24 novembre 2006

Outilex. Présentation synthétique des résultats

Éric Laporte, Olivier Blanc, Matthieu Constant
IGM - Université de Marne-la-Vallée
5, bd Descartes - 77454 Marne-la-Vallée CEDEX 2
eric.laporte@univ-mlv.fr, olivier.blanc@univ-mlv.fr,
matthieu.constant@univ-mlv.fr

Résumé La plate-forme logicielle Outilex, qui est mise à la disposition de la recherche, du développement et de l'industrie, comporte des composants logiciels qui effectuent toutes les opérations fondamentales du traitement automatique du texte écrit : traitements sans lexiques, exploitation de lexiques et de grammaires, gestion de ressources linguistiques. Les données manipulées sont structurées dans des formats XML, et également dans d'autres formats plus compacts, soit lisibles soit binaires, lorsque cela est nécessaire ; les convertisseurs de formats nécessaires sont inclus dans la plate-forme ; les formats de grammaires permettent de combiner des méthodes statistiques avec des méthodes fondées sur des ressources linguistiques. Enfin, des lexiques du français et de l'anglais issus du LADL, construits manuellement et d'une couverture substantielle sont distribuées avec la plate-forme sous licence LGPL-LR.

1 Introduction

Le projet Outilex a été financé par le Ministère de l'Industrie dans le cadre du Réseau national des technologies logicielles (RNTL). Il visait à mettre à la disposition de la recherche, du développement et de l'industrie une plate-forme logicielle de traitement des langues naturelles ouverte et compatible avec l'utilisation d'XML, d'automates finis et de ressources linguistiques. Ce rapport présente de manière synthétique les principaux résultats d'Outilex.

Le projet Outilex a regroupé 10 partenaires français, dont 4 académiques, 3 PME et 3 grands groupes industriels. Il était coordonné par l'IGM. Préparé sous la direction de Maurice Gross, il a été lancé en 2002 et s'est terminé en 2006.

Les méthodes de traitement des langues naturelles sont encore aujourd'hui, la plupart du temps, mises en oeuvre par des logiciels dont la diffusion est limitée. De plus, on dispose rarement de formats d'échange ou de convertisseurs de formats qui permettraient de combiner plusieurs composants logiciels pour un même traitement. Quelques plates-formes font exception à cette situation générale, mais aucune n'est satisfaisante. Intex (Silberztein, 1993), FSM (Mohri *et al.*, 1998) et Xelda[1] sont fermés au développement collaboratif. Unitex (Paumier, 2003), inspiré d'Intex mais dont le code source est pour la quasi-totalité sous licence LGPL, ne fournit pas de formats XML. Les systèmes NLTK (Loper & Bird, 2002) et Gate (Cunningham, 2002) n'ont pas de fonctionnalités de gestion de ressources lexicales.

La plate-forme Outilex sera prochainement distribuée pour un coût annuel très modique (100

[1] http ://www.dcs.shef.ac.uk/ hamish/dalr/baslow/xelda.pdf.

euros). Le code source des logiciels est sous licence LGPL[2]. Les ressources linguistiques, qui sont des lexiques du français et de l'anglais issus du LADL, construits manuellement et d'une couverture substantielle, sont sous licence LGPL-LR (voir section 2.2). Outre le développement de la plate-forme, placée pour l'essentiel sous la responsabilité des partenaires IGM et Systran, le projet comportait la réalisation de démonstrateurs propriétaires. En raison de son ambition internationale, Outilex a également participé aux efforts actuels de définition de normes en matière de modèles de ressources linguistiques.

Ce rapport est organisé en deux parties. La première décrit la plate-forme elle-même : ses fonctionnalités, les aspects liés à la normalisation, sa diffusion. La deuxième partie est consacrée aux démonstrateurs réalisés dans le cadre du projet. Le bilan et les perspectives sont présentés en conclusion.

2 La plate-forme Outilex

Les modules d'Outilex sont intégrés sous la forme d'une interface graphique programmée en Java qui appelle des programmes en C++. Tous les formats de données représentant des textes, des lexiques ou des grammaires sont en XML ou convertibles en un format XML. Cela permet d'importer et d'exporter les données depuis ou vers d'autres environnements, et assure l'interopérabilité d'Outilex avec les autres systèmes existants.

Les différents modules qui composent Outilex ont d'abord été testés isolément par leurs auteurs. Les tests d'intégration et de déploiement ont été effectués par les partenaires qui devaient proposer des démonstrateurs, avec leurs données applicatives. Les retours de ces tests ont permis d'améliorer la plate-forme sur trois plans :
– l'interface graphique utilisateur,
– la portabilité (Windows XP et Linux avec processeur Intel),
– l'interface de programmation d'application (API), qui n'avait pas encore pu être réalisée mais qui s'est avérée indispensable lorsque le traitement porte sur des documents nombreux.

La plate-forme permet trois types de traitement : sans lexiques, par lexiques, et par grammaires. De plus, elle permet à l'utilisateur de gérer des ressources linguistiques. Dans la suite, nous décrivons les fonctionnalités correspondant à ces quatre points, puis les aspects liés à l'ergonomie de l'interface, nos activités de normalisation, et enfin la diffusion de la plate-forme. La plate-forme est documentée plus en détail dans les différents rapports de l'IGM, notamment le rapport final (M. Constant, novembre 2006), ainsi que dans le guide de l'utilisateur (Blanc & Constant, 2006b).

2.1 Traitement sans lexiques

Nous regroupons dans cette partie les opérations qui ne font pas appel à des informations extraites de ressources lexicales. Ces opérations ont pour résultat une représentation du texte comme séquence de tokens[3]. Le module qui les met en oeuvre prend en entrée un texte brut ou HTML et il produit en sortie le texte segmenté en paragraphes, en phrases et en tokens dans

[2]Lesser General Public License, http ://www.gnu.org/copyleft/lesser.html.

[3]Nous n'avons pas inclus dans la plate-forme de traitements applicatifs opérant sur ce modèle de texte, ni sur le modèle du sac de tokens. En effet, les outils existants qui mettent en oeuvre de tels traitements sont faciles à interfacer avec Outilex, en raison de la simplicité des modèles sous-jacents.

un format XML (seg.xml) proche de celui proposé par le projet de norme ISO d'annotation morpho-syntaxique de textes (MAF), d'ailleurs élaboré avec la participation d'Outilex (cf. section 2.6.1). Les éventuelles balises de mise en page HTML sont conservées et placées dans des éléments XML qui les distinguent des données textuelles. Les règles de segmentation en tokens

<?xml version="1.0"?> <document original_format="txt"><par id="1"><tu id="s0"><token type="word" id="t1" alph="latin" case="capit">La</token> <token type="word" id="t2" alph="latin">police</token> <token type="word" id="t3" alph="latin">a</token> <token type="word" id="t4" alph="latin">saisi</token> <token type="numeric" id="t5">164</token> ... <token type="punctuation" id="t11">.</token></tu></par> </document>

FIG. 1 – Texte segmenté au format seg.xml

et en phrases sont basées sur la catégorisation des caractères définie par la norme Unicode (ex. lettres, chiffres). À chaque token est associé un certain nombre d'informations telles que son type (mot, nombre, ponctuation, etc.), son alphabet (latin, grec), sa casse (mot en minuscule, commençant par une majuscule, etc.) ainsi que d'autres informations pour les autres symboles (signe de ponctuation ouvrant ou fermant, etc.). De plus, un identifiant est associé à chaque token qui sera conservé durant toutes les phases du traitement. Par exemple, la phrase *La police a saisi 164 procès-verbaux jeudi dernier* est segmentée comme dans la figure 1. Appliqué à un corpus de dépêches AFP (352 464 tokens), ce module traite 22 185 mots par seconde[4].

Dans sa dernière version (cf. rapport final de Systran, Jean Senellart, novembre 2006), ce module permet de traiter les formats RTF et PDF en plus du HTML, et d'insérer l'enrichissement typographique dans le document après traitement, dans le même format que le document source.

2.2 Traitement par lexiques

Les traitements évoqués dans la partie précédente ont pour résultat une représentation du texte comme séquence de tokens. Une de nos motivations de départ pour créer le projet Outilex était qu'une plate-forme généraliste doit aller plus loin et intégrer certaines notions fondamentales absentes de ce modèle, comme celle de mots composés ou expressions multi-mots, ou la séparation des emplois en cas d'ambiguïté. Les produits de la linguistique de corpus seuls (Schmid, 1994) ne sont pas de nature à résoudre les problèmes posés par de telles notions. L'un des moyens pour y parvenir est l'utilisation de lexiques et de grammaires. L'utilisation de lexiques par les entreprises du domaine s'est d'ailleurs largement étendue au cours des dernières années. C'est pourquoi Outilex fournit un jeu complet de composants logiciels pour les opérations sur les lexiques. De plus, dans le cadre de sa contribution à Outilex, l'IGM a rendu publique [5] une proportion substantielle des lexiques du LADL[6] pour le français (109 912 lemmes simples et 86 337 lemmes composés) [7] et l'anglais (166 150 lemmes simples et 13 361 lemmes composés). Ces ressources sont proposées sous la licence LGPL-LR[8], créée dans le cadre d'Outilex

[4]Ce test et les tests suivants ont été effectués sur un ordinateur de bureau équipé d'un processeur Intel Pentium cadencé à 2.8 GHz et de 512 Mo de mémoire RAM.

[5]http ://www.at-lci.com/outilex, suivre Membres, puis Télécharger.

[6]Laboratoire d'automatique documentaire et linguistique, Université Paris 7, 1968-2000.

[7]Le jeu d'étiquettes pour le français combine 13 catégories morpho-syntaxiques, 18 traits flexionnels et divers traits syntaxico-sémantiques.

[8]Lesser General Public License for Language Resources, http ://infolingu.univ-mlv.fr/lgpllr.html. Les droits et devoirs donnés aux utilisateurs par la licence LGPL-LR sont l'équivalent, pour les ressources linguistiques, de ceux donnés aux utilisateurs de la licence LGPL pour les logiciels.

et agréée par la FSF[9]. Les programmes d'Outilex sont compatibles avec toutes les langues européennes à flexion par suffixes. Des extensions seront nécessaires pour les autres types de langues.

2.2.1 Formats de lexiques

L'absence de formats génériques et de documentation sur les données sont deux obstacles à l'utilisation et à l'échange de lexiques pour le traitement automatique des langues. Outilex a adopté deux formats de lexique étroitement liés au projet de norme ISO de balisage de lexiques (LMF), auquel nous avons d'ailleurs contribué (cf. section 2.6.2). D'une part, nous avons traduit en XML le format Dela et inséré dans les balises de la documentation sur les données. Le format obtenu pour les lexiques de formes fléchies, dic.xml, illustré par la fig. 2, est adapté à l'échange de données et compatible avec le modèle LMF. Le format Dela, quant à lui, est plus compact : *appelés du contingent,appelé du contingent.N+hum :mp*, et adapté à la visualisation sur écran et à la maintenance manuelle par les linguistes (Laporte, 2005). Nous avons donc réalisé des convertisseurs dans les deux sens entre ces deux formats. Il existe en outre un format opérationnel que nous décrirons dans la section 2.4.2.

```
<entry>
  <lemma>appelé du contingent</lemma>
  <pos name='noun'/>
  <feat name='subcat' value='human'/>
  <inflected>
    <form>appelés du contingent</form>
    <feat name='gender' value='masculine'/>
    <feat name='number' value='plural'/>
  </inflected>
</entry>
```

FIG. 2 – Un extrait de lexique au format dic.xml

```
<attrtype name='antepos' type='bool'>
  <true alias='g'/>
</attrtype>
<pos name='adj' cutename='A'>
  <attribute name='postpos' type='postpos' default='false' shortcut='yes'/>
  <attribute name='antepos' type='antepos' default='false' shortcut='yes'/>
  <attribute name='gender' type='gender' shortcut='yes'/>
  <attribute name='number' type='number' shortcut='yes'/>
</pos>
```

FIG. 3 – Extrait de la description d'un jeu d'étiquettes au format LingDef pour le français

En pratique, l'utilisation de lexiques permet de manipuler des jeux d'étiquettes complexes et d'une granularité fine, qui nécessitent une description formelle telle que celle de la fig. 3.

2.2.2 Consultation des lexiques

Notre étiqueteur morpho-syntaxique[10] prend un texte segmenté au format *seg.xml* en entrée et attribue à chaque forme (simple ou composée) l'ensemble des étiquettes lui correspondant extraites des lexiques indexés (cf. section 2.4.2). Il est possible d'appliquer un ensemble de lexiques à un texte dans la même passe de traitement. De plus, un système de priorités permet de bloquer des analyses issues de lexiques à faible priorité si la forme considérée est également présente dans un lexique de priorité supérieure. Ainsi, nous fournissons par défaut un lexique général proposant un grand nombre d'analyses pour la langue standard, que l'utilisateur peut, pour une application spécifique, enrichir à l'aide de lexiques complémentaires et/ou filtrer avec un lexique prioritaire. Enfin, plusieurs options, qui peuvent se combiner entre elles, permettent de paramétrer la méthode de consultation : il est possible d'ignorer complètement la casse, ou les accents et autres signes diacritiques. Ces paramètres permettent d'adapter notre étiqueteur au texte analysé (article de journal, page internet, e-mail, etc.). De plus, cette fonctionnalité

[9] Free Software Foundation, http ://www.fsf.org/ .

[10] Les modules de la chaîne de traitement ayant été implémentés de manière à être indépendants les uns des autres, il est possible de substituer un autre étiqueteur à celui proposé par défaut.

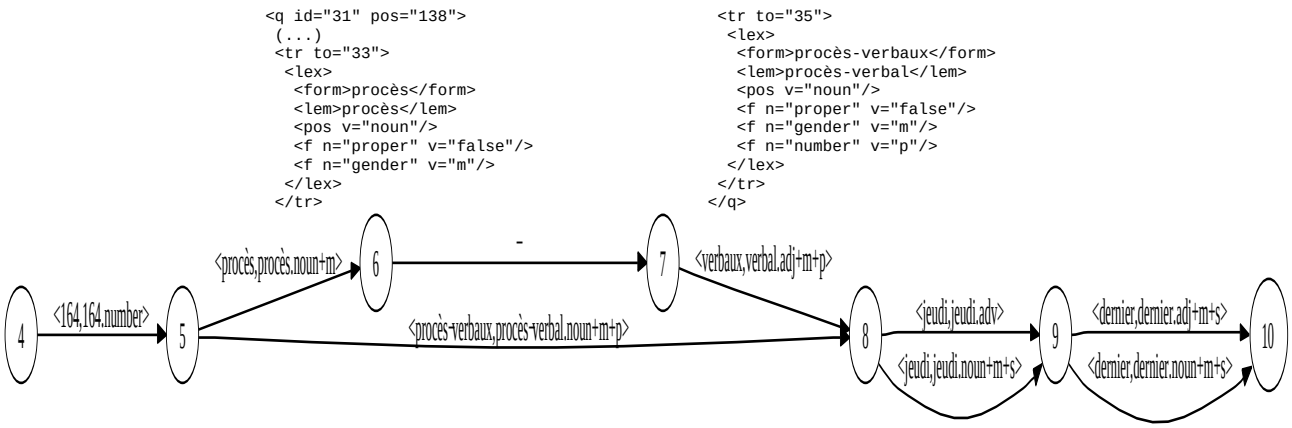

FIG. 4 – Extrait d'un automate acyclique représentant un texte étiqueté

permet d'assurer un redressage orthographique et typographique en copiant dans le texte la forme fléchie trouvée dans le lexique. Ce redressage nécessite cependant une révision humaine.

Appliqué au corpus de dépêches AFP (cf. section 2.1) avec les lexiques décrits dans la section 2.2, Outilex étiquette une moyenne de 6 650 mots par seconde[11].

2.2.3 Représentation du texte étiqueté

L'utilisation exclusive de lexiques pour étiqueter les textes produit des ambiguïtés lexicales. Le modèle le plus adapté pour représenter le texte étiqueté dans ces conditions est l'automate fini acyclique, en général appelé "treillis" dans ce contexte[12]. La figure 4 présente une partie de l'automate du texte obtenu après l'étiquetage de la phrase segmentée présentée dans la section 2.1. Ce modèle prend en compte la notion de mot, distincte de la notion de token en raison notamment des expressions multi-mots.

La plate-forme Outilex a mis au point deux formats nouveaux pour la représentation du texte étiqueté. Le premier est le format binaire de sortie de l'outil de consultation des lexiques (cf. section 2.2.2). Il permet un traitement rapide de textes de grande taille. Le deuxième, fsa.xml, est destiné à l'échange de données (fig. 4). C'est une traduction en XML du format fst2 d'Unitex. Le projet MAF propose un format voisin (cf. section 2.6.1). Des fonctionnalités d'import/export entre ces deux formats sont prévues. Un convertisseur entre le format binaire et le format fsa.xml est disponible. De plus, les deux formats peuvent être exportés vers le format dot (Gansner & North, 2000).

2.3 Traitement par grammaires

Lors de l'étiquetage de mots par lexique, les étiquettes sont assignées aux mots d'une façon indépendante du contexte. Cette procédure est généralement complétée d'une façon ou d'une autre, dans les applications, par la prise en compte de contraintes sur des séquences de mots,

[11] 4,7 % des occurrences de tokens n'ont pas été trouvées dans le lexique ; cette valeur tombe à 0,4 % si on déduit le nombre de celles qui commencent par une majuscule.

[12] Nous n'avons pas adopté ce terme dans ce rapport car un tel réseau n'est en général pas un treillis au sens mathématique du terme. Nous utilisons le terme d'automate acyclique qui lui n'est pas critiquable.

et donc de "grammaires", au sens de ressources linguistiques spécifiant formellement de telles contraintes. Les formalismes grammaticaux étant la tour de Babel du traitement des langues naturelles, la plate-forme Outilex mise sur un formalisme minimal, qu'on peut résumer en trois points :

– pour la représentation des mots et des paradigmes de mots, la notion de masque lexical (Blanc & Dister, 2004), spécification d'un ensemble de mots par un ensemble de traits ;
– pour la représentation des contraintes sur les séquences, la notion de réseau de transitions récursif (RTN), outil purement formel, dépourvu de toute notion linguistique, au même titre que les transducteurs finis ou les grammaires algébriques[13] ;
– les automates constituant les RTN peuvent être des transducteurs, c'est-à-dire comporter des sorties, utiles par exemple pour insérer des balises dans les textes et formaliser ainsi des relations entre les segments identifiés.

Ces points permettent de construire des grammaires locales au sens de (Gross, 1993; Gross, 1997). Ce formalisme est utilisé dans des situations variées : extraction d'informations (Poibeau, 2001; Nakamura, 2005), reconnaissance d'entités nommées (Krstev *et al.*, 2005), identification de structures grammaticales (Mason, 2004; Danlos, 2005)... avec pour chacune de ces applications des taux de rappel et de précision qui égalent l'état de l'art du domaine. Nous avons ajouté la possibilité de pondérer les transitions. Nous appelons le formalisme obtenu réseau de transitions récursif pondérées (WRTN). Dans la fig. 5, par exemple, le chemin de poids 1, qui

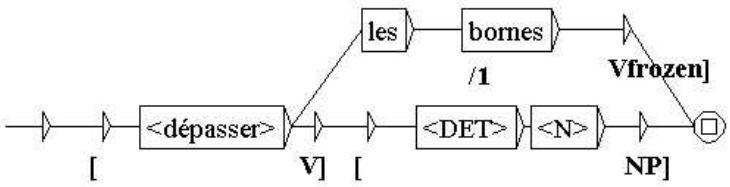

FIG. 5 – Exemple de WRTN

correspond à l'analyse avec expression figée, est prioritaire par rapport à l'autre (de poids 0 par défaut) correspondant à l'analyse compositionnelle.

Ce formalisme WRTN est conçu pour permettre la réalisation de systèmes hybrides utilisant à la fois des méthodes statistiques et des méthodes fondées sur des ressources linguistiques. Les méthodes statistiques déterminent les valeurs des poids à associer aux expressions décrites dans les grammaires. Ces méthodes étant d'une grande variété, nous les avons laissées au choix de l'utilisateur qui peut interfacer à Outilex les nombreux outils statistiques existants. C'est ce qui a été réalisé, par exemple, pour le démonstrateur de Thales RT (cf. section 3.4).

2.3.1 Opérations sur les grammaires

Les WRTN sont construits sous la forme de graphes à l'aide d'un éditeur et sont sauvegardés dans un format XML appelé xgrf, élaboré à partir de (Sastre, 2005). Ces grammaires sont ensuite compilées dans un format XML appelé wrtn, plus adéquat aux traitements informatiques.

[13] Sur ce point, nous sommes allés plus loin que l'annexe technique du projet Outilex ne le prévoyait. En effet, il n'était question au départ que d'automates et transducteurs finis, comme dans le système FSM d'AT&T. Cependant, cette limitation théorique amène en pratique à compiler toutes les grammaires en automates ou transducteurs finis au sens strict, ce qui produit des ressources particulièrement lourdes lorsque les grammaires deviennent complexes. Nous avons donc décidé de traiter les RTN, qui sont plus généraux que les automates et transducteurs finis, et qui ont nécessité des opérateurs plus puissants.

Au cours de cette opération, chaque graphe est optimisé par les opérations d'émondation, suppression des transitions vides, déterminisation et minimisation. Il est également possible de transformer une grammaire en un transducteur fini équivalent, en faisant remonter les sous-graphes dans le graphe principal, éventuellement à une approximation près. Le résultat occupe plus d'espace mémoire mais accélère les traitements. Outilex offre la possibilité de transcoder les graphes du format grf (Unitex) au format xgrf (et inversement) et de les exporter vers le format dot (Gansner & North, 2000).

2.3.2 Traitements utilisant des grammaires

Les traitements utilisant des grammaires identifient dans le texte étiqueté les occurrences des motifs représentés dans les grammaires et peuvent produire plusieurs types de résultat :
– une concordance,
– une modification du texte linéaire ou de l'automate du texte,
– ou une forêt d'arbres d'analyse.

Tous ces traitements reposent sur un même moteur d'analyse, utilisant l'algorithme d'Earley (Earley, 1970) adapté pour traiter d'une part des WRTN (au lieu de grammaires algébriques) et d'autre part un texte sous forme d'automate acyclique (au lieu d'une séquence de mots). Appliqué au corpus de dépêches AFP avec une grammaire des groupes nominaux inspirée de (Paumier, 2003), il a traité 12 466 mots par seconde et trouvé 39 468 occurrences. Ce moteur d'analyse peut être utilisé de façon classique ou de façon glissante, c'est-à-dire en commençant l'analyse à n'importe quel point de l'automate acyclique pour pouvoir la terminer à n'importe quel point.

– Présentation de concordances. Nous avons développé un concordancier qui permet de lister dans leur contexte d'apparition les différentes occurrences des motifs décrits par la grammaire. La taille des contextes gauche et droit peut être paramétrée par l'utilisateur. Les concordances peuvent être classées soit suivant leur ordre d'apparition dans le texte, soit par ordre lexicographique. Nous avons réalisé un format XML de fichier de concordance. Selon ce format, chaque occurrence donne lieu à un élément XML qui comprend les 2 contextes, une référence au motif recherché et le contenu de l'occurrence, segmenté en fonction de la strcuture du motif recherché.
– Application d'un transducteur au texte. Nous avons également développé une fonctionnalité d'application d'un transducteur sur le texte produisant un texte brut comportant les sorties spécifiées dans la grammaire[14]. Dans le cas de grammaires pondérées, les poids fournissent un critère de filtrage entre plusieurs analyses concurrentes. L'analyse retenue est celle dont le chemin a le poids le plus élevé. Un critère supplémentaire sur la longueur des séquences reconnues peut également être utilisé.

  Pour des traitements plus complexes, une variante de cette fonctionnalité produit en sortie un automate correspondant à l'automate du texte auquel sont ajoutées de nouvelles transitions étiquetées par les sorties de la grammaire[15]. Ce procédé peut être utilisé comme complément à l'étiquetage morpho-syntaxique pour la reconnaissance d'unités lexicales semi-figées dont les variations sont trop complexes pour être énumérées sous forme de liste mais qui peuvent être décrites dans des grammaires locales. Par exemple, la figure 6 présente l'automate de la phrase précédente après l'application de la grammaire des adverbes de temps de M. Gross.

[14] Les sorties peuvent au choix être insérées dans le texte d'origine ou remplacer les segments reconnus. Il s'agit d'une option du mode d'application de la grammaire

[15] Ici encore, ces transitions peuvent au choix être simplement ajoutées à l'automate acyclique d'origine, ou remplacer les chemins reconnus.

De plus, le procédé est facilement itérable, ce qui permet de reconnaitre des segments de plus

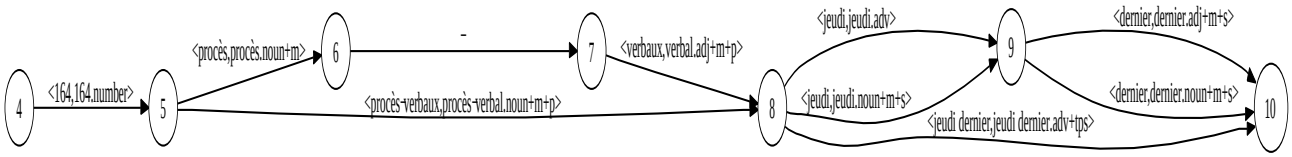

FIG. 6 – Résultat de l'application d'un transducteur sur l'automate du texte

en plus grands. L'itération correspond à la notion mathématique de composition des transducteurs, connue en traitement des langues naturelles sous le nom de cascade de transducteurs (Abney, 1996), à ceci près que les grammaires appliquées par Outilex sont des RTN à sorties, donc plus puissants que des transducteurs finis. Une cascade de RTN à sorties peut être mise en oeuvre avec Outilex par l'intermédiaire de l'interface graphique, en indiquant les RTN et les modes d'application à chaque étape (Friburger & Maurel, 2004).

– Sélection des occurrences en fonction des poids dans la grammaire. Quel que soit le type de résultat produit : concordance, texte brut, automate acyclique, il est possible de sélectionner les occurrences en fonction des poids associés aux transitions de la grammaire. Dans ce cas, on spécifie un seuil numérique. Les chemins de la grammaire dont le poids est inférieur à ce seuil sont inactivés pour la détection des occurrences[16].
– Forêt d'arbres d'analyse. Lorsque l'analyseur fournit comme résultat une forêt partagée d'arbres d'analyse pondérés pour chaque phrase analysée, les noeuds des arbres sont décorés par les éventuelles sorties présentes dans la grammaire.
– Grammaire d'unification pour l'analyse syntaxique. Nous proposons enfin un module d'analyse syntaxique plus élaboré, basé sur des grammaires d'unification dans le formalisme des WRTN décorés (Blanc & Constant, 2005). Ce formalisme allie au WRTN des équations fonctionnelles sur les traits permettant ainsi de formaliser des phénomènes syntaxique d'extraction ou de coréférence. Le résultat de notre analyseur consiste en une forêt partagée d'arbres syntaxiques ; à chaque arbre est associée une structure de traits dans laquelle sont représentées des relations grammaticales entre les constituants qui ont été identifiés durant l'analyse. Cet analyseur est présenté en détail dans (Blanc, 2006).

2.4 Gestion de ressources linguistiques

La réutilisation de lexiques présuppose une certaine flexibilité[17]. La gestion des lexiques et grammaires, dont l'IGM et Systran sont spécialistes, repose sur deux points : construction et maintenance manuelles des ressources dans un format lisible, sous la forme de petites unités cumulables (entrées lexicales, par exemple) ; compilation en un format directement opérationnel. Cependant, les manuels de traitement des langues naturelles, même généralistes et réputés, comme (Jurafsky & Martin, 2000), ne traitent pas ces techniques, qui nécessitent une collaboration étroite d'informaticiens et de linguistes ; et peu de systèmes fournissent les fonctionnalités

[16]Cette fonctionnalité a été implémentée et utilisée dans les démonstrateurs. Elle sera intégrée prochainement à la version distribuée de la plate-forme.

[17]Un lexique n'est pas une ressource statique. En raison de l'évolution de la langue, et en particulier de la langue technique, des mises à jour régulières sont nécessaires ; une nouvelle application d'un lexique peut mettre en jeu la sélection d'un vocabulaire spécifi que au domaine. Il en est de même des grammaires, pour peu qu'elles soient lexicalisées.

requises (Xelda, Intex, Unitex). La plate-forme Outilex propose donc un jeu complet d'outils de gestion de ressources linguistiques.

2.4.1 Flexion automatique des lexiques

Le module de flexion automatique prend en entrée un lexique de lemmes et des règles de flexion et produit en sortie un lexique fléchi. Par exemple, le verbe *carry* appartient à la classe de flexion de la fig. 7 et le module produit entre autres la forme fléchie *carries* avec le code "prs_3s" (troisième personne du singulier au présent). La flexion des mots composés est plus complexe mais

```
<Paradigm code="4">
   <StemTrigger>y</StemTrigger>
   <Inflection caseId="inf"><Form>y</Form></Inflection>
   <Inflection caseId="prs_1s"><Form>y</Form></Inflection>
   <Inflection caseId="prs_2s"><Form>y</Form></Inflection>
   <Inflection caseId="prs_3s"><Form>ies</Form></Inflection>
   <Inflection caseId="prs_p"><Form>y</Form></Inflection>
   <Inflection caseId="prt"><Form>ied</Form></Inflection>
   <Inflection caseId="imp"><Form>y</Form></Inflection>
   <Inflection caseId="ppt"><Form>ied</Form></Inflection>
   <Inflection caseId="ppr"><Form>ying</Form></Inflection>
</Paradigm>
```

FIG. 7 – Règle de flexion

tout aussi nécessaire que celle des mots simples pour la gestion des lexiques. Un formalisme général et ergonomique a été conçu et testé sur trois langues : le français, l'anglais et le polonais. Le formalisme est indépendant de la langue traitée et des outils de flexion des mots simples, qu'il invoque. Il nécessite un codage des entrées de mots composés. Ce codage fait référence à des grammaires de flexion représentées par des RTN (Savary, 2005).

2.4.2 Indexation des lexiques

Afin d'accélérer leur consultation (cf. section 2.2.2), les lexiques sont indexés sur les formes fléchies en utilisant une représentation par automate fini minimal (Revuz, 1991) qui permet de les comprimer tout en offrant un accès rapide à l'information. Le format binaire obtenu (fichier .idx) est adapté aux lexiques à jeu d'étiquettes riche. Le tableau suivant décrit l'indexation du DELAF français (Courtois, 1990) et la taille du fichier obtenu après l'indexation du même lexique par les outils équivalents délivrés avec Unitex[18].

# formes fléchies	# formes canoniques	taille DELA (utf8)	taille XML (xml.gz)	taille idx	temps d'indexation	taille Unitex
1264170	122035	35 Mo	220 Mo (5.3 Mo)	9.5 Mo	1m03s	59 Mo

2.4.3 Alimentation des lexiques

Un travail d'extraction d'information a été réalisé en vue de tester l'applicabilité du traitement par grammaires pour la mise à jour des ressources lexicales. Il s'agissait d'une recherche d'entités nommées dans des textes journalistiques, en vue de l'alimentation d'un lexique de noms géographiques et de noms de personnes, d'entreprises ou d'organisations. Les ressources utilisées sont de trois types : graphes de découpages en phrases, lexiques morpho-syntaxiques, et grammaires à appliquer en cascade.

[18]Le temps d'indexation avec Unitex n'est pas indiqué ici. Le programme d'Unitex étant trop demandeur en ressources mémoire, nous avons dû lancer l'opération sur une autre machine. Le temps de calcul n'était donc pas comparable

Les informations extraites ont permis de constituer des lexiques de 12 000 noms géographiques, validant la méthode utilisée.

2.4.4 Consultation des lexiques par l'utilisateur

Une fonctionnalité de consultation des lexiques par l'utilisateur est opérationnelle.

2.5 Interface utilisateur

La plate-forme Outilex correspond à une conception selon laquelle le traitement des langues naturelles se situe au carrefour de plusieurs domaines : non seulement l'ingénierie informatique, mais aussi la linguistique ou au moins la terminologie descriptive, de façon à pouvoir disposer de ressources linguistiques adaptées aux textes à traiter et suffisamment formalisées. Pour cette raison, l'interface et l'ergonomie de la plate-forme ne pouvaient pas reposer sur la supposition que ses utilisateurs seraient exclusivement des ingénieurs. En particulier, il était essentiel qu'Outilex dispose d'une interface utilisateur graphique dont la prise en main ne nécessite pas l'apprentissage d'un langage formel.

Outre cette interface graphique, Outilex est utilisable dans deux modes : en ligne de commande et par l'interface de programmation d'application (API).

2.5.1 Interface graphique utilisateur

Le projet de plate-forme prévoyait de consacrer la plus grande attention à l'ergonomie de cette interface utilisateur. Un des partenaires du consortium, l'Université de Rouen, a assuré le contrôle de cet aspect du projet à plusieurs stades de la réalisation de la plate-forme (cf. rapport final du Laboratoire de psychologie et neurosciences de la cognition, Amine Rezrazi et al., novembre 2006).

Les ergonomes ont livré leurs premières recommandations dès avant la première version d'Outilex, en se fondant sur le système Unitex, créé en 2002 à l'Université de Marne-la-Vallée. En effet, ce système pouvait être vu comme un prototype d'Outilex et correspondait partiellement à ce que nous en attendions. Ces premières remarques nous ont guidés dans la conception de l'interface utilisateur d'Outilex, qui a été développée en Java. Elle permet de travailler sur différents projets regroupant un ensemble de ressources (textes, lexiques et grammaires). L'utilisateur définit une chaîne de traitement. Les entrées, les sorties et les résultats intermédiaires sont visualisés dans un conteneur à onglets dédié à cet effet. La première version a été à son tour examinée et critiquée. La prise en compte des critiques a permis d'améliorer les qualités de l'interface.

Les tests finaux d'utilisabilité, réalisés avec 10 sujets, ont donné un taux de réussite de 99 % dans la réalisation des tâches.

L'interface utilisateur est un des points forts d'Outilex. Peu d'environnements de traitement des langues naturelles sont aussi faciles à utiliser par des linguistes ou des terminologues pour mettre au point et tester des ressources linguistiques et les traitements qui les utilisent. Les systèmes Intex et Unitex, précurseurs d'Outilex, font partie de la même catégorie d'environnements.

2.5.2 Utilisation en ligne de commande

L'utilisation en ligne de commande apporte plus de flexibilité que l'interface utilisateur et permet de mettre au point une chaîne de traitement dont les ressources linguistiques sont déjà déterminées.

2.5.3 Utilisation de l'interface de programmation d'application (API)

L'API permet d'éviter que les résultats intermédiaires du traitement soient sauvegardés sur disque. Les communications entre traitements sont assurées par des flux XML, ce qui accélère le traitement, en particulier lorsque les documents à traiter sont nombreux. Les principales opérations proposées par Outilex sont utilisables par l'intermédiaire de l'API. De plus, certaines opérations de bas niveau, comme les opérations mathématiques de base sur les automates finis (intersection, union, complémentation, concaténation...) ne sont disponibles que via l'API.

2.6 Aspects liés à la normalisation

En raison de son ambition internationale, Outilex a collaboré avec le projet RNIL[19] et avec les autres experts internationaux pour apporter sa contribution à la définition de normes en matière de modèles de ressources linguistiques. Les projets de normes qui sont issus de ce travail sont en cours d'élaboration au niveau de l'ISO.

Les partenaires les plus impliqués dans cette activité ont été le Loria (cf. rapport final du Loria, Gil Francopoulo, mars 2006), l'IGM et Systran. Les activités de normalisation d'Outilex ont concerné trois aspects de la plate-forme : le traitement sans lexiques, le traitement par lexiques, et la gestion des ressources linguistiques. Nous avons contribué à l'élaboration de deux projets de normes de formats, l'une pour la représentation de textes, et l'autre pour la représentation de lexiques.

2.6.1 Représentation de textes

Le projet de norme intitulé "Cadre d'annotation morpho-syntaxique" ou MAF (Clément & de la Clergerie, 2005) concerne la représentation de l'annotation morpho-syntaxique des textes pour le traitement automatique des langues. Il fait suite à d'autres contributions, notamment la Text Encoding Initiative (TEI), qui était plutôt tournée vers les applications éditoriales et la visualisation des textes. Dans la plate-forme Outilex, deux formats de représentation de textes sont étroitement liés aux formats MAF :
– le format seg.xml de représentation du texte tokenisé (cf. section 2.1),
– le format fsa.xml de représentation du texte étiqueté (cf. section 2.2.3).

2.6.2 Formats de représentation de lexiques

Le projet de norme "Cadre de balisage lexical" ou LMF (Francopoulo, 2003; Francopoulo *et al.*, 2006) concerne la représentation des lexiques pour le traitement automatique des langues. L'un

[19] Ressources normalisées en ingénierie des langues.

des deux formats adoptés par Outilex pour les lexiques morpho-syntaxiques, dic.xml (cf. section 2.2.1), est directement lié à l'extension morpho-syntaxique du projet de norme LMF. Ces formats Outilex tirent parti de deux circonstances : d'une part, l'émergence actuelle de modèles de données consensuels dans le cadre de l'ISO ; d'autre part, le fait que l'IGM dispose de lexiques du français construits manuellement et d'une grande couverture[20].

Dans le cas particulier des lexiques de noms propres, particulièrement importants pour certains traitements tels que l'extraction d'information ou l'alignement de textes, un format XML spécialisé a été élaboré ainsi qu'une DTD. Ce format permet de coder des informations sémantiques, faisant ainsi du lexique une ontologie distinguant, par exemple, noms de personnes, de lieux, d'oeuvres ou d'évènements. La flexion, qui peut sembler peu utile en français pour les noms propres, est cependant représentée dans le format en raison des nombreuses langues dans lesquelles les noms propres se fléchissent normalement.

LCI a fait passer une série de ses lexiques terminologiques en XML (cf. rapport final de LCI, Tita Kyriacopoulou, novembre 2006). Cela a nécessité une normalisation des informations et une homogénéisation des données. Les ressources obtenues ont été placées sur le site d'Outilex.

2.7 Diffusion de la plate-forme

La diffusion des résultats d'Outilex est assurée par le site web d'Outilex et par les publications scientifiques.

La communication entre les partenaires pendant le projet a été assurée par un serveur de fichiers CVS maintenu par Systran, et par le site web (http ://www.at-lci.com/outilex) mis en place par LCI. Comme prévu dans le protocole de collaboration entre les partenaires, les données offertes à la suite d'Outilex (logiciels et ressources linguistiques) le seront moyennant une cotisation annuelle de 100 euros. C'est le même site qui servira au paiement électronique sécurisé, toujours sous la responsabilité de LCI (cf. rapport final de LCI, Tita Kyriacopoulou, novembre 2006).

Nous avons fait un effort de promotion de la plate-forme au sein de la communauté du traitement automatique des langues en France et à l'étranger par le biais de publications scientifiques. Citons notamment des publications sur les formats XML (Laporte, 2005; Sastre, 2005), la normalisation (Francopoulo *et al.*, 2006) et des présentations générales (Blanc *et al.*, 2006; Blanc & Constant, 2006a), qui ont été favorablement perçues par les participants aux colloques. Lors de la présentation à TALN, notamment, plusieurs des participants ont fait part de leur désir de voir assurée la pérennité de la plate-forme.

3 Démonstrateurs

Les démonstrateurs réalisés par les partenaires visent à tester les possibilités industrielles offertes par la plate-forme Outilex. Parmi les trois types de traitement proposés par Outilex, ils couvrent les deux plus innovants : par lexiques et par grammaires. En effet, les partenaires disposaient déjà d'outils satisfaisants pour le traitement sans lexiques, qui est le plus simple. Parmi les fonctionnalités utilisées dans les démonstrateurs, on peut citer :

[20] Ces deux circonstances sont liées à des travaux effectués au LADL sous la direction de Maurice Gross, la première par l'intermédiaire du projet Genelex de normalisation de lexiques (Normier & Nossin, 1990), la deuxième à travers le système de lexiques Dela (Courtois, 1990; Courtois, 2004).

– la consultation de lexiques d'expressions multi-mots,
– l'utilisation d'automates finis à transitions pondérées reconnaissant des transcriptions phonétiques,
– l'application de transducteurs finis reconnaissant des expressions grâce à des lexiques et les reformulant sous une forme normalisée ou étendue.

Ces fonctionnalités correspondent à une nouvelle génération d'outils de traitement des langues naturelles, beaucoup plus fondée sur les ressources linguistiques que les outils antérieurs.

Les démonstrateurs ont validé, entre autres, l'interopérabilité d'Outilex avec d'autres systèmes, en mettant en oeuvre des chaînes de traitement qui combinent plusieurs environnements.

3.1 Moteur de recherche interlingue

Ce démonstrateur applicatif permet d'effectuer une recherche d'information interlingue dans des dépêches de l'AFP en français, en anglais et en espagnol. Il a été développé par le laboratoire LIC2M du CEA (cf. rapport final du CEA, Romaric Besançon, novembre 2006). Il utilise les bibliothèques d'analyse à l'aide d'automates à états finis qui ont été développées par les partenaires. Il s'agit d'un moteur de recherche interlingue qui utilise une analyse linguistique profonde des documents et des requêtes, en différentes langues. Cette analyse est effectuée par l'outil LIMA, développé au LIC2M, dans lequel s'intègrent des modules de traitements qui s'appuient sur les technologies d'automates à états finis d'Outilex. S'appuyant sur cette analyse, le moteur de recherche de démonstration est interrogeable à partir d'une interface web.

Les traitements d'Outilex sont intégrés dans les chaînes de traitement de l'analyseur LIMA. Les ressources linguistiques passent par une conversion de formats.

3.2 Reconnaisseur d'expressions multi-mots

Lingway a inclus dans son application Lingway Knowledge Management (LKM) un composant issu d'Outilex et dont la fonction est d'engendrer des variantes d'expressions terminologiques multi-mots trouvées dans la requête de l'utilisateur (cf. rapport final de Lingway, Hugues de Mazancourt, Vincent Le Maout, octobre 2006) . L'application LKM permet de retrouver, organiser et comprendre l'information. Elle comprend un moteur de recherche sémantique multilingue, des outils d'indexation et de catégorisation.

Le reconnaisseur d'expressions multi-mots permet de reformuler et d'étendre une requête de l'utilisateur. Il est composé de règles. Chaque règle fonctionne en deux temps. Elle reconnaît d'abord une séquence de mots comportant une ou plusieurs expressions multi-mots recensées dans le lexique, malgré des différences éventuelles entre la forme trouvée dans la requête et la forme canonique stockée dans le lexique (par exemple, *salaire encadrement* et *salaire des cadres*). Ensuite, elle reformule la séquence détectée de manière à faire apparaître dans la requête les formes canoniques des expressions reconnues, ainsi éventuellement que d'autres expressions qui n'ont pas été détectées directement, en raison, par exemple, de la présence d'une coordination (*fabricant de cycles* dans *fabricant et réparateur de cycles*). Dans cette étape de reformulation, certaines règles identifient également des groupes nominaux non récursifs ou chunks.

3.3 Moteur de recherche dans des documents XML

Le Laboratoire d'informatique de Paris 6 (LIP6) a développé une plate-forme pour la recherche d'information structurée, SIRXQL, qui utilise certaines fonctionnalités d'Outilex (cf. rapport final du LIP6, Benjamin Piwowarski, septembre 2006) . La principale innovation de ce travail est la possibilité de rechercher des informations dans des documents XML en prenant en compte leur structure XML. Ainsi, l'unité élémentaire n'est plus le document mais un élément dans la structure du document. Outilex est utilisé par SIRXQL pour détecter les mots composés. L'interfaçage entre Outilex et le reste du système SIRXQL est réalisé grâce un module en Python. Des tests comparatifs ont été réalisés pour évaluer l'apport d'Outilex. Ces tests ont montré que lorsque l'utilisateur considère un nombre d'éléments égal, par exemple, à 4 fois le nombre d'éléments qui sont en fait pertinents à sa requête, la présence d'Outilex dans le système permet d'obtenir environ 2 fois plus d'éléments pertinents parmi les éléments considérés.

3.4 Filtre thématique de messages audio

Thales RT a développé un démonstrateur de filtrage audio s'appuyant sur la plate-forme Outilex et sur un composant d'apprentissage de pondération (cf. rapport final de Thales RT, Bénédicte Goujon, septembre 2006). Ce composant engendre des graphes pondérés au format xgrf d'Outilex. Ces graphes sont des représentations de la requête de l'utilisateur, traitée grâce à un échantillon de messages représentatif des messages audio à filtrer. Les messages audio sont traités par un programme de reconnaissance de parole. La plate-forme Outilex applique la requête aux messages et indique dans quels messages les mots et expressions de la requête ont été trouvés. Les tests effectués montrent que l'utilisation d'Outilex permet un traitement plus rapide qu'avec l'environnement FSM, précédemment utilisé. De plus, la construction de la requête de l'utilisateur a été jugée plus simple à effectuer qu'avec FSM, en raison d'une meilleure ergonomie.

3.5 Extracteur d'informations pour l'alimentation d'une base de connaissances

Thales Com a développé un démonstrateur métier portant sur l'extraction d'informations dans des textes tels que des dépêches et des rapports, à des fins d'alimentation d'une base de connaissances exploitée par de la fouille de données et des outils d'analyse de réseaux sémantiques (cf. rapport final de Thales Com, Catherine Gouttas, novembre 2006). Des composants d'extraction d'information ont été élaborés sous la forme de ressources linguistiques, et notamment de grammaires locales, au format Outilex. Ces composants, utilisés avec la plate-forme Outilex, permettent une normalisation des informations reconnues. Les travaux réalisés ont validé l'intérêt de la plate-forme Outilex dans un contexte industriel, en vue de répondre à des besoins métier non triviaux. En particulier, elle permet de développer rapidement de nouveaux composants et elle comporte des formats normalisés.

4 Conclusion et perspectives

La plate-forme Outilex, dans sa version actuelle, effectue toutes les opérations fondamentales du traitement automatique du texte écrit : traitements sans lexiques, exploitation de lexiques et de grammaires, gestion de ressources linguistiques. Les données manipulées sont structurées dans des formats XML, et également dans d'autres formats plus compacts, soit lisibles soit binaires ; les convertisseurs de formats nécessaires sont inclus dans la plate-forme ; le formalisme des WRTN permet de combiner des méthodes statistiques avec des méthodes fondées sur des ressources linguistiques. Enfin, des lexiques issus du LADL, construits manuellement et d'une couverture substantielle seront distribués avec la plate-forme sous licence LGPL-LR.

Le développement de la plate-forme a nécessité une expertise conjointe des éléments informatiques et linguistiques du problème ; il a pris en compte les besoins de la recherche fondamentale et ceux des applications. Nous pensons qu'il n'aurait pas été possible sans un consortium aussi varié.

L'avenir de la plate-forme Outilex est conçu dans le cadre du développement collaboratif. L'IGM et Systran ont indiqué leur intention de continuer à développer la plate-forme. Le consortium a discuté de plusieurs perspectives d'évolution.

– La recherche approximative pourrait être conçue comme le comptage d'indices qui ne sont pas en eux-mêmes décisifs, mais auxquels on sait affecter un certain poids : reconnaissance de synonymes approximatifs, de formes erronées, de collocations... Cette direction de recherche est une extension des expériences sur les grammaires pondérées qui ont déjà été effectuées dans le cadre d'Outilex.
– Nous nous intéressons également aux textes phonétiques (transcrits ou obtenus par reconnaissance de la parole) et aux SMS, donc à des genres plus spontanés que le texte écrit de qualité auquel Outilex a été pour l'essentiel appliqué jusqu'à présent.
– Enfin, la plate-forme offre une bonne base pour le traitement de nouvelles langues. Elle est déjà compatible avec la plupart des langues flexionnelles ou agglutinantes à suffixes, en raison de l'expérience des partenaires dans ce domaine et des précautions qui ont été prises pendant le projet. Cependant, l'extension au chinois et à l'arabe, par exemple, nécessiterait de nouveaux développements.

Références

ABNEY S. (1996). Partial parsing via finite-state cascades. In *Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information, Prague*, p. 8–15.

BLANC O. (2006). *Algorithmes d'analyse syntaxique par grammaires lexicalisées. Optimisation et traitement de l'ambiguïté*. PhD thesis, IGM, Université de Marne-la-Vallée.

BLANC O. & CONSTANT M. (2005). Lexicalization of grammars with parameterized graphs. In *Proc. of RANLP 2005*, p. 117–121, Borovets, Bulgarie : INCOMA Ltd.

BLANC O. & CONSTANT M. (2006a). Outilex, a linguistic platform for text processing. In *Proc. of COLING/ACL*, p. 117–121. Demonstration.

BLANC O. & CONSTANT M. (2006b). *Outilex Platform Graphical Interface. User Guide*. Rapport interne 2006-07, IGM, Université de Marne-la-Vallée. 52 pages.

BLANC O., CONSTANT M. & LAPORTE E. (2006). Outilex, plate-forme logicielle de traitement de textes écrits. In C. FAIRON & P. MERTENS, Eds., *Actes de TALN 2006 (Traitement automatique des langues naturelles)*, p. 83–92, Louvain.

BLANC O. & DISTER A. (2004). Automates lexicaux avec structure de traits. In *Actes de RECITAL*, p. 23–32.

CLÉMENT L. & DE LA CLERGERIE É. (2005). MAF : a morphosyntactic annotation framework. In *Proc. of the Language and Technology Conference, Poznan, Poland*, p. 90–94.

COURTOIS B. (1990). Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, **87**, 11–22.

COURTOIS B. (2004). Dictionnaires électroniques DELAF anglais et français. In C. LECLÈRE, É. LAPORTE, M. PIOT & M. SILBERZTEIN, Eds., *Lexique, Syntaxe et Lexique-Grammaire/Syntax, Lexis and Lexicon-Grammar*, Lingvisticae Investigationes Supplementa, p. 113–123. Amsterdam/Philadelphie : Benjamins.

CUNNINGHAM H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, **36**, 223–254.

DANLOS L. (2005). Automatic recognition of French expletive pronoun occurrences. In *Companion Volume of the International Joint Conference on Natural Language Processing, Jeju, Korea*, p. 2013.

EARLEY J. (1970). An efficient context-free parsing algorithm. *Comm. ACM*, **13**(2), 94–102.

FRANCOPOULO G. (2003). *Proposition de norme des lexiques pour le traitement automatique du langage*. AFNOR. 21 p.

FRANCOPOULO G., GEORGE M., CALZOLARI N., MONACHINI M., BEL N., PET M. & SORIA C. (2006). Lexical markup framework (LMF). In *Proc. of LREC, Genoa*.

FRIBURGER N. & MAUREL D. (2004). Finite-state transducer cascades to extract named entities in texts. *Theoretical Computer Science*, **313**(1), 93–104.

GANSNER E. R. & NORTH S. C. (2000). An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, **30**(11), 1203–1233.

GROSS M. (1993). Local grammars and their representation by finite automata. In M. HOEY, Ed., *Data, Description, Discourse, Papers on the English Language in honour of John McH Sinclair*, p. 26–38. London : Harper-Collins.

GROSS M. (1997). The construction of local grammars. In E. ROCHE & Y. SCHABÈS, Eds., *Finite-state language processing*, Language, Speech, and Communication Series, p. 329–354. Cambridge (Mass.) : MIT Press.

JURAFSKY D. & MARTIN J. (2000). *Speech and language processing*. Prentice Hall. 934 p.

KRSTEV C., VITAS D., MAUREL D. & TRAN M. (2005). Multilingual ontology of proper names. In *Proc. of the Language and Technology Conference, Poznan, Poland*, p. 116–119.

LAPORTE É. (2005). Lexicon management and standard formats. In *Proc. of the Language and Technology Conference, Poznan, Poland*, p. 318–322.

LOPER E. & BIRD S. (2002). NLTK : the natural language toolkit. In *Proc. of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Philadelphia*.

MASON O. (2004). Automatic processing of local grammar patterns. In *Proc. of the 7th Annual CLUK (the UK special-interest group for computational linguistics) Research Colloquium*.

MOHRI M., PEREIRA F. & RILEY M. (1998). A rational design for a weighted finite-state transducer library. *Lecture Notes in Computer Science*, **1436**.

NAKAMURA T. (2005). Analysing texts in a specific domain with local grammars : The case of stock exchange market reports. In *Linguistic Informatics - State of the Art and the Future*, p. 76–98. Amsterdam/Philadelphia : Benjamins.

NORMIER B. & NOSSIN M. (1990). Genelex project : Eureka for linguistic engineering. In *Proc. of the International Workshop on Electronic Dictionaries, OISA, Kanagawa, Japan*, p. 63–70.

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

POIBEAU T. (2001). Extraction d'information dans les bases de données textuelles en génomique au moyen de transducteurs à états finis. In D. MAUREL, Ed., *Actes de TALN 2001 (Traitement automatique des langues naturelles)*, p. 295–304, Tours : ATALA Université de Tours.

REVUZ D. (1991). *Dictionnaires et lexiques : méthodes et algorithmes*. PhD thesis, Université Paris 7.

SASTRE J. M. (2005). XML-based representation formats of local grammars for NLP. In *Proc. of the Language and Technology Conference, Poznan, Poland*, p. 314–317.

SAVARY A. (2005). Towards a formalism for the computational morphology of multi-word units. In Z. VETULANI, Ed., *Proceedings of the 2nd Language and Technology Conference, Poznan, Poland*, p. 305–309.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. *Proceedings of the International Conference on New Methods in Language Processing*.

SILBERZTEIN M. (1993). *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*. Paris : Masson. 234 p.