Une infrastructure pour l'annotation linguistique de documents issus du web : le projet ALVIS

Sophie Aubin, Julien Derivière, Thierry Hamon,
Adeline Nazarenko, Thierry Poibeau, Davy Weissenbacher

Laboratoire d'Informatique de Paris-Nord
Université Paris 13 & CNRS (UMR 7030)
99, avenue J.-B. Clément – F-93430 Villetaneuse
{prenom.nom}@lipn.univ-paris13.fr
http://www-lipn.univ-paris13.fr

Résumé. Cet article présente une architecture logicielle, la plate-forme Ogmios, permettant l'annotation automatique de documents issus du web. Cette architecture est fondée sur l'intégration de composants d'analyse linguistique et présente une double originalité : elle peut être adaptée en fonction du domaine visé et elle peut analyser de manière robuste des collections de documents hétérogènes, ce qui est le propre des collections construites à partir du web. Cet article prend comme exemple une collection de documents du domaine de la biologie. Nous montrons comment la plateforme Ogmios peut être adaptée à ce domaine et nous détaillons les performances obtenues suite à cette adaptation. Les résultats de l'analyse des documents par la plate-forme peuvent ensuite être pris en compte par des moteurs spécialisés sur internet.

1 Introduction

Les moteurs de recherche comme Yahoo ou Google permettent aujourd'hui d'accéder à des milliards de pages web. Ces outils sont très populaires et semblent suffisants pour répondre aux requêtes les plus courantes sur Internet. Mais l'utilisateur cherche parfois une information plus complexe : il peut alors souhaiter formuler sa requête en s'appuyant sur des techniques de recherche avancées (filtrage sur le sens, élimination d'ambiguïtés, exclusion des sites marchands, etc.) et sur des connaissances du domaine. Il n'existe pas à l'heure actuelle d'outil permettant d'exprimer ce genre de requêtes.

Le projet ALVIS vise à développer un moteur de recherche libre de droit, dont les sources sont en accès libre (*open source*), incluant des techniques de recherche avancées, notamment du point de vue sémantique. Par rapport aux moteurs de recherche actuels, ALVIS cherche à prendre en compte à la fois le thème et le contexte de la recherche, pour affiner l'analyse de la requête et du document. Le projet s'appuie sur une architecture pair à pair (*peer-to-peer*) : le système est constitué d'un réseau de « nœuds » assurant l'infrastructure de recherche globale ; certains nœuds peuvent être spécialisés pour des domaines particuliers. Dans cette optique, un nœud peut gérer une collection particulière de documents, qui est généralement construite à l'aide d'un moissonneur (*crawler*) dédié et qui est peut être indexée de manière spécifique.

Au-delà des moteurs de recherche généralistes, il y a en effet un besoin pour des moteurs de recherche spécialisés. Dans le domaine de la bio-médecine, par exemple, il existe des moteurs de recherche dédiés aux grandes bases de données textuelles qui recensent l'essentiel des publications scientifiques (Flybase est spécialisé sur l'espèce Drosophilia Menogaster, Medline traite de tout ce qui a trait à la médecine et à la biologie). Même si ces outils sont couramment utilisés par les scientifiques, on constate qu'ils ne répondent qu'imparfaitement à leurs besoins de fouille de texte. Les moteurs disponibles sont trop généraux, ils renvoient des centaines ou des milliers d'articles pour la moindre requête. Retrouver beaucoup de documents ne suffit pas, il faut les sélectionner avec précision, en analyser rapidement le contenu voire en extraire des informations, et, à terme, fusionner les informations extraites avec celles qui peuvent être déjà enregistrées au sein de bases de données.

Pour répondre à ces besoins, il faut une véritable analyse du contenu textuel, plus ou moins fine selon l'application visée et le volume de données concerné. En recherche d'information, on sait par exemple que le repérage des noms d'entités ou de certains termes techniques est un élément clé pour juger de l'importance d'un document. Alphonse et al. (2004) ont montré par ailleurs que l'identification des interactions entre gènes demande l'étiquetage des noms de gènes, la reconnaissance de termes du domaine ainsi qu'une analyse syntaxique fiable.

Nous avons donc développé une architecture logicielle pour l'enrichissement et l'annotation de documents issus du web. Cette plate-forme, Ogmios, est générique dans la mesure où elle inclut des modules permettant d'analyser de manière robuste tout type de documents textuels dans une langue donnée (pour l'instant le français et l'anglais sont instanciés, mais des chaînes de traitement ont également été développées pour le chinois et le slovène à travers ALVIS). Cette plate-forme peut également être spécialisée pour des domaines particuliers. Dans le cadre du projet ALVIS, les premières expériences ont porté sur le domaine de la biologie : nous montrons dans cet article comment nous avons pu analyser une collection de documents fournis par un moissonneur spécialisé dans ce domaine.

Dans la section 2, nous donnons un aperçu de l'état de l'art concernant les plates-formes d'annotation de documents. La plate-forme elle-même est décrite dans la section 3. Nous présentons ensuite, dans la section 4, les modules de traitement intégrés puis, dans la section 5, l'adaptation au domaine de la biologie. Des éléments permettant d'évaluer les performances d'Ogmios sont fournis dans la section 6.

2 Etat de l'art

La dernière décennie a vu se développer diverses architectures d'ingénierie du texte organisant les traitements linguistiques (Cunningham et al., 2000).

La plate-forme GATE (*General Architecture for Text Engineering*) (Bontcheva et al., 2004) a été conçue essentiellement pour des tâches d'extraction d'information. Elle vise à réutiliser des outils de Traitement Automatique des Langues (dorénavant TAL) sous forme de modules intégrés. Le format d'annotation et d'échange (CPSL – Common Pattern Specific Language) repose sur le format d'annotation TIPSTER (Grishman, 1997).

La plate-forme KIM (Popov et al., 2004) peut être considérée comme une « méta-plate-forme », exploitant une plate-forme d'annotations linguistiques externe, en l'occurrence GATE. Il s'agit en effet d'une architecture dédiée à l'enrichissement d'ontologies, l'indexation sémantique et la recherche d'information. KIM a été utilisée dans des projets d'annotation sémantique

massive tels que SWAN[1] et SEKT[2]. Les auteurs de cette plate-forme considèrent le passage à l'échelle comme un paramètre critique pour deux raisons : (1) la plate-forme doit être en mesure de traiter d'importants volumes de données pour construire et entraîner des modèles statistiques pour l'extraction d'information ; (2) elle doit également pouvoir être utilisée en tant que service web, ce qui signifie traiter des volumes importants de documents de façon dynamique.

UIMA (Ferrucci et Lally, 2004) – une nouvelle implémentation de l'architecture de TEXTRACT (Neff et al., 2004) – est très similaire à GATE. La principale différence entre les deux réside dans le modèle de représentation des données. UIMA est un « squelette » servant au développement de moteurs d'analyse. Les composants proposés permettent l'analyse de flux d'information peu structurés (par exemple des pages HTML) ; ils sont capables de traiter des volumes de données très variables. Le format d'annotation d'UIMA nommé CAS (Common Analysis Structure) reprend le format TIPSTER (Grishman, 1997). Afin de préserver une certaine flexibilité, les annotations y sont déportées (c'est-à-dire stockées en dehors du document original). La plate-forme offre la possibilité de traiter les documents les uns après les autres ou sous forme d'une collection : les collections sont gérées par le Collection Processing Engine (CPE), qui regroupe des fonctionnalités comme le filtrage, la surveillance des performances et la parallélisation.

La plate-forme Textpresso (Müller et al., 2004) a été conçu pour la fouille de documents traitant de biologie, aussi bien des résumés que des articles complets. Le système a été évalué sur les résultats de Medline pour des requêtes telles que *Caenorhabditis elegans*. Il est ainsi possible de traiter 16 000 résumés et 3 000 articles en texte brut. Textpresso est aussi conçu pour extraire les relations entre gènes. La plate-forme est composée de plusieurs modules de traitement linguistique : tokeniseur, segmenteur en phrases, étiqueteur morpho-syntaxique et étiqueteur d'ontologie exploitant les informations fournies par Gene Ontology (Consortium, 2001). Bien que Textpresso ait été conçu spécifiquement pour les textes bio-médicaux, l'objectif de notre plate-forme est plus proche de celui de GATE : proposer une plate-forme générique capable de traiter des corpus de documents volumineux issus d'un domaine spécialisé.

D'une manière générale, on dispose de peu d'informations pour évaluer le comportement de systèmes sur un corpus donné, alors que, de notre point de vue, il s'agit d'un aspect crucial pour ce type de systèmes. Un premier test nous a montré que GATE ne convient pas au traitement de gros corpus. GATE a été conçue comme un environnement puissant de développement et de conception d'applications de TAL dans le cadre de l'extraction d'information. Tant que ce type d'application ne traite que de petits volumes de documents, le passage à l'échelle n'est pas un objectif central.

Notre approche est différente de celle de GATE dans la mesure où nous avons cherché à proposer une plate-forme capable de gérer d'importants volumes de documents en nous concentrant sur l'efficacité et la robustesse des traitements effectués. Elle diffère également de celle de Textpresso : Ogmios n'est pas une plate-forme dédiée à la biologie mais une plate-forme générique qui peut être adaptée à différents domaines, notamment la biologie.

[1] `http ://deri.ie/projects/swan`
[2] `http ://sekt.semanticweb.org`

3 Une plate-forme modulaire et adaptable

Nous avons choisi de nous appuyer sur l'existant en matière de traitement automatique des langues (TAL) : Ogmios exploite en priorité des outils de TAL disponibles[3]. Notre effort a davantage porté sur la robustesse des traitements : il s'agit d'annoter rapidemment de grandes quantité de documents issus du web, donc hétérogènes. Nous verrons que ceci a imposé une gestion distribuée des traitements. Nous avons également mis l'accent sur le processus d'adaptation permettant de spécialiser l'analyse pour un domaine particulier.

La plate-forme Ogmios offre par ailleurs la possibilité de tester différentes combinaisons d'annotations pour identifier, par exemple, les informations dont l'impact est significatif dans le cadre d'un apprentissage de règles d'extraction ou la recherche d'information.

3.1 Contraintes spécifiques

Le développement d'une telle plate-forme impose différents types de contraintes.

La réutilisation d'outils de TAL existants impose de gérer l'hétérogénéité des formats d'entrées/sorties des outils utilisés dans la plate-forme. Comme chaque outil a généralement ses formats propres, il faut un format d'échange permettant l'interconnexion de plusieurs outils et leur intégration. La réutilisation d'outils de TAL nécessite aussi de repenser la modularité de ces outils pour les intégrer dans un même processus d'analyse. Nous verrons dans la suite le découpage en modules tel que nous l'avons proposé et son importance pour l'adaptation de la plate-forme à des sous-langages et des domaines spécialisés.

La plate-forme Ogmios devant permettre à la fois une analyse rapide de gros volumes de données textuelles et une analyse plus approfondie de corpus plus modestes, elle doit être facilement paramétrable.

Analyser des collections de documents construites à partir du web impose de fortes contraintes en termes de robustesse. Cela concerne principalement le volume et l'hétérogénéité des données d'entrée : il faut pouvoir traiter plusieurs centaines de millions de mots à un rythme compatible avec celui du moissonneur qui rassemble la collection de documents ; il faut pouvoir analyser tous types de documents (des gros comme des petits, des articles scientifiques comme des résumés ou des ouvrages en ligne, des documents écrits dans différentes langues et avec des jeux de caractères variés).

Dans la mesure où nous avons proposé de distribuer les traitements, il faut également répondre à certains critères de robustesse dans l'exploitation des ressources informatiques utilisées.

3.2 Architecture générale

Les différentes étapes de traitement sont traditionnellement prises en chage par un ensemble de modules (Bontcheva et al., 2004). Chaque module est dédié à un type de traitement : reconnaissance d'entités nommées, segmentation en mots, étiquetage morpho-syntaxique, analyse syntaxique, etc. Un module encapsule un outil effectuant un certain type d'analyse linguistique et assure la conformité du format des entrées/sorties avec le format général des annota-

[3]Nous avons développé des outils lorsque aucun outil répondant à nos besoin n'était disponible ou nous convenait. Nous avons, de plus, choisi de préférence des logiciels sous licence GPL ou gratuits pour un usage non commercial.

tions de la plate-forme. Les annotations sont enregistrées dans un format XML déporté (*stand-off* ou *offline annotations*) afin de pouvoir mieux gérer l'hétérogénéité des entrées/sorties des outils de TAL (ce format d'annotation est décrit dans (Nazarenko et al., 2006)). La modularité de l'architecture facilite la substitution d'un outil par un autre.

La spécialisation de la plate-forme pour un domaine spécifique est assurée soit par le remplacement d'un module « générique » par un module spécialisé, soit par l'intégration de connaissances du domaine. Par exemple, pour analyser des résumés de biologie de Medline, une liste d'espèces ou de gènes peut être ajoutée au module de repérage d'entités nommées.

L'architecture de la plate-forme est présentée à la figure 1. Les différents modules composant la chaîne de traitement linguistique sont représentés sous forme de boîtes. Ces modules sont décrits dans la section 4. Les flèches en trait plein représentent le flux de données lors du traitement. Les flèches en pointillés montrent comment les différentes ressources peuvent être utilisées dans la plate-forme.

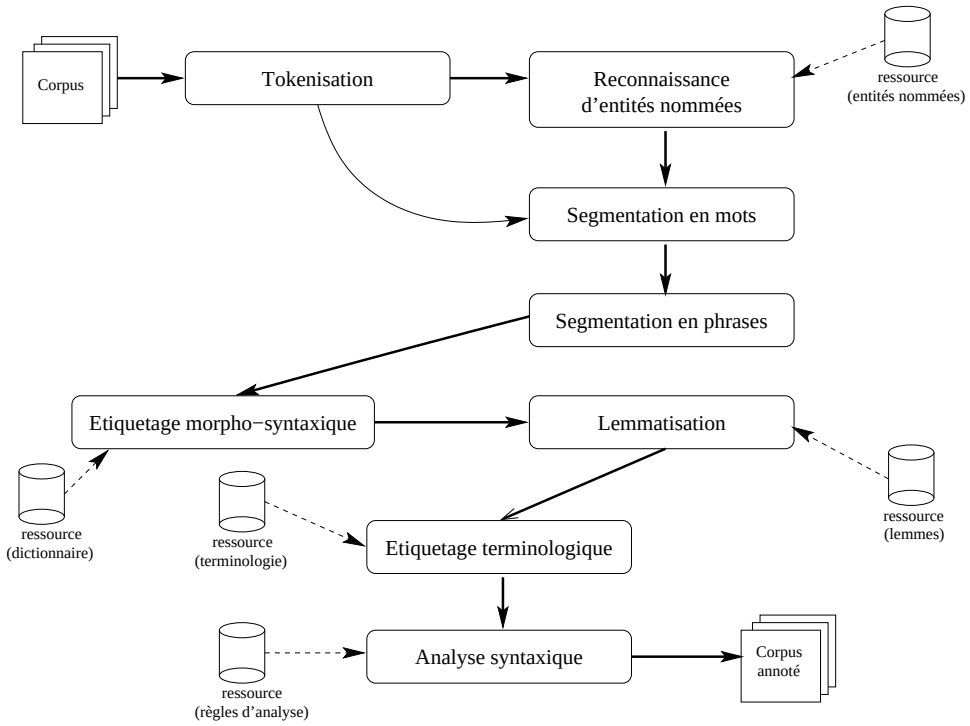

FIG. 1 – *Architecture de la chaîne de traitement. Le module de résolution des anaphores n'est pas encore intégré.*

Nous partons du principe que les documents Web donnés en entrée ont déjà été téléchargés, nettoyés, codés en UTF-8 et convertis au format XML (Nazarenko et al., 2006). Dans ALVIS, ce prétraitement est assuré par le moissonneur mais un module de prétraitement autonome pourra être développé. Dans un premier temps, les documents sont tokenisés, ce qui permet de définir des offsets (indices délimitant une séquence, en nombre de caractères par rapport au

début du document) pour garantir l'homogénéité des différentes annotations. Les documents sont ensuite traités par divers modules : repérage d'entités nommées, segmentation en mots et en phrases, lemmatisation, étiquetage morpho-syntaxique, étiquetage terminologique, analyse syntaxique et résolution des anaphores.

Cette architecture est assez traditionnelle, mais certains points méritent d'être commentés :

- La tokenisation constitue la première étape de la chaîne. Elle procède à une segmentation préliminaire, non linguistique, qui est utilisée ensuite par les autres outils. Le token est donc l'unité textuelle de base dans la chaîne de traitement : ce n'est qu'un point de départ pour les autres annotations. Ce niveau d'annotation suit les recommandations du groupe ISO/TC 37/SC 4, même si nous employons le terme d'*offset de caractère* plutôt que celui de *pointeur d'élément* pour désigner les frontières de chaque token. Pour simplifier les traitements suivants, nous distinguons quatre types de tokens (alphabétiques, numériques, séparateurs et symboliques) en fonction des caractères qui les composent.
- L'étiquetage des entités nommées se produit très tôt dans la chaîne de traitement car l'identification des entités nommées facilite la désambiguïsation d'un certain nombre de ponctuations lors de la segmentation en mots ou en phrases. Il s'agit en réalité d'un premier étiquetage assez fruste, que l'étiquetage terminologique peut venir compléter.
- L'étiquetage terminologique précède l'analyse syntaxique. Même s'il exploite des informations syntaxiques partielles, nous considérons l'étiquetage terminologique comme une étape préparatoire à l'analyse syntaxique.
- L'analyse peut s'interrompre après chaque étape, ce qui permet de produire différents niveaux d'annotation pour les documents.

Les modules sont appelés de manière séquentielle pour chaque document. Les sorties (annotations) sont stockées en mémoire jusqu'à la fin du traitement du document en cours. Les sorties sont ensuite enregistrées au format XML.

4 Description des modules de traitement

Cette section décrit les différents modules intégrés à l'heure actuelle au sein de la chaîne de traitement. Cette description s'appuie sur une instance de la plate-forme pour le traitement de l'anglais mais il va de soi que d'autres traitements et d'autres modules auraient pu être choisis.

En l'état actuel du TAL, le découpage en modules élémentaires ou en fonctionnalités de base n'est pas stabilisé. Certains outils regroupent plusieurs fonctionnalités, d'autres n'en assurent qu'une seule... Ceci empêche dans certains cas le remplacement d'un module par un autre. Dans la mesure du possible, nous avons cependant limité ces cas de figure.

Reconnaissance des entités nommées Le module assurant la reconnaissance des entités nommées vise à identifier les séquences textuelles référant à une entité, à lui donner un type sémantique et, le cas échéant, à normaliser cette séquence (le nom "B. Subtilis" est identifié comme un nom d'espèce et rattaché à la forme canonique "Bacillus Subtilis"). Chaque élément reçoit une étiquette en fonction de son type sémantique (qui dépend donc du domaine[4]). Du point de vue de l'analyse morpho-syntaxique, les tokens formant une entité nommée sont

[4]Pour la biologie, les étiquettes `gene` et `species` ont été définies pour annoter les gènes et les espèces.

regroupés et sont assimilés à des groupes nominaux. En les reconnaissant à un stade très préliminaire dans l'analyse, on évite des ambiguïtés ultérieures. Le module intégré est TagEN (Berroyer et Poibeau, 2004), qui repose sur des dictionnaires et un jeu de règles écrites sous la forme de transducteurs.

Segmentation en phrases et en mots Ce module identifie les phrases et les mots. Il s'appuie sur un ensemble d'expressions régulières reprenant l'algorithme proposé par Grefenstette (1994). Une partie de la segmentation est effectuée par le module de reconnaissance des entités nommées dans la mesure où celui-ci résout un grand nombre des problèmes liés à la ponctuation. C'est par exemple le module traitant les entités qui permet de reconnaître la séquence "B. subtilis", et qui met en rapport l'abréviation "B." avec la forme étendue "Bacillus". Du coup, le point présent dans la séquence "B. subtilis" n'a plus à être analysé (par défaut, le segmenteur ne redécoupe pas des séquences déjà identifiées).

Analyse morpho-syntaxique Ce module associe une étiquette morpho-syntaxique à chaque mot du texte. Il repose sur la segmentation effectuée à l'étape précédente. Nous utilisons à l'heure actuelle le TreeTagger (Schmid, 1997) mais le GENIA Tagger[5] a également été utilisé pour l'analyse des corpus de biologie.

Lemmatisation Ce module associe un lemme à chaque mot du texte ("protein" est le lemme de "proteins", par exemple). Si le mot ne peut pas être lemmatisé (cas des nombres, des mots étrangers et des mots inconnus), aucune information n'est associée à la forme. Ce module suppose que l'analyse morpho-syntaxique a été préalablement effectuée. Dans notre implémentation, la lemmatisation est effectuée en même temps que l'analyse morpho-syntaxique par le TreeTagger. Certains analyseurs ne fournissent pas le lemme en même temps que l'analyse morpho-syntaxique : dans ce cas, il faut faire appel à un module spécifique pour la lemmatisation.

Etiquetage terminologique Ce module vise à repérer les expressions du domaine qui ne sont pas des entités nommées, comme *gene expression* ou *spore coat cell* dans le domaine de la biologie. Des listes de termes certifiées peuvent être utilisées pour améliorer l'analyse, comme Gene Ontology (Consortium, 2000), le MeSH (MeSH, 1998) ou UMLS (National Library of Medicine, 2003). L'analyse morpho-syntaxique et la lemmatisation du texte est nécessaire pour procéder à l'analyse terminologique.

Analyse syntaxique L'analyse syntaxique vise à produire, pour chaque phrase du texte, un graphe reflétant les dépendances entre mots au sein de la phrase. La plupart des analyseurs n'exigent pas une analyse terminologique préalable mais celle-ci permet de réduire l'ambiguïté et donc la complexité de l'analyse (Aubin et al., 2005). L'analyse syntaxique demande encore aujourd'hui des temps de calcul beaucoup plus importants que les autres étapes d'analyse, dans la mesure où elle opère sur un espace de recherche très vaste (tous les mots de la phrase peuvent potentiellement être reliés entre eux). L'outil choisi est la grammaire de dépendance de Sleator et Temperley (1993).

[5] `http ://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/`

Coréférence Deux entités textuelles sont en relation de coréférence lorsque la première entité réfère à la seconde (*B. Subtilis... it...*). Le module tente de résoudre automatiquement les coréférences en retouvant les antécédents des pronoms (Weissenbacher et Nazarenko, 2007). Il se limite au type de coréférence le plus simple et le mieux connu (Mitkov, 1999), l'utilisation anaphorique du pronom *'it'*. Ce module peut exploiter n'importe quelle annotation préalablement apposée sur le document.

5 Processus d'adaptation

Dans les sections précédentes, nous avons décrit la plate-forme Ogmios dans ses grandes lignes. Dans la perspective du développement de moteurs de recherche spécialisés, il faut enrichir les documents d'annotations sémantiques adaptées à chaque domaine. Cela suppose une analyse précise du contenu des documents qui dépend à la fois des connaissances du domaine concerné et des spécificités du sous-langage enployé. Cette section présente les stratégies que nous avons mises en place pour adapter la plate-forme à un nouveau domaine et elle décrit en détail l'adaptation de l'analyseur syntaxique, le module de traitement le plus complexe.

5.1 Les stratégies d'adaptation

Adapter un outil générique pour un type de langue et un domaine particuliers, constitue un défi. On peut évidemment développer des outils adaptés à des sous-langages spécifiques mais l'entreprise paraît vaine : il y autant de sous-langages que de domaines et de genres de textes, voire de communautés de locuteurs. Nous avons adopté une stratégie alternative qui consiste à dériver des outils spécialisés à partir d'outils génériques, par des procédures d'adaptation, en veillant à ce que ce processus d'adaptation soit le plus automatique et le plus reproductible possible (Nazarenko, 2004). L'avantage de cette approche est évident lorsqu'on prévoit des adaptations à plusieurs domaines et/ou à plusieurs sous-langages différents. C'est une stratégie naturelle dans le cadre du projet ALVIS qui vise à développer les outils logiciels permettant de construire à moindre coût des moteurs de recherches sémantiques spécialisés.

Le processus d'adaptation que nous avons mis en place repose sur une distinction claire entre des outils d'analyse qui doivent être le plus générique et polyvalent possible et des ressources spécialisées qui décrivent un domaine de connaissances et un sous-langage particuliers. Même si cette approche est très classique en soi (on sait par exemple que la reconnaissance d'entités nommées nécessite des dictionnaires qui recensent les noms d'entités pour un domaine d'activité donné, comme par exemple des noms d'entreprises ou de médicaments), sa mise en oeuvre est délicate.

L'acquisition de ressources La première difficulté vient du manque de ressources spécialisées. Elles font souvent défaut, ou bien elles n'ont pas la couverture ou la granularité de description souhaitées, ou bien elles sont en partie périmées faute d'avoir été mises à jour régulièrement. Adapter les traitements linguistiques nécessite donc de définir des procédures d'acquisition automatique ou semi-automatique des ressources nécessaires. C'est l'approche que nous avions proposée dans (Alphonse et al., 2004). La démarche est la suivante :

– Dans un premier temps, on construit un sous-corpus représentatif des données textuelles à analyser ;

– Ce corpus est ensuite exploité comme corpus d'acquisition pour construire des ressources spécialisées : selon les cas, il peut s'agir de dictionnaires d'entités nommées, de terminologie, d'une base de règles d'extraction ou d'une ontologie. Cette acquisition peut nécessiter une analyse riche du sous-corpus initial (une analyse syntaxique complète par exemple) mais celle-ci se limite au sous-corpus d'acquisition.
– Les ressources ainsi construites sont enfin utilisées pour analyser de larges collections de documents : à cette étape la qualité des ressources utilisées compensent pour partie la faible profondeur d'analyse imposée par le volume des données à traiter.

C'est la démarche que nous suivons par exemple pour construire des terminologies spécialisées. Nous appliquons un extracteur de termes (Aubin et Hamon, 2006) sur le corpus d'acquisition pour produire la terminologie qui est ensuite utilisée par l'étiqueteur terminologique pour annoter un flux important de documents.

L'entraînement des outils Une variante de l'approche précédente consiste à utiliser le corpus d'acquisition pour entraîner un outil d'analyse plutôt que pour construire de nouvelles ressources mais cela suppose de mettre en oeuvre des méthodes d'apprentissage, le plus souvent supervisées, et donc de construire un corpus d'entraînement annoté, par nature difficile à produire. C'est néanmoins l'approche que nous utilisons pour adapter le module de résolution d'anaphore aux spécificités de certains sous-langages (Weissenbacher, 2007) ou lorsque nous remplaçons un étiqueteur morpho-syntaxique générique comme le TreeTagger par un étiqueteur entraîné sur des textes de biologie (GENIA Tagger).

La modularisation des traitements Distinguer clairement les processus d'analyse et les ressources qu'ils utilisent suppose également de décomposer l'analyse en étapes indépendantes. C'est ainsi que nous avons été amenés à dissocier la phase d'étiquetage à proprement parler du TreeTagger et la phase préliminaire de segmentation qui nécessite de prendre en compte l'étiquetage préalable des entités nommées. De la même manière, comme nous le montrons ci-dessous, distinguer clairement l'étiquetage morpho-syntaxique, la reconnaissances des termes complexes (étiquetage terminologique) et l'analyse syntaxique améliore globalement la qualité de l'analyse.

5.2 Le cas de l'analyse syntaxique

Nous illustrons ce processus d'adaptation à travers le cas de l'analyse syntaxique, qui constitue certainement l'étape la plus complexe de la plate-forme Ogmios. L'analyse syntaxique complète des textes est en effet coûteuse en temps de calcul en comparaison d'autres tâches comme l'identification des entités nommées ou l'étiquetage morpho-syntaxique.

Actuellement, il n'est pas possible de produire à la volée une analyse syntaxique des textes sélectionnés par un moteur de recherche. Préanalyser une large collection de documents a également un coût, souvent considéré comme prohibitif. De plus, il n'est pas avéré que l'analyse syntaxique profonde et complète des documents permette l'obtention de meilleurs résultats par rapport à l'exploitation traditionnelle de mots-clés. L'analyse syntaxique est cependant utile dans la phase d'acquisition (par exemple pour la construction de classes sémantiques à partir d'une analyse distributionnelle) ou pour les moteurs qui offrent des fonctionnalités de

recherche avancées, comme des requêtes relationnelles nécessitant une analyse syntaxique ciblée des passages de texte pertinents (Alphonse et al., 2004). L'analyse syntaxique permet en effet l'extraction et la formalisation de relations exprimées dans le texte entre des unités textuelles. On s'intéresse généralement en priorité aux relations entre entités nommées ou termes, relations qui sont exprimées par des verbes ou des noms prédicatifs (par ex. « le médicament X est préconisé pour les maladies de type Y »). Repérer ces relations permet l'indexation à l'aide d'informations structurées et ouvre la voie à de véritables moteurs de recherche sémantiques.

Dans le cadre du projet ALVIS, nous avons choisi d'utiliser le Link Grammar Parser[6] (LGP), un analyseur syntaxique symbolique (Sleator et Temperley, 1991) qui représente la structure syntaxique des phrases sous la forme d'un graphe de dépendances (*i.e.* relations syntaxiques entre paires de mots). LGP présente plusieurs avantages : conçu comme un analyseur robuste, il fournit des bribes de résultats si l'analyse complète de la phrase d'entrée échoue ; des tests comparatifs ont montré la supériorité de la qualité des analyses de LGP sur celles d'autres analyseurs, au moins sur un corpus d'évaluation constitué de résumés d'articles scientifiques (Aubin et al., 2005) et plus largement la supériorité des analyseurs à base de dépendances (Ding et al., 2003) ; enfin, le code de LGP est suffisamment ouvert pour qu'on puisse ajouter de nouvelles ressources.

Les tests préliminaires destinés au choix de l'analyseur ont également montré la nécessité de procédures d'adaptation au domaine et au sous-langage étudiés. On a étudié un échantillon composé de 212 phrases extraites aléatoirement d'un corpus de résumés Medline[7] traitant de la transcription chez la bactérie modèle *Bacillus subtilis*. On constate que les phrases des résumés Medline sont souvent longues et complexes[8], le lexique est très technique, les notations scientifiques sont nombreuses et variées. De nombreuses phrases présentent des constructions agrammaticales puisque les résumés Medline peuvent être rédigés par des personnes dont la langue maternelle n'est pas l'anglais. Chacune de ces caractéristiques a un impact sur la qualité de l'analyse syntaxique. Une expérience (Mollá et al., 2000) a par exemple montré que 76% des 2 781 phrases d'un corpus de manuels Unix étaient analysés complètement par LGP (sans considération de la qualité) alors que nous m'atteignons que 54% sur notre corpus de test. Ceci montre que le degré de spécificité par rapport à la langue générale varie d'un domaine à l'autre et qu'il est particulièrement important pour la génomique.

A partir des résultats obtenus sur le corpus de test, nous avons identifié les phénomènes linguistiques qui dégradent les performances de LGP. Les paragraphes suivants décrivent les mécanismes que nous avons mis en place pour y remédier. L'analyseur a été testé sur le corpus TRANSCRIPT (438 385 mots) composé de de résumés Medline traitant de la transcription chez la bactérie modèle *Bacillus subtilis*.

Elimination du bruit textuel Les textes scientifiques présentent des particularités typographiques et stylistiques que nous choisissons de traiter à l'aide d'un module de normalisation appliqué avant l'analyse syntaxique. La segmentation en mots, normalement gérée par LGP, a été externalisée et associée à la segmentation en phrases (LGP attend un corpus avec une phrase par ligne). Les deux types de segmentation ont été enrichis par l'exploitation des entités nommées identifiées préalablement dans la chaîne de traitement. Au-delà des entités nommées

[6] `http ://www.link.cs.cmu.edu/link/`
[7] `http ://www.ncbi.nlm.nih.gov/entrez/query.fcgi`
[8] Les phrases contiennent en moyenne 25.4 mots (de 8 à 59).

proprement biologiques, ce module a également vocation à repérer les abrévations ou les références bibliographiques qui sont nombreuses.

Traitement des mots inconnus L'existence de mots inconnus (termes techniques, noms propres, néologisme) dégradent fortement les performances des outils d'analyse (Cartoni, 2006) et en particulier celles de LGP. Ils sont source d'ambiguïté, ce qui augmente considérablement les temps d'analyse. Pour le corpus TRANSCRIPT, nous avons calculé la couverture du dictionnaire de LGP qui contient 60 000 entrées (*i.e.* formes fléchies). Les 438 385 occurrences de mots de TRANSCRIPT relèvent de 20 564 formes distinctes. Le dictionnaire de LGP couvre plus de 78% des occurrences, mais seulement 25% des formes du corpus. Une étude comparable décrite dans (Szolovits, 2003) a montré que LGP ne couvrait que 38% du lexique d'un corpus médical. L'écart entre la couverture des occurrences et celle des formes fléchies indique qu'un grand nombre de mots inconnus du dictionnaire de LGP sont peu fréquents. Ces formes inconnues sont principalement des entités biologiques (noms de gènes, protéines, substances chimiques, séquences d'ADN, etc.), des nombres et formules mathématiques, des mots étrangers (le plus souvent latins), des termes techniques ou encore des mots mal orthographiés.

Nous avons étudié et comparé trois méthodes différentes pour la prise en charge de ces mots inconnus (Pyysalo et al., 2006)[9] :

- L'extension du lexique est la méthode la plus classique pour la résolution des lacunes lexicales. Il s'agit d'ajouter des ressources spécialisés à un outil générique, la première des procédures d'adaptation mentionnée ci-dessus. Toutefois, la nature et la basse fréquence des mots inconnus de notre corpus fait qu'il est difficile d'obtenir un dictionnaire avec une bonne couverture : en dépit de sa taille, l'apport du vocabulaire du Specialist Lexicon[10] de l'UMLS (Pyysalo et al., 2004) est décevant (Pyysalo et al., 2006).
- Une autre approche consiste à prendre en compte une base de règles morphologiques. Nous avons pour cela exploité le module de *morpho-guessing* (MG) de LGP qui permet de prédire la classe de certains mots (*i.e.* leur comportement syntaxique) à partir de leurs caractéristiques morphologiques (*e.g.* leur suffixe, voir (Grover et al., 2004)). Dans la version d'origine, le module MG exploitait quatre règles basées sur les suffixes. Nous avons ajouté vingt-trois nouvelles règles pour les mots les plus fréquents qui présentent des caractéristiques suffixales exploitables : des noms communs (suffixés en *-ase*, *-ity*, *-ol*, *-in*, etc), des adjectifs (*-al*, *-ous*, etc.), des mots d'origine latine de fonction adjectivale (*-us*, *-ae*, etc.). Cette stratégie d'adaptation repose sur une modularisation accrue de la chaîne de traitement (on dissocie les analyses morphologique et syntaxique) qui permet l'exploitation de connaissances morphologiques spécialisées.
- La troisième stratégie consiste à exploiter dans LGP l'information calculée lors de l'étiquetage morpho-syntaxique. Cette stratégie d'adaptation est décrite et évaluée dans (Pyysalo et al., 2006) : elle permet d'éliminer presque totalement les problèmes liés aux mots inconnus. L'utilisation du GENIA Tagger, entraîné sur des textes du domaine biologique, permet une réduction significative du taux d'erreur de l'analyse.

[9] La version du Link Grammar Parser adapté à la biologie (*biolg*) incluant l'extension du module de morpho-guessing et l'exploitation de l'étiquetage morpho-syntaxique est disponible sur `http ://www.it.utu.fi/biolg/`.

[10] `http ://groups.csail.mit.edu/medg/projects/text/lexicon.html`

Traitement des ambiguïtés structurelles Au-delà de l'ambiguïté catégorielle des mots inconnus, de nombreuses phrases ont des structures ambiguës qui se caractérisent par le fait qu'un mot ou un syntagme a deux points de rattachement possibles. C'est souvent le cas des groupes prépositionnels : dans les phrases *manger une glace à la fraise/à la terrasse d'un café*, le complément en *à* peut se rattacher soit au verbe soit au complément d'objet.

Ce problème de rattachement est généralement traité à l'aide d'informations statistiques calculées dans le texte même (Hindle et Rooth, 1993), dans d'autres corpus (Bourigault et Frérot, 2004), sur le web (Volk, 2002) ou à partir de ressources externes comme WordNet (Stetina et Nagao, 1997), mais aucun mécanisme ne permet l'exploitation de telles informations dans LGP.

Nous proposons dans ce cas une stratégie originale qui consiste à mieux exploiter la complémentarité entre les différents modules : nous utilisons les résultats de l'étiquetage terminologique pour simplifier la phrase ; les termes sont identifiés préalablement à l'analyse syntaxique et sont marqués dans le texte avec leur analyse interne. Lors du traitement d'une phrase, LGP ne considère que la tête syntaxique des termes marqués. Les autres mots contenus dans le terme sont ignorés. L'analyse interne des termes telle qu'elle est fournie lors de l'étiquetage terminologique est ensuite intégrée à l'analyse de la phrase. Si cette intégration échoue, les termes sont ignorés et LGP procède à une analyse complète de la phrase. Cette stratégie est en cours d'évaluation mais les résultats préliminaires de (Aubin et al., 2005) montrent qu'elle a un impact significatif sur les temps de traitement.

Prise en compte de constructions particulières Certains mots de la langue générale, définis dans le lexique de LGP, présentent un comportement syntaxique spécifique lorsqu'ils sont employés dans un contexte biomédical. Certains verbes, d'ordinaire transitifs, deviennent intransitifs (*e.g. "initiate"*) ; d'autres qui sont transitifs directs apparaissent également comme transitifs indirects (*e.g. "code"*, *"code for"*). Pour traiter de telles constructions, il faut retoucher le lexique et la grammaire de LGP.

On touche ici aux limites du processus d'adaptation tel que nous l'avons défini plus haut.

6 Analyse des performances

La plate-forme Ogmios vise à analyser des textes provenants du web pour des moteurs spécialisés dans certains domaines techniques. Comme nous l'avons souligné plus haut, cela implique certaines contraintes de robustesse. Même s'il ne s'agit pas d'analyser les documents « en temps réel » au moment de la requête de l'utilisateur, les performances doivent malgré tout être compatibles avec le flux de documents récoltés par le moissonneur : il faut pouvoir analyser plusieurs giga-octets de données par jour. Ce type de performances implique une architecture distribuée, qui permet d'ajouter de nouvelles machines en fonction de la charge (à condition bien entendu de disposer de la force de calcul nécessaire). Par ailleurs, comme nous l'avons souligné plus haut, le système doit être robuste face aux documents fournis en entrée, qui peuvent être très variables quant à leur taille ou leur contenu.

Une expérience d'annotation massive a été menée sur un ensemble de 55 329 documents collectés par un moissonneur spécialisé dans le domaine de la biologie (le corpus BIO). La figure 2 montre la distribution de la taille des documents en entrée (les deux axes ont une

échelle logarithmique). La plupart des documents sont compris entre 1 kilo-octet et 100 kilo-octets. La taille du plus grand document est 5,7 méga-octets. L'objectif était d'enrichir ces documents d'entités nommées, d'étiquettes morpho-syntaxiques et de termes.

Une autre expérience est menée en parallèle : il s'agit cette fois d'appliquer la chaîne complète de traitement – y compris l'analyse syntaxique et la résolution des anaphores – sur un petit corpus. Elle n'est pas développée ici.

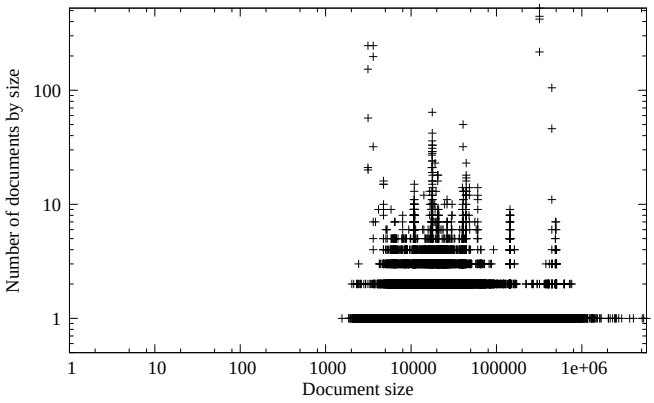

FIG. 2 – *distribution de la taille des documents en entrée*

Nous avons utilisé 20 ordinateurs pour annoter massivement les documents. Il s'agit d'ordianteurs classiques de type PC avec 1 ou 2 giga-octet de mémoire vive (RAM) et un processeur cadencé à 2,9 ou 3,1 GHz.Nous avons également utilisé un ordinateur avec 8 giga-octets de RAM et deux processeurs Xeon cadencés à 2,8 GHz (*dual core xeon processor*). Le système d'exploitation utilisé est Linux (Debian ou Mandrake linux). Nous avons utilisé la dernière version de la plate-forme Ogmios, disponible sous forme de modules CPAN [11]). Le serveur et trois clients étaient hébergés sur la machine Xeon à bi-processeur. Chaque ordinateur personnel abritait une seule instance de la chaîne de traitement, qui tournait en tâche de fond avec une priorité faible.

Nous nous sommes placés dans le contexte d'annotation d'un flux de documents venant du Web et nous avons réalisé l'ensemble des traitements jusqu'à l'étiquetage terminologique. Nous avons exploité une liste d'environs 400 000 entités nommées, incluant des noms d'espèce et de gènes ainsi qu'une liste de 375 000 termes issus du MeSH et de Gene Ontology.

Les performances obtenues donnent une bonne idée des performances globales de la plate-forme (une évaluation complète aurait demandé des séries plus importantes de test). Le temps d'exécution de chaque module a été enregistré à l'aide du module Perl `Time::Hires`. Les temps d'analyse sont inscrits dans le fichier XML produit en sortie.

L'annotation de la collection a été effectuée en 35 heures. Le tableau 1 montre le nombre total d'entités (c'est-à-dire d'objets XML) trouvé au sein de la collection de documents. Les corpus est composé de 106 millions de mots et 4,72 millions de phrases. 176 documents ne

[11] `http ://search.cpan.org/ thhamon/Alvis-NLPPlatform-0.3/`

contenaient aucun mot et n'ont pas été analysés au-delà de l'étape de tokenisation. Un des clients a analysé un document composé de 414 995 mots.

Les documents du corpus BIO sont analysés en 35 secondes, en moyenne. La génération du fichier XML prend en moyenne 2 secondes supplémentaires. La table 2 montre les temps moyens d'analyse pour chaque module. Les étapes les plus coûteuses en temps de traitement sont celles qui demandent le plus de ressources, à savoir la reconnaissance des termes (56 % du temps de traitement global) et la reconnaissance des entités nommées (16 % du total).

TAB. 1 – *Nombres d'éléments analysés et moyenne par document*

	Nombre moyen d'éléments par document	Nombre total d'éléments dans la colection de documents
Tokens	5 021,9	277 846 470
Entités nommées	81,88	4 530 368
Mots	1 912,65	105 821 243
Phrases	85,41	4 726 003
Morpho-syntaxe et lemmatisation	1883,5	104 208 536
Termes	250,76	13 874 089

TAB. 2 – *Temps moyen de traitement par document, en secondes*

	Temps de traitement moyen	Percentage
Chargement du document XML en entrée	0,38	1,02
Tokenization	0,7	1,88
Reconnaissance des entités nommées	6,12	16,42
Segmentation en mots	5,19	13,92
Segmentation en phrases	0,18	0,48
Analyse morpho-syntaxique et lemmatisation	1,84	4,94
Reconnaissance des termes	20,83	55,89
Restauration du document XML annoté	2,03	5,45
Total	37,27	100

La collection de documents a pu être analysée sans problème. En effet, grace à la distribution des traitements, l'impossibilité de traiter convenablement deux documents n'a pas eu de conséquences importantes sur l'ensemble de l'annotation. Les deux clients en charge de l'analyse de ces documents ont simplement dû être redémarrés. Cependant, au cours du développement, nous avions noté que certains outils avaient des difficultés à analyser les documents UTF-8, ceci dépendant étroitement de la configuration de la machine et de l'environnement d'exploitation.

Les performances obtenues montrent que la plate-forme développée est robuste et qu'elle peut traiter de grandes masses de textes dans des temps raisonnables. Cette analyse permet une indexation précise de documents spécialisés.

7 Conclusion

Nous avons présenté une plate-forme destinée à l'annotation sémantique de collections spécialisées de documents issus du web.

L'architecture proposée est générique mais peut être adaptée pour permettre l'analyse de documents spécialisés, sans perdre en qualité d'analyse. Nous avons montré que cette procédure d'adaptation repose à la fois sur l'acquisition de ressources sémantiques à partir d'un corpus échantillon, sur l'entraînement des outils d'analyse par des techniques d'apprentissage et sur la décomposition de l'analyse en étapes élémentaires complémentaires. Les expériences rapportées ici portent sur une collection de documents constituée par un moissonneur spécialisé dans le domaine de la biologie. D'autres expériences sont en cours dans le domaine des bibliothèques numériques (*digital libraries*).

La stratégie adoptée consiste à réutiliser des modules existants et à les adapter au domaine visé. Ces modules peuvent bien évidemment être remplacés par d'autres et les traitements peuvent être enchaînés de différentes façons, en fonction du résultat visé. Les modules intégrés sont pour l'instant la reconnaissance des entités nommées, la segmentation en phrases et en mots, l'analyse morpho-syntaxique, la lemmatisation et la reconnaissance des termes techniques. Nous travaillons actuellement à une meilleure intégration de l'analyseur syntaxique, prenant complètement en compte l'analyse terminologique. Nous intégrerons ensuite le module de résolution des anaphores.

Nous avons enfin fourni une analyse des performances, qui est le point clé de ce type d'application. Nous avons montré une implantation distribuée de la plate-forme, afin de permettre le traitement d'une collection de documents sur plusieurs machines. Cette stratégie a permis d'obtenir des temps de calcul acceptables pour la tâche.

8 Remerciements

Ce travail a été pour l'essentiel effectué dans le cadre du projet ALVIS (projet IST-1-002068-STP du 6ème programme cadre européen). Les données et les exemples fournis ont été obtenus en interaction avec les partenaires du projet, notamment l'unité MIG de l'INRA pour tout ce qui concerne les expériences sur la biologie.

Références

Alphonse, E., S. Aubin, P. Bessieres, G. Bisson, T. Hamon, S. Laguarrigue, A. P. Manine, A. Nazarenko, C. Nedellec, M. O. A. Vetah, T. Poibeau, et D. Weissenbacher (2004). Event-based information extraction for the biomedical domain : the caderige project. In *Workshop BioNLP (Biology and Natural language Processing), Conférence Computational Linguisitics (Coling 2004)*, Geneva.

Aubin, S. et T. Hamon (2006). Improving Term Extraction with terminological resources. In T. Salakoski et al. (Eds.), *Advances in Natural Language Processing (FinTAL 2006)*, LNAI 4139, pp. 380–387. Springer.

Aubin, S., A. Nazarenko, et C. Nédellec (2005). Adapting a general parser to a sublanguage. In *The international conference RANLP 2005*, Borovets, Bulgaria.

Berroyer, J.-F. et T. Poibeau (2004). TagEN, un analyseur d'entités nommées. LIPN Internal Report, Université Paris-Nord.

Bontcheva, K., V. Tablan, D. Maynard, et H. Cunningham (2004). Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering 10*(3-4), 349–374.

Bourigault, D. et C. Frérot (2004). Ambiguïté de rattachement prépositionnel : introduction de ressources exogènes de sous-catégorisation dans un analyseur syntaxique de corpus endogène. In *Actes des 11èmes journées sur le Traitement Automatique des Langues Naturelles, Fès, Maroc*.

Cartoni, B. (2006). Constance et variabilité de l'incomplétude lexicale. In *Actes de RECITAL*, Louvain, pp. 661–669.

Consortium, T. G. O. (2000). Gene ontology : tool for the unification of biology. *Nature genetics 25*, 25–29.

Consortium, T. G. O. (2001). Creating the Gene Ontology Resource : Design and Implementation. *Genome Res. 11*(8), 1425–1433.

Cunningham, H., K. Bontcheva, V. Tablan, et Y. Wilks (2000). Software infrastructure for language resources : a taxonomy of previous work and a requirements analysis. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2)*, Athens.

Ding, J., D. Berleant, J. Xu, et A. W. Fulmer (2003). Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser. In *15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, pp. 467–471.

Ferrucci, D. et A. Lally (2004). UIMA : an architecture approach to unstructured information processing in a corporate research environment. *Natural Language Engineering 10*(3-4), 327–348.

Grefenstette, G. (1994). *Exploration in Automatic Thesaurus Discovery*. Boston, USA : Kluwer Academic Publishers.

Grishman, R. (1997). Tipster architecture design document version 2.3. Technical report, DARPA.

Grover, C., M. Lapata, et A. Lascarides (2004). A Comparison of Parsing Technologies for the Biomedical Domain. *Natural Language Engineering 11*(1), 27–66.

Hindle, D. et M. Rooth (1993). Structural Ambiguity and Lexical Relations. In *Meeting of the Association for Computational Linguistics*, pp. 229–236.

MeSH (1998). Medical subject headings. WWW page `http ://www.nlm.nih.gov/mesh/ meshhome.html`, Library of Medicine, Bethesda, Maryland.

Mitkov, R. (1999). Anaphora resolution : The state of the art. Technical report, University of Wolverhampton.

Mollá, D., G. Schneider, R. Schwitter, et M. Hess (2000). Answer Extraction Using a Dependency Grammar in ExtrAns. *Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammars*, 145–178.

Müller, H.-M., E. E. Kenny, et P. W. Sternberg (2004). Textpresso : an ontology-based in-

formation retrieval and extraction system for biological literature. *PLoS Biology 2*(11), 1984–1998.

National Library of Medicine (Ed.) (2003). *UMLS Knowledge Source* (13^{th} ed.).

Nazarenko, A. (2004). *Donner accès au contenu des documents textuels : Acquisition de connaissances et analyse de corpus spŐcialisés*. Habilitation Ĺ diriger des recherches, UniversitŐé Paris 13, Villetaneuse.

Nazarenko, A., E. Alphonse, J. Derivière, T. Hamon, G. Vauvert, et D. Weissenbacher (2006). The alvis format for linguistically annotated documents. Alvis project deliverable, Université Paris-Nord.

Neff, M. S., R. J. Byrd, et B. K. Boguraev (2004). The talent system : Textract architecture and data model. *Natural Language Engineering 10*(3-4), 307–326.

Popov, B., A. Kiryakov, D. Ognyanoff, D. Manov, et A. Kirilov (2004). Kim – a semantic platform for information extraction and retrieval. *Natural Language Engineering 10*(3-4), 375–392.

Pyysalo, S., T. S. S. Aubin, et A. Nazarenko (2006). Lexical adaptation of link grammar to the biomedical sublanguage : a comparative evaluation of three approaches. In J. F. Sophia Ananiadou (Ed.), *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*, Jena, Germany, pp. 60–67.

Pyysalo, S., F. Ginter, T. Pahikkala, J. Boberg, J. Järvinen, T. Salakoski, et J. Koivula (2004). Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions. In *Proceedings of the international Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pp. 15–21.

Schmid, H. (1997). Probabilistic part-of-speech tagging using decision trees. In D. Jones et H. Somers (Eds.), *New Methods in Language Processing Studies in Computational Linguistics*.

Sleator, D. et D. Temperley (1991). Parsing English with a Link Grammar. Technical report, Carnegie Mellon University.

Sleator, D. D. et D. Temperley (1993). Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*.

Stetina, J. et M. Nagao (1997). Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In J. Zhou et K. W. Church (Eds.), *Proceedings of the Fifth Workshop on Very large Corpora*, Beijing, China, pp. 66–80.

Szolovits, P. (2003). Adding a Medical Lexicon to an English Parser. In *AMIA 2003 Annual Symposium*, pp. 639–643.

Volk, M. (2002). Using the Web as Corpus for Linguistic Research. In R. Pajusalu et T. Hennoste (Eds.), *Tähendusepüüdja. Catcher of the Meaning. A Festschrift for Professor Haldur Õim*. Estonia : Publications of the Department of General Linguistics 3. University of Tartu.

Weissenbacher, D. (2007). *Revue d'Intelligence Artificielle - Modèles Graphiques Probabilistes*, Chapter Les réseaux bayésiens : un formalisme adapté au Traitement automatique des langues ? Lavoisier.

Weissenbacher, D. et A. Nazarenko (2007). A bayesian classifier for the recognition of the impersonal occurrences of the *it* pronoun. In *Proceedings of Discourse Anaphora and Anaphor*

Une infrastructure pour l'annotation linguistique de documents

Resolution Colloquium'07.

Summary

This paper focuses on the design of a text processing architecture exploiting NLP tools to produce linguistically annotated web documents. The originality of this architecture is twofold: it can be tuned to specific domains and it is robust enough to process crawled document collections, which are usually quite heterogeneous. Taking as an example the biological domain, we show how the Ogmios platform can be adapted to biology and we detail performance obtained on a large collection of specialized documents. The result of the analysis is then taken into account by specialized search engines.