste alors à créer $C5$ par conjonction de $C2$ et $C4$, imposant qu'un schéma soit de type SE et comporte une amorce, et $C6$ par conjonction de $C3$ et $C4$. Nous en tirons respectivement $Schema1$ et $Schema2$, qui répondent au problème posé. La dernière colonne "Matches" indique pour chacune des $ConstrainedAnnotations$ quel en est le nombre d'occurrences dans le texte annoté en cours. Rappelons que par ce processus incrémental, nous obtenons des résultats intermédiaires à chacune des étapes, comme $Unit1$ qui nous donne l'ensemble des unités "Amorces", au nombre de 6 dans cet exemple.

3.3 Exploration et exploitation des résultats

Navigation : À tout moment, chacune des $ConstrainedAnnotations$ se voit attribuer l'ensemble des occurrences correspondantes dans le texte actif, et un clic sur l'une d'entre elles permet d'accéder à son contenu. Dans l'exemple de la figure 3, un clic sur $Schema1$ fait apparaître dans la fenêtre du dessous les 6 occurrences de schémas concernées (flèche 1 de la figure 3). Dès lors, un clic sur l'une de ces occurrences permet de sélectionner et d'accéder immédiatement à son contenu dans la fenêtre principale (flèche 2 de la figure 3), ce qui rend possible son observation dans le texte ainsi que, éventuellement, sa modification immédiate.

Panier : Un système de panier permet de collecter sélectivement des paquets de résultats (par exemple tous les élements de $Schema1$ et de $Schema2$). Il est alors notamment possible d'enregistrer le contenu du panier en tant que nouveau fichier d'annotation, ou au contraire de demander à ce que les annotations contenues dans le panier soient supprimées du texte en cours d'annotation.

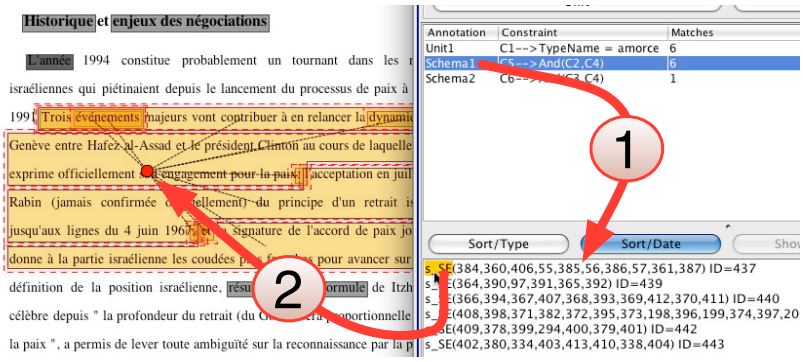

FIG. 3 – Exploration des résultats d'une requête GlozzQL au sein du processus d'annotation

Unification : Un mécanisme optionnel appelé "unification", dans le détail duquel nous ne pouvons entrer dans le cadre de cet article, permet de considérer chaque $ConstrainedAnnotation$ comme une variable entrant dans le système d'équations constitué par l'ensemble des contraintes. De la sorte, il est notamment possible de rechercher des configurations telles que trois unités liées par des relations triangulaires, de chercher toutes les relations orientées du bas vers le haut, etc. (ce qui serait impossible par défaut).

Usages de l'outil au sein d'une campagne d'annotation : Les requêtes créées avec GlozzQL peuvent être sauvegardées (dans un format spécifique XML), et des jeux de requêtes peuvent donc faire partie intégrante du matériau fourni aux annotateurs d'une campagne donnée. Par ailleurs, le fait de disposer d'un tel outil au sein même du processus d'annotation permet notamment : **1)** d'accéder à tout moment à des configurations particulières pouvant servir de base à la suite du processus d'annotation et **2)** de contrôler à tout moment la conformité des annotations produites. Il suffit pour cela de créer des $ConstrainedAnnotations$ correspondant à des configurations interdites pour la campagne d'annotation en cours.

Références

CHIARCOS C., DIPPER S., GÖTZE M., LESER U., LÜDELING A., RITZ J. & STEDE M. (2008). A Flexible Framework for Integrating Annotations from Different Tools and Tag Sets. *Revue Traitement Automatique des Langues (TAL)*, **49**(2), 189–215.

CUNNINGHAM H., MAYNARD D., BONTCHEVA K. & TABLAN V. (2002). GATE : A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

HERNANDEZ N., POULARD F., VERNIER M. & ROCHETEAU J. (2010). Building a French-speaking community around UIMA, gathering research, education and industrial partners, mainly in Natural Language Processing and Speech Recognizing domains. In *Workshop Abstracts LREC 2010 Workshop 'New Challenges for NLP Frameworks'*, p. p64, La Valleta Malta.

LOISEAU S. (2008). CorpusReader : construction et interrogation de corpus multiannotés. *Revue Traitement Automatique des Langues (TAL)*, **49**(2), 189–215.

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

WIDLÖCHER A. (2008). *Analyse macro-sémantique des structures rhétoriques du discours - Cadre théorique et modèle opératoire*. PhD thesis, Université de Caen Basse-Normandie.

WIDLÖCHER A. & BILHAUT F. (2008). Articulation des traitements en TAL - Principes méthodologiques et mise en œuvre dans la plate-forme LinguaStream. *Revue Traitement Automatique des Langues (TAL)*, **49**(2), 73–101.

WIDLÖCHER A. & MATHET Y. (2009). La plate-forme Glozz : environnement d'annotation et d'exploration de corpus. In *Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis, France : ATALA LIPN.