THÈSE DE DOCTORAT D’ÉTAT
SPÉCIALITÉ : INFORMATIQUE

Présentée par
Farida YAMOUNI AOUGHLIS

Sujet

Construction d’un dictionnaire électronique de terminologie informatique et analyse automatique de textes par grammaires locales

Soutenue le 12/12/2010 devant le jury d’examen composé de :

M’ Soltane AMEUR
Professeur
UMMTO
Président

M’me Elisabeth Métai
Professeur
CNAM, Paris
Rapporteur

M’ Max Silberztein
Professeur
U. de Franche-Comté
Co-Rapporteur

M’ Mohand Boughanem
Professeur
U. P. Sabatier Toulouse
Examinateur

M’ Rachid Ahmed Ouamer
Maître de Conférences
UMMTO
Examinateur
Remerciements

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance à ma directrice de thèse, le professeur Elisabeth Métais pour tous ses encouragements, ses conseils.

Je tiens aussi à remercier très vivement mon co-directeur de thèse, Max Silberztein, professeur à l’Université de Franche-Comté pour tout le temps précieux qu’il m’a consacré, son aide, sa disponibilité.

Mes respects et ma gratitude vont également aux membres de mon jury qui m’ont fait l’honneur de juger ce travail :

A Monsieur Soltane Ameur, professeur à l’Université Mouloud Mammeri de Tizi Ouzou, pour m’avoir fait l’honneur d’être président du jury ;

A Monsieur, Mohand Boughanem, professeur à l’université de Toulouse pour avoir accepté de juger mon travail ;

A Monsieur Rachid Ahmed Ouamer, Maître de conférences à l’Université de Tizi Ouzou pour avoir accepté de juger mon travail.
A mes chers enfants

A la mémoire de ma très chère maman

A la mémoire de Maurice Gross
Table des matières

Introduction ............................................................................................................................. 18
Problématique ............................................................................................................................ 18
Objectifs ..................................................................................................................................... 19
Plan de la thèse .......................................................................................................................... 19

1 Etat de l’art sur l’extraction automatique de terminologie .............................................. 22
   1.1 Introduction ......................................................................................................................... 22
   1.2 L’extraction de terminologie ............................................................................................... 22
      1.2.1 Extraction manuelle ..................................................................................................... 22
         1.2.1.1 La collecte de textes écrits ......................................................................................... 22
         1.2.1.2 La lecture des textes et l’extraction .......................................................................... 23
         1.2.1.3 L’acquisition ............................................................................................................. 23
      1.2.2 Extraction automatique ................................................................................................. 23
   1.3 Les outils d’extraction automatique de terminologie ..................................................... 23
   1.4 Les différentes approches ................................................................................................ 24
      1.4.1 Les méthodes linguistiques .......................................................................................... 24
         1.4.1.1 TERMINO .................................................................................................................. 24
         1.4.1.2 LEXTER .................................................................................................................... 26
         1.4.1.3 FASTER .................................................................................................................... 27
         1.4.1.4 XTERM ..................................................................................................................... 28
         1.4.1.5 TERMINAE ................................................................................................................ 28
      1.4.2 Les méthodes statistiques ............................................................................................ 28
         1.4.2.1 ANA (Acquisition Naturelle Automatique) ................................................................. 29
         1.4.2.2 MANTEX ................................................................................................................... 30
         1.4.2.3 LIKES ......................................................................................................................... 30
   1.4.3 Les méthodes mixtes ..................................................................................................... 30
1.4.3.1 ACABIT (Automatic Corpus Based Acquisition of Binary terms) ........................................... 30
1.4.3.2 ASIUM ................................................................. 31
1.4.3.3 XTRACT ................................................................. 31
1.5 Tableau récapitulatif ................................................................. 32
1.6 Conclusion ............................................................................. 33
2 Les dictionnaires électroniques et les bases de données terminologiques ........................................... 35
  2.1 Introduction........................................................................... 35
  2.2 Le système DELA ................................................................. 35
    2.2.1 Introduction .................................................................. 35
    2.2.2 Les dictionnaires des mots simples DELAS .................... 36
    2.2.3 Le dictionnaire des mots composés DELAC .................... 36
      2.2.3.1 Exemples d’entrées .............................................. 37
      2.2.3.2 Les différentes classes de noms composés ............... 37
        2.2.3.2.1 Noms composés binaires (longueur 2) ................ 37
        2.2.3.2.2 Noms composés ternaires ............................... 38
        2.2.3.2.3 Plus longs .................................................... 38
    2.2.4 Le dictionnaire des mots composés fléchis DELACF ....... 39
    2.2.5 Le Lexique-grammaire ................................................... 39
      2.2.5.1 Introduction ........................................................ 39
      2.2.5.2 Le lexique-grammaire des verbes du français .......... 40
  2.3 Les dictionnaires de terminologie et bases de données terminologiques ........................................... 41
    2.3.1 IATE .......................................................................... 41
    2.3.2 Le dictionnaire SensAgent .......................................... 43
    2.3.3 Le glossaire OSINET ................................................... 45
    2.3.4 Le dictionnaire électronique LVF ................................. 45
    2.3.5 Dictionnaire de l’informatique et de l’internet Dicofr.com ................................................. 46
    2.3.6 Le répertoire terminologique 2000 ................................. 47
    2.3.7 UNBIS ...................................................................... 48
3.3.6 Traitement de corpus ................................................................. 69
3.3.7 Construction, édition et gestion de concordances sophistiquées ............. 69
3.3.8 Annotation interactive de corpus ..................................................... 70
3.4 Les dictionnaires NooJ ........................................................................ 70
3.4.1 Les ALUs (Atomic Linguistic Units) .................................................. 70
3.4.2 Ressources pour reconnaître les unités linguistiques atomiques............. 71
   3.4.2.1 Les Dictionnaires ........................................................................ 72
   3.4.2.2 La Morphologie ............................................................................ 72
3.4.3 Outils pour décrire la morphologie ................................................... 73
   3.4.3.1 Descriptions flexionnelles et dérivationnelles ............................ 73
   3.4.3.2 Grammaires flexionnelles et dérivationnelles ............................. 73
3.5 Format des dictionnaires NooJ .............................................................. 73
   3.5.1 Exemples d’entrées .......................................................................... 73
   3.5.2 Informations linguistiques ............................................................... 74
   3.5.3 Codes d’information spéciaux .......................................................... 74
   3.5.4 Propriétés lexicales ......................................................................... 75
   3.5.5 Variantes lexicales .......................................................................... 75
3.6 Fichiers de définition des propriétés ".DEF" ........................................... 75
3.7 Représentation formelle des dictionnaires électroniques .......................... 76
3.8 Conclusion .............................................................................................. 77
4 Aspects linguistiques de la terminologie .................................................. 79
   4.1 Introduction .......................................................................................... 79
   4.2 Définitions ........................................................................................... 79
       4.2.1 Terminologie .................................................................................. 79
       4.2.2 Mot .................................................................................................. 79
       4.2.3 Terme ............................................................................................. 79
       4.2.4 Forme simple (définition orthographique) ....................................... 80
4.2.5 Mot simple ........................................................................................................80
4.2.6 Forme composée ...............................................................................................80
4.2.7 Mot composé ......................................................................................................80
4.2.8 Synapsie .............................................................................................................81
4.3 Mots composés et notions de composition .........................................................81
  4.3.1 Mots composés ..................................................................................................81
  4.3.2 Notion de composition .....................................................................................81
4.4 Degré de figement des noms composés ...............................................................82
  4.4.1 "Il n’y a pas de relation syntaxique entre les 2 noms" ........................................82
  4.4.2 Pronominalisation ............................................................................................83
  4.4.3 Figement partiel ..............................................................................................83
4.5 Les groupes nominaux productifs et les noms composés lexicalisés ...............84
  4.5.1 L’atomicité sémantique ..................................................................................84
  4.5.2 L’institutionnalisation de l’usage ....................................................................85
    4.5.2.1 Termes institutionnalisés ..........................................................................85
    4.5.2.2 Termes inexistants ...................................................................................85
  4.5.3 Restrictions distributionnelles .........................................................................85
  4.5.4 Analyse transformationnelle ..........................................................................86
4.6 Variantes terminologiques ..................................................................................86
  4.6.1 La surcomposition ..........................................................................................86
  4.6.2 Les insertions ..................................................................................................86
  4.6.3 La coordination ................................................................................................87
4.7 Conclusion .............................................................................................................87
5 Elaboration des dictionnaires électroniques de la terminologie informatique ..........89
  5.1 Introduction ..........................................................................................................89
  5.2 Dictionnaires usuels et dictionnaires spécialisés ...............................................90
  5.3 Extraction des termes .........................................................................................91
5.4  Dictionnaire des mots composés "Info_comp.dic" ......................................................... 91

5.4.1  Caractéristiques des composants des noms composés (notion de tête) ...................... 91

5.4.2  Les déterminants possibles pour les noms composés ............................................. 92

5.4.3  Notion de mot vide, mot plein ..................................................................................... 92

5.4.4  Etude et classification syntaxique des termes ............................................................ 94

   5.4.4.1  Codes grammaticaux ............................................................................................. 94

   5.4.4.2  Longueur 2 ............................................................................................................ 94

   5.4.4.3  Longueur 3 ............................................................................................................ 95

   5.4.4.4  Longueur 4 (termes contenant 4 mots pleins) ...................................................... 96

   5.4.4.5  Longueur 5 (termes contenant 5 mots pleins) ...................................................... 96

5.4.5  Format d’une entrée ..................................................................................................... 96

5.4.6  Les informations sémantiques ..................................................................................... 98

   5.4.6.1  Systèmes d’information........................................................................................ 98

   5.4.6.2  Compilation: "+comp" ......................................................................................... 99

   5.4.6.3  Algorithmique : "+algo" ...................................................................................... 99

   5.4.6.4  Langages (programmation) : "+lang" .................................................................. 99

   5.4.6.5  Architecture des ordinateurs : +arch" ............................................................... 99

   5.4.6.6  Systèmes d’exploitation : "+expl" ....................................................................... 99

   5.4.6.7  Réseaux et télécommunications : "+ rest" ......................................................... 100

   5.4.6.8  Internet and groupware : "+intn" ....................................................................... 100

   5.4.6.9  Informatique appliquée : "+ iapl" ....................................................................... 101

   5.4.6.10 Intelligence artificielle : "+iart" ......................................................................... 102

5.5  Variantes ......................................................................................................................... 103

5.5.1  Définitions ................................................................................................................. 103

   5.5.1.1  Abréviations ....................................................................................................... 103

   5.5.1.2  Acronymes ......................................................................................................... 104

   5.5.1.3  Sigles ................................................................................................................... 104
Conclusion ........................................................................................................... 123

7 Grammaires locales .......................................................................................... 125

7.1 Introduction....................................................................................................... 125

7.2 Les automates finis ......................................................................................... 125

7.2.1 Automates finis et graphes ........................................................................ 125

7.2.2 Exemples ...................................................................................................... 127

7.2.3 Graphes et RTN ............................................................................................ 128

7.2.4 Utilisation des grammaires locales dans le TAL ....................................... 129

7.2.4.1 Reconnaissance et étiquetage automatiques ........................................ 129

7.2.4.2 Extraction de données ............................................................................. 130

7.2.4.3 Ambiguïtés .............................................................................................. 130

7.3 Construction des Grammaires locales des mots composés d’informatique .... 131

7.3.1 Construction des graphes : une méthode empirique ................................ 131

7.3.2 Les grammaires locales des termes d’informatique .................................. 131

7.3.2.1 Construction après recensement manuel (textes écrits) ..................... 131

7.3.2.2 Construction après concordance (le texte est un fichier) .................. 133

7.4 Application d’une grammaire à un texte d’informatique ................................ 135

7.5 Quelques grammaires locales des termes d’informatique ................................ 137

7.5.1 Recensement des ”mots-clés” d’informatique ........................................ 137

7.5.2 Grammaire locale ”mémoire” .................................................................... 139

7.5.3 Grammaire locale ”Registre” .................................................................... 141

7.6 Conclusion ....................................................................................................... 142

8 Analyse automatique de textes ........................................................................ 144

8.1 Introduction....................................................................................................... 144

8.2 Annotation automatique de textes d’informatique ....................................... 144

8.2.1 Analyse textuelle et annotation automatique .......................................... 144

8.2.2 Recherches linguistiques et concordances .............................................. 148
8.3 Constitution du corpus ..............................................................................................................151
  8.3.1 Caractéristiques générales du corpus étudié...........................................................................151
  8.3.2 Liste des formes dans le corpus ...............................................................................................154
  8.3.3 Table des concordances du token sélectionné ...........................................................................155
  8.3.4 Affichage du token sélectionné dans le texte source ...............................................................156
8.4 Evaluation et couverture lexicale ..................................................................................................156
  8.4.1 Analyse linguistique de corpus .................................................................................................156
  8.4.2 Résultats ......................................................................................................................................157
8.5 Extraction automatique et acquisition de termes composés ..........................................................158
  8.5.1 Cas de la coordination : acquisition de nouveaux termes .......................................................158
    8.5.1.1 Différents cas de coordination ..........................................................................................158
    8.5.1.2 Grammaires locales pour la coordination et extraction de termes ..................................159
  8.5.2 Extraction de terminologie avec NooJ .....................................................................................161
    8.5.2.1 Extraction à partir d’un texte ............................................................................................161
    8.5.2.2 Extraction à partir d’un corpus ..........................................................................................163
  8.5.3 Extraction de terminologie à l’aide de la méthode statistique ANA ........................................165
  8.5.4 Étude des résultats ....................................................................................................................167
  8.5.5 Statistiques ...............................................................................................................................167
  8.5.6 Indexation automatique de textes ............................................................................................168
    8.5.6.1 Indexation automatique .....................................................................................................168
8.6 Conclusion ......................................................................................................................................171

Conclusion et perspectives ................................................................................................................173

Annexe1 : Liste des ouvrages utilisés pour l’extraction manuelle ....................................................176
Annexe2 : Outils formels de base, théorie .......................................................................................177
Bibliographie ........................................................................................................................................182
Liste des figures

Figure 2-1 : Extrait de la table 32CL du lexique-grammaire (Gross M. 1975) ...........................................40
Figure 2-2 : Préférences de recherche avec IATE pour le terme "mémoire". ..............................................42
Figure 2-3 : Résultats de la recherche avec IATE pour le terme "mémoire". ..............................................43
Figure 2-4 : Page d’accueil du dictionnaire en ligne “SenSagent”. ..............................................................44
Figure 2-5 : Résultats de la recherche du mot "mémoire". ............................................................................44
Figure 2-6 : Recherche pour le terme "mémoire". .......................................................................................46
Figure 2-7 : Recherche avec UNBIS........................................................................................................48
Figure 2-8 : Résultats de la recherche du terme "mémoire" dans le GDT. .....................................................49
Figure 2-9 : Recherche WordNet pour le terme ‘mémoire’. ........................................................................50
Figure 2-10 : Page d’accueil GENOMA. ..................................................................................................51
Figure 2-11 : Navigation dans le dictionnaire des développeurs. .................................................................52
Figure 2-12 : Page de recherche et navigation EuroVoc ..............................................................................56
Figure 2-13 : Résultats de la recherche avec EuroVoc pour le terme « mémoire » ......................................56
Figure 2-14 : Alexandria, le web 2.0 pour les dictionnaires. ....................................................................57
Figure 2-15 : Dictionnaires en ligne. .........................................................................................................58
Figure 2-16 : Page d’accueil du DiColInfo. ...............................................................................................60
Figure 2-17 : RechercheTermSciences pour le terme "mémoire". ...............................................................62
Figure 2-18 : Page de recherche du DicoduWeb pour la liste « M ». .........................................................63
Figure 3-1 : Ressources "morphology" et "dictionary". ..............................................................................71
Figure 3-2 : la zone "Morphology". .......................................................................................................72
Figure 5-1 : Extraits du dictionnaire "info_comp". .....................................................................................97
Figure 5-2 : Extrait du fichier de descriptions flexionnelles "composés-Flex.nof". .................................98
Figure 5-3 : Liste des sigles INRIA. .........................................................................................................106
Figure 5-4 : Un extrait du dictionnaire des sigles d’informatique "sigles.dic" .............................................107
Figure 5-5 : Extrait de la liste des abréviations d’informatique Wikipedia, lettre "A". ..........................108
Figure 5-6 : Extrait du dictionnaire des abréviations d’informatique. ..................................................109
Figure 6-1 : Graphe du paradigme TABLE (Silberztein M. 2003). .......................................................... 114
Figure 6-2 : Graphe du paradigme flexionnel "cousin" (Silberztein M. 2003). ........................................ 115
Figure 6-3 : Sous Graphes "Genre" et "Nombre" (Silberztein M. 2003). ................................................ 115
Figure 6-4 : Extrait du dictionnaire "Info_comp". ................................................................................ 120
Figure 6-5 : Extrait de la description des modèles de flexions ............................................................... 121
Figure 7-1 : Automate à états finis des variantes orthographiques du mot tsar en français. ............... 127
Figure 7-2 : Grammaire morphologique "_Tsar.nom" du mot tsar (Silberztein M.). .......................... 127
Figure 7-3 : Déterminants numéraux de 100 à 999 en français (Silberztein M.). ............................... 128
Figure 7-4 : Quelques adverbes de temps en français ........................................................................... 129
Figure 7-5 : Une grammaire locale pour le terme "Carte" ................................................................... 132
Figure 7-6 : Concordance dans le fichier "text1-chantal.not" pour le terme "Opérateur". ............... 133
Figure 7-7 : Grammaire locale "Opérateur". ......................................................................................... 134
Figure 7-8 : Annotation1 du texte "dic-lhomme.not" .......................................................................... 135
Figure 7-9 : Un extrait des ressources syntaxiques pour les textes d’informatique. ............................ 136
Figure 7-10 : Annotation2 du texte "dic-lhomme.not" ....................................................................... 137
Figure 7-11 : Un extrait du dictionnaire des mots-clés "concepts_info". ............................................. 139
Figure 7-12 : Grammaire locale du terme "mémoire" ........................................................................ 141
Figure 7-13 : Grammaire locale du terme "Registre" ......................................................................... 142
Figure 8-1 : Fenêtre de la structure des annotations d’un texte (TAS). ................................................ 145
Figure 8-2 : Liste des annotations regroupées dans le dictionnaire "annotations_phrase.dic". ......... 147
Figure 8-3 : Fenêtres de recherche linguistique "Locate programmer" et table de concordance de la requête .................................................................................................................................................. 149
Figure 8-4 : Fenêtres de recherche linguistique "Locate <N><A>" et table de concordance de la requête .................................................................................................................................................. 150
Figure 8-5 : Fenêtre de concordance ..................................................................................................... 150
Figure 8-6 : Les fichiers constituant le corpus d’informatique constitué ........................................... 152
Figure 8-7: Analyse linguistique du corpus d’informatique ............................................................... 153
Figure 8-8 : Les fichiers constituant le corpus2 d’informatique "textes-chantal.noc" .................... 153
Figure 8-9 : liste des formes dans le corpus. .................................................................154
Figure 8-10 : table de concordances du token sélectionné "disques". ..............................155
Figure 8-11 : Affichage, dans son texte source, du token sélectionné "disque". ..................156
Figure 8-12 : Les fichiers du corpus traité et temps d'exécution de l'analyse linguistique. ....157
Figure 8-13 : Patron syntaxique pour "N1 Mod1 et Mod2 " .............................................159
Figure 8-14 : Termes extraits ("outputs") pour "mémoire de masse et auxiliaire" ..................160
Figure 8-15 : Grammaire locale pour "(N1 et N2 Mod)" ...................................................160
Figure 8-16 : Résultats de l'extraction ("outputs") pour " (N1 et N2 Mod) ". ...................... 161
Figure 8-17 : Localisation de termes avec le patron syntaxique "<N><A>" .............................162
Figure 8-18 : Les trois textes testés avec NooJ. ...............................................................163
Figure 8-19 : Grammaire locale appliquée pour l'extraction avec NooJ. ......................... 164
Figure 8-20 : Extraction des termes de longueur deux et trois avec NooJ. ......................165
Figure 8-21 : statistiques obtenues avec NooJ, pour rechercher les termes de longueur 2 et 3. 167
Figure 8-22 : Graphique des statistiques obtenues pour la colonne SS. ..........................168
Figure 8-23 : Concordance du texte en appliquant le dictionnaire "concepts_info" ................169
Figure 8-24 : Exportation de la concordance du texte......................................................170
Liste des tableaux

Tableau 1-1: Récapitulatif des outils d’extraction .................................................................33
Tableau 2-1 : Liste des classes syntaxiques de longueur 2 avec exemples. .............................................38
Tableau 2-2 : Liste des classes syntaxiques de longueur 3 avec exemples. .............................................38
Tableau 2-3 : Genre et nombre des noms composés (Silberztein M. 1990). ...........................................39
Tableau 2-4 : Liste des abréviations. .................................................................................................47
Tableau 5-1: Tête des mots composés. ...............................................................................................92
Tableau 5-2 : Codes grammaticaux utilisés .........................................................................................94
Tableau 5-3 : Description des classes syntaxiques pour les composés de longueur 2. .........................95
Tableau 5-4 : Description des classes syntaxiques pour les composés de longueur 3. .........................95
Tableau 6-1 : Genre et nombre des noms composés (Silberztein M. 1990). .......................................117
Tableau 6-2 : Extrait explicatif du fichier "composes-flx.nof". .........................................................123
Tableau 8-1 : Résultats trouvés ..........................................................................................................157
Tableau 8-2 : résultats obtenus. ..........................................................................................................158
Tableau 8-3 : Résultats obtenus avec NooJ. .........................................................................................162
Tableau 8-4 : résultats obtenus avec NooJ ..........................................................................................165
Tableau 8-5 : Résultats obtenus avec ANA. ..........................................................................................166
Introduction
Introduction

Problématique

Cette thèse s’inscrit dans le cadre général du Traitement Automatique des Langues (TAL). Cette discipline se situe à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle ; elle concerne la conception et le développement de programmes et techniques informatiques capables de traiter automatiquement des données exprimées dans une langue. Les applications liées au TAL ont fait l’objet d’une attention toute particulière depuis plusieurs décennies. Les langues naturelles ont été surtout privilégiées, mais actuellement un grand intérêt est accordé aux langues de spécialité avec la construction de ressources terminologiques.

Les besoins de plus en plus importants et variés en ressources terminologiques sont dus à la production toujours croissante de documents sous format électronique, l’internationalisation des échanges, les enjeux liés au développement d’internet et des réseaux intranet et les besoins des traducteurs. Il peut s’agir de simples listes de termes plus ou moins structurées (index structurés, thésaurus, réseaux lexicaux) utilisés par des systèmes d’indexation automatique ou de recherche d’information, ou de référentiels terminologiques plus documentés (Term, 2003).

La construction d’une terminologie dépend de l’application (Aussenac-Gilles et Condamines, 2002) dans laquelle on veut l’utiliser. Les termes retenus ainsi que leur degré de description sont différents selon que l’on veuille construire une terminologie de référence pour un système d’aide à la rédaction ou un réseau lexical pour améliorer la recherche d’informations sur internet.

Dans le domaine du traitement automatique des langues, des dictionnaires électroniques sont construits et utilisés. On notera qu’il existe différents types de dictionnaires. Les différents dictionnaires du commerce ne sont pas tous équivalents car chacun a une couverture lexicale limitée liée aux objectifs visés par l’éditeur.

Les dictionnaires d'usage, même transcrits sur support informatique, sont destinés à des lecteurs humains et sont orientés vers la définition des mots et la signification de leur utilisation. Les dictionnaires traditionnels informatisés ne sont pas adaptés aux besoins de la machine. Il est fait appel au lecteur pour ses connaissances du monde, sa faculté d'interprétation des définitions, sa compréhension des mots par analogie avec d'autres mots ou avec des mécanismes de création connus.

Les dictionnaires électroniques sont construits pour l'ordinateur et sont focalisés sur la description formelle des objets de la langue et leur classification systématique. Ce sont donc des ensembles très spécifiques de données qui sont ainsi élaborés. On entend aussi par dictionnaires électroniques des bases de données lexicales où toutes les informations sont explicites car elles sont destinées à l’usage des programmes informatiques. Ces bases visent la modélisation de la langue, ce qui les distingue des lexiques électroniques créés pour les besoins d’applications particulières.

Lors de l’analyse automatique de textes, la première tâche est la consultation des mots dans le dictionnaire. L’absence d'un seul mot entraîne l'échec de la consultation, aussi tous les mots valides rencontrés dans des textes variés doivent être représentés, avec une large couverture lexicale, en entrée de dictionnaire électronique mis en place. Ce qui signifie que les informations associées à chaque mot d'entrée doivent toutes être explicites.
Objectifs

Le cadre de travail est le système NooJ, nous nous intéressons à l’analyse automatique d’une langue technique, qui est l’informatique, en français.

Nous avons entrepris la construction d’un ensemble de ressources linguistiques pour la terminologie informatique afin d’implémenter dans l’environnement du système NooJ une composante d’analyse morpho-syntaxique automatique et reconnaître les mots composés d’informatique, les sigles et abréviations. Cette composante, qui comporte la description de son vocabulaire et de sa morpho-syntaxe, permet de mieux comprendre la langue technique.

Aussi, dans cette thèse, notre objectif est de mettre en place les ressources indispensables pour l’analyse de textes. Pour cela, nous construisons, le dictionnaire français des termes composés de l’informatique et les dictionnaires des sigles et abréviations, ainsi que les grammaires locales des termes d’informatique.

Les dictionnaires sont construits à partir de termes recensés et extraits à partir de textes, aussi nous nous intéressons à l’extraction de terminologie.

Plan de la thèse

Le premier chapitre de cette thèse fait un état de l’art sur l’extraction automatique de terminologie. L’extraction manuelle est examinée, puis les différentes approches d’extraction (linguistiques, statistiques, mixtes) sont présentées, avec des exemples de systèmes. Un tableau récapitulatif permet d’avoir une synthèse de la plupart des systèmes existants.

Le deuxième chapitre présente les dictionnaires électroniques et les bases de données terminologiques. Les dictionnaires électroniques des mots simples et des mots composés du système DELA sont décrits plus en détail. Nous recensons ensuite les dictionnaires de terminologie et les bases de données terminologiques, disponibles sur Internet, relatifs à divers domaines de spécialité ; les domaines sont l’informatique, l’internet, la médecine, l’économie, etc. Enfin une présentation des caractéristiques principales est faite pour les dictionnaires et bases de données terminologiques telles qu’IATE, OSINET, le GDT, ALEXANDRIA, etc.

Le chapitre 3, décrit l’environnement de recherche qui est le système NooJ. La plateforme de développement linguistique NooJ est utilisée comme outil de développement de nos dictionnaires. Le passage d’INTEX à NooJ a permis à leur auteur Silberztein M. de développer un environnement linguistique robuste de développement de ressources linguistiques à large couverture. Le système NooJ est présenté avec ses principales fonctionnalités, permettant la description de morphologie flexionnelle, dérivationnelle, la construction de corpus, la construction des dictionnaires électroniques, l’annotation, l’analyse linguistique de textes.

Dans le chapitre 4, les aspects linguistiques de la terminologie sont abordés, avec un ensemble de définitions : mot, terme, mot composé, synapsie, etc., ainsi qu’un examen des critères ou règles permettant de décider si un groupe nominal est libre ou figé, donc pourra être retenu ou non comme entrée de dictionnaire. Cette analyse des critères nous sera utile pour la construction de notre dictionnaire des termes composés d’informatique.
Le cinquième chapitre porte sur l’élaboration de nos dictionnaires électroniques de terminologie. Dans NooJ, les dictionnaires de spécialité sont pratiquement inexistants, nous nous intéressons à la terminologie informatique en français. Cette partie est consacrée à la présentation de l’organisation et du contenu du dictionnaire de la langue technique que nous construisons et appelons "info_comp". Nous commençons par une étude et classification syntaxique des termes d’informatique. Les classes recensées sont décrites avec des exemples. Une description du format d’une entrée est faite. Nous effectuons Une étude des traits sémantiques des termes d’informatique, différentes classes sont mises en place. Les dictionnaires de sigles et d’abréviations sont décrits et seront séparées des termes composés. Les versions compilées des dictionnaires sont utilisées dans les étapes ultérieures d’analyse linguistique.

Dans le chapitre 6, la morphologie est abordée. Les outils de flexion et dérivation de Nooj sont décrits. Nous étudions les modes de flexion, de dérivation des mots composés d’informatique. Nous élaborons les règles de flexion des mots composés d’informatique, en liaison avec les entrées du dictionnaire, "info_comp" (le paradigme "FLX" se trouvant dans l’entrée fait appel à la règle de flexion associée). Une étude de l’interaction entre les entrées lexicales décrites dans le dictionnaire (au chapitre 5) et les descriptions formalisées des règles de flexion, doit être faite afin de pouvoir générer automatiquement toutes les formes fléchies potentielles.

Dans le chapitre 7, concernant les grammaires locales, les automates à états finis sont abordés. Les outils de base utilisés pour construire les descriptions formalisées à large couverture des langues naturelles (dictionnaires et grammaires électroniques) sont les automates à états finis. Les automates à états finis, déterministes, acycliques et minimaux se sont très largement répandus en TAL pour la représentation et le stockage de gros volumes de données sous forme de dictionnaires électroniques. Nous construisons les grammaires locales des termes de l’informatique de deux manières : après recensement manuel à l’aide de textes écrits, ou bien après concordance, à partir d’un fichier. Nous proposons des exemples de grammaires locales de termes d’informatique, élaborées manuellement et construites graphiquement grâce à l’éditeur graphique de grammaires de NooJ.

Chapitre 1 :
Etat de l’art sur l’extraction automatique de terminologie
1 Etat de l’art sur l'extraction automatique de terminologie

1.1 Introduction

L'extraction de terminologie consiste à identifier des termes potentiels dans un texte spécifique ou un ensemble de textes (corpus) ainsi que les informations pertinentes liées à l'emploi de ces termes ou aux concepts auxquels ils renvoient (définition, contexte, etc.). L'extraction de terminologie s'accomplit manuellement ou bien automatiquement grâce à l'utilisation d'outils d'extraction terminologique. L'extraction automatique de terminologie est un moyen rapide d'acquérir des connaissances sur un domaine et le langage spécialisé qui s'y rattache. Cependant, après l'extraction de termes potentiels par les logiciels, des spécialistes doivent décider si les résultats obtenus sont adaptés ou non.

Une caractéristique importante de la terminologie est la multiplicité de ses utilisateurs. Aux usagers traditionnels que sont les traducteurs et les rédacteurs techniques s'ajoutent de nos jours de nombreuses catégories d'usagers humains ou d'usagers machines tels que systèmes experts, aides à la traduction, aides à la rédaction, gestion des données documentaires et textuelles, bases de connaissances, outils d’indexation, etc. Cette multiplicité d’usagers entraîne la conception de gammes variées de produits terminologiques (TIA, 99).

L'extraction de terminologie est une part intégrante du travail terminologique et représente une étape importante lors de la création de bases de données terminologiques. Elle permet une identification rapide de termes susceptibles d'intéresser un spécialiste des langues. Elle permet également une intégration de termes dans une base de données.

1.2 L’extraction de terminologie

Il existe deux procédés d’extraction de terminologie : manuel et automatique.

1.2.1 Extraction manuelle

L’extraction manuelle de terminologie comprend trois grandes étapes : la collecte de textes écrits, la lecture des textes et l'extraction et enfin l'acquisition.

1.2.1.1 La collecte de textes écrits

Les textes doivent être dans le domaine de la spécialité retenue, les textes seront trouvés dans diverses sources :
- livres,
- journaux,
- dictionnaires,
- etc.
1.2.1.2 La lecture des textes et l’extraction

La lecture se fait phrase par phrase en faisant un repérage des termes du domaine en :
- Se focalisant sur les concepts du domaine ;
- Essayant de rattacher au concept des mots voisins.

Afin de décider si les groupes de mots trouvés sont des termes de la spécialité, on pourra utiliser les critères degré de figement et lexicalisabilité (quand ils s’appliquent) qui seront présentés au chapitre 4.

1.2.1.3 L’acquisition

L’acquisition consiste à ajouter l’entrée dans le dictionnaire, elle suit l’extraction des termes retenus comme valides. La lecture des textes est longue, mais tous les termes extraits par un spécialiste du domaine sont valides.

1.2.2 Extraction automatique

L’extraction automatique de terminologie est réalisée à l’aide d’outils utilisant différentes méthodes pour l’extraction.

Dans (L'Homme M.C. 2001) les techniques d'extraction des données terminologiques et leur impact sur le travail du terminographe sont examinées.

Dans la plupart des cas, une validation des termes extraits est faite par le spécialiste du domaine en vue de l’acquisition des termes retenus. Les candidats termes extraits sont examinés pour voir s’ils constituent une entrée (du dictionnaire électronique d’informatique dans notre cas), donc si ce sont des termes du domaine de spécialité).

1.3 Les outils d’extraction automatique de terminologie

Ils permettent d’extraire de nouveaux termes à partir de textes ou corpus. Les méthodes linguistiques, statistiques ou mixtes sont utilisées pour développer les outils d’extraction automatique.

Pour les traducteurs, l’apport d’un logiciel d’extraction de terminologie est indéniable quand on sait le peu de temps dont ils disposent pour alimenter une base de données terminologique. Ce type d’extraction offre une possibilité de relever des syntagmes nominaux qu’un traducteur confirmé n’aurait pas retenus comme entrées de base, les jugeant trop simples.

Un recensement des différents systèmes existants est fait dans (De Chalendar G. 2002). Quelques critères d’évaluation de logiciels d’extraction sont examinés dans (L'homme, 2000). Dans (Drouin, 2003), des techniques courantes d’extraction de termes sont examinées. Dans la thèse de (Drouin, 2002) et dans (Drouin, 2003b) des pivots lexicaux spécialisés sont utilisés pour l’acquisition automatique de termes. Dans (Drouin, 2003a) un corpus non technique est utilisé pour l’extraction de terminologie. Une évaluation d’outils d’acquisition

Il existe différents travaux dans le cadre des corpus spécialisés en terminologie et l’acquisition automatique de collocations¹ (Lemay, 2003), (Orliac, 2003) et (Rochibeau, 2003).

Il convient (L’Homme M.C., 2000) de souligner que le repérage des collocations est compliqué notamment par le fait que les mots les composant peuvent être permutés ou séparés par plusieurs autres mots. Il est noté que les cloisons entre extracteurs de collocations et extracteurs de termes ne sont pas étanches : les extracteurs de collocations relèvent des termes complexes; les extracteurs de termes complexes relèvent forcément des collocations. Ce chevauchement s’explique en grande partie par la similitude des techniques utilisées.

La plupart des auteurs (De Chalendar, 2002) estiment indispensable de maintenir une activité humaine dans les systèmes d’acquisition, avec une acceptation ou un refus des résultats mais aussi jusqu’à être le centre de l’acquisition, le traitement informatique ne sera alors qu’un outil de présentation et d’enregistrement des données.

1.4 Les différentes approches

Les méthodes d’extraction peuvent être classées en trois catégories : les méthodes linguistiques, les méthodes statistiques et les méthodes mixtes.

1.4.1 Les méthodes linguistiques

Elles sont aussi appelées méthodes symboliques. Ces méthodes sont essentiellement syntaxiques et s’appuient sur une analyse syntaxique des textes, mais il existe aussi des méthodes basées sur la sémantique (Morin E., 1999).

Un étiquetage morpho-syntaxique préalable permet de repérer les syntagmes nominaux, de les analyser pour dégager les candidats termes. Les candidats termes sont soumis à l’expert pour retenir les mots composés qui sont des termes.

Quatre systèmes sont recensés dans (Term, 2003), nous avons TERMINO (David et al., 1990), LEXTER (Bourigault, 1994), FASTER (Jacquemin, 1997), et XTERM développé par Cerbah, F. développé en 1999. Le système LEXPRO (Savary, 2000), (Chrobot A. 1999), utilise les dictionnaires DELAS et DELAC, élaborés au LADL pour l’anglais, ainsi que le système INTEX (Silberztein M. 1993) pour acquérir des termes anglais de l’informatique.

Des travaux axés sur une approche sémantique et l’acquisition de terminologie peuvent être trouvés dans (De Chalendar G. 2001). Nous présentons ici quelques systèmes utilisant les méthodes linguistiques.

1.4.1.1 TERMINO

¹ Linguistiquement les collocations peuvent être décrites comme des associations syntagmatiques binaires restreintes, semi-figées et fortement dépendantes du contexte d’utilisation (Grossman et Tutin 2003).
Termino de David et Plante, (David et al., 1990,1996), fut, une des premières applications d'acquisition automatique de termes.

Termino est un logiciel de dépouillement terminologique assisté par ordinateur, développé par le groupe de Recherche et Développement en Linguistique Computationnelle (RDLC).

La théorie linguistique sur laquelle est fondé le logiciel propose que la grammaire soit organisée en composants autonomes et non hiérarchisés (en d’autres termes, les unités d’un composant et les règles qui lui sont associées sont spécifiques). Parmi ces composants, celui de la morphologie et celui de la syntaxe sont susceptibles de construire des unités lexicales. On distingue dans ce modèle :

- un ensemble de positions, c’est-à-dire un ensemble de points hiérarchiquement organisés entre eux et qui forment une configuration,

- une relation d’occupation entre les positions et les constituants lexicaux.

L’objet de la syntaxe est de rendre compte des positions (identification des positions et de leurs propriétés spécifiques) et de l’occupation de ces positions. Les relations entre les éléments lexicaux ne relèvent pas du composant syntaxique mais du composant lexical.

Le système TERMINO (selon, Enguehard C. 1992) n’utilise pas de connaissances lexicales (pas de dictionnaire). Le but est de trouver des unités nominales composées de plusieurs mots pour lesquelles on peut faire l’hypothèse d’une construction syntaxique.

Ces unités terminologiques recherchées sont appelées des synapsies. (Benveniste E., 1966,1974) nomme les unités complexes des synapsies. Il s’agit d’unités polylexicales caractérisées par une structure hiérarchique interne et qui occupent une position noyau en tant que groupe nominal. Ces groupes nominaux n’incluent aucune marque de localisation, ni de déterminant possessif, ni de démonstratif ou de modificateur pour l’adjectif.

**Exemples** :
- un nœud coulant,
- allocation dynamique de la mémoire,
- *le chat de ma voisine* ²
- pomme de terre,
- mur du son.

TERMINO se concentre sur les syntagmes nominaux qui sont les seules structures à même de produire des termes. Le programme commence par prétraiter le corpus pour enlever tous les caractères de formatage et ne conserver que les textes eux-mêmes.

² * indique que ce n’est pas une synapsie.
Ensuite commence le véritable travail d'extraction par une analyse morphologique à base de règles, suivie de l'analyse des syntagmes nominaux à l'aide d'une grammaire. Vient ensuite la génération des synapsies à partir de l'analyse des têtes et compléments, ce qui permet de lever un certain nombre d'ambiguïtés.

Enfin, une interface graphique permet de gérer une base de données terminologique constituée à l'aide des termes extraits. TERMINO est souvent cité comme un outil fondateur du domaine.

1.4.1.2 LEXTER

LEXTER développé par Didier Bourigault, (Bourigault, 1994, Assadi et al., 1996), dans le cadre d'un projet de gestion de la documentation technique de l'entreprise EDF France est dédié à l'aide pour l'acquisition de terminologie, il repose sur la constatation qu'un certain nombre de constituants ne peuvent faire partie d'un terme. Parmi ces constituants, on trouve les mots de certaines parties du discours (comme les verbes, les pronoms ou les adverbes) ou encore la ponctuation. Par conséquent, ces constituants forment des frontières que les termes ne peuvent franchir.

Ce système a été initialement conçu pour constituer et mettre à jour des thésaurus utilisés par un système d'indexation automatique de textes. Il reçoit en entrée un corpus de textes techniques portant sur un domaine quelconque et fournit en sortie un réseau de candidats termes sous forme d'hypertexte terminologique c'est-à-dire des noms ou groupes de noms susceptibles de désigner des concepts du domaine. Chacun des candidats termes trouvés lors des étapes précédentes (syntagmes nominaux et leurs composants) est relié aux candidats termes dont il est tête ou expansion. Les textes traités ont été soumis à une analyse morphologique : chaque mot est étiqueté avec sa catégorie grammaticale.

Il repère les syntagmes nominaux maximaux en s’appuyant sur les marqueurs de frontières qui sont tous les éléments qui ne peuvent pas faire partie d’un syntagme nominal: déterminants, verbes conjugués, pronoms etc., ainsi que les groupes grammaticaux vides sont retirés. Les syntagmes maximaux (candidats termes) sont décomposés de façon binaire en une tête et une expansion.

LEXTER utilise un étiqueteur externe pour identifier les parties du discours des mots du corpus. Ensuite, il repère les groupes nominaux bornés par des frontières. Ces groupes nominaux sont dits maximaux. La décomposition se fait par une analyse syntaxique locale : des règles de grammaire déterminent quelles sont les structures pouvant correspondre à la tête et à l'expansion.

Instruction du dossier de divorce

| tête | expansion |

Les groupes nominaux de longueur maximale sont ensuite analysés dans le but d’obtenir des sous-groupes. Les modules de LEXTER sont composés de règles qui indiquent quels sous-groupes sont à extraire de la structure grammaticale de base.

Disque dur de la station de travail
On obtient :
- disque dur,
- station de travail,
- disque dur de la station de travail.

Les cas d'ambiguïté de rattachement sont traités par des techniques d'apprentissage endogène sur le corpus. Enfin, un réseau dit terminologique est créé en reliant chaque candidat terme à sa tête, à son expansion et aux candidats termes dont il est lui même tête ou expansion. Il est important de noter que ce réseau est organisé en fonction de critères syntaxiques et non conceptuels.

Évidemment, tous ces syntagmes ne sont pas des termes et un filtrage doit être effectué par un expert (spécialiste du domaine, terminologue ou cogniticien). Les candidats et leur structuration en réseau ne sont dédiés qu'à être une aide à l'élaboration d'une terminologie.

Utilisé pour l'extraction de termes dans le projet SADE (pour faciliter la modélisation du domaine), LEXTER est utilisé principalement dans des applications de gestion électronique de documents techniques pour la constitution semi-automatique d'index terminologiques.

1.4.1.3 FASTER

FASTER, de Christian Jacquemin (Jacquemin C. 1997) est un outil de repérage de variations de termes. Il a besoin de données terminologiques initiales. Il permet de repérer dans un corpus les variantes de termes attestés. Des métarègles sont utilisées pour repérer :

- La coordination de termes, avec la combinaison de deux termes avec un mot tête commun ou un argument commun. Par exemple, le terme : "modulation de fréquence et d'amplitude" correspond aux deux termes "modulation de fréquence" et "modulation d'amplitude" ;

- Les modifications/substitutions, la substitution étant le remplacement d'un mot par un syntagme, la modification est l'insertion d'un modificateur sans référence à un autre terme. Par exemple "maladie cardiaque congénitale" est une variante du terme "maladie congénitale" ;

- Les permutations : ce type de variante est rare en français mais fréquente en anglais, par exemple "blood flow" et "flow blood" ;

- Les variantes morpho-syntaxiques où il y a modification morphologique d’un des deux éléments, par exemple "sol granitique" et "sol de granit" ;

- Les variantes sémantico-syntaxiques où les termes sont en relation sémantique (synonyme, hyponyme), un mot de syntagme pouvant être remplacé par un synonyme.

---

3 Projet SADE (1993 -EDF-GDF : Acquisition et modélisation de connaissances juridiques pour la gestion de prêts financiers
Par exemple "évaluation de flux" est une variante sémantique de "mesure de flux".

Christian Jacquemin part de la constatation qu'un même terme peut apparaître sous des formes variées dans un corpus. FASTER permet de repérer des variantes de termes. Mais FASTER peut aussi servir à acquérir de nouveaux termes simples par un processus inverse. Parmi ces termes simples, ceux qui ne font pas partie des termes d'origine peuvent être proposés comme nouveaux termes pour évaluation. Christian Jacquemin dans (Jacquemin, 1996) parle de près de 85% de précision globale sur un corpus en anglais.

Le système détecte les variantes des termes en corpus (formes linguistiques différentes). Il capture les variations par des métarègles opérant à trois niveaux :

- Morpho-syntaxique : "flux de sève mesurés" est une variation de "mesure de flux" s'appuyant sur le lien morphologique entre le verbe mesurer et le nom mesure ;
- Syntaxique : "mesure de volume et de flux" est une variante de "mesure de flux" obtenue par coordination ;
- Syntaxico-sémantique : "évaluation de flux" est une variante de "mesure de flux" s’appuyant sur le lien sémantique existant entre les noms évaluation et mesure.

1.4.1.4 XTERM

Le système développé (Cerbah F. 1999) effectue tout d’abord un prétraitement du corpus afin de repérer les parties textuelles puis un étiquetage morphosyntaxique des segments textuels est réalisé. Un repérage des candidats termes est ensuite effectué par projection de patrons syntaxiques propres aux termes sous forme d’automates d’état fini. Une réorganisation hiérarchique de ces candidats termes est faite en fonction de leurs composantes (tête, expansion). Pour terminer les variantes sont regroupées à l’aide de règles.

1.4.1.5 TERMINAE

TERMINAE (S. Szulman 1999) est un outil d’aide à la construction de terminologie et d’ontologies à partir de l’étude de textes, il utilise les méthodes et outils linguistiques. Une terminologie est ici constituée d’un ensemble de termes dont les caractéristiques linguistiques sont décrites dans des fiches et dont les relations entre sens des termes sont représentés dans un réseau sémantique. Une ontologie est ici un ensemble structuré de concepts définis formellement et dont certains sont étiquetés par des termes. Une fiche terminologique est créée pour un terme et ses notions associées. Chaque notion est décrite à l’aide de synonymes et des termes proches, d’un ensemble de cooccurrences associées, d’une définition et éventuellement d’informations lexicales.

1.4.2 Les méthodes statistiques

Elles permettent d’extraire des candidats termes sans analyse linguistique préalable. Les méthodes statistiques sont devenues très présentes dans le traitement du langage naturel. La plupart d’entre elles se résument en calculs de valeurs numériques tels que les fréquences et
les critères d'association. Elles sont fondées sur le fait que les termes apparaissent fréquemment dans les textes spécialisés : des mots simples qui apparaissent fréquemment ensemble sont forcément significatifs. Dans ce cadre il y a des outils qui se basent sur les syntagmes nominaux, les segments répétés et les schémas productifs. Parmi ces outils on trouve ANA (Enguehard C. 1992), Mantex (Oueslati R. 1999) et LIKES (Frath et al. 2000).

1.4.2.1 ANA (Acquisition Naturelle Automatique)

A partir de textes techniques ou scientifiques, le système ANA de Chantal Enguehard (Enguehard C. 1992) extrait automatiquement des concepts pour produire un réseau sémantique, il se base sur le principe de schémas productifs.

Il est intéressant de copier l'apprentissage du langage par les enfants. Celui-ci se fait en grande partie par l'association de sons avec des perceptions puis par induction et généralisation de ces associations.

Le système filtre statistiquement les mots fonctionnels du corpus (la liste obtenue recouvre très certainement en grande partie les listes de "mots vides" existantes), il détermine automatiquement une liste restreinte de mots représentant des concepts du domaine (l'amorce du système) en détectant des cooccurrences de ces mots amorces séparés par un mot fonctionnel. Cela forme de nouveaux concepts. De plus les mots fonctionnels utilisés dans ce cadre sont enregistrés dans une liste appelés «mots de schémas ». Enfin ANA détecte de nouveaux concepts, relève les mots qui co-occurent avec un concept déjà enregistré en utilisant les connaissances prélablement sélectionnées.

Le système se compose de deux modules (Enguehard C. 1992). Le premier module est une étape de familiarisation avec le domaine et la langue, à partir de l'étude d’un échantillon de textes. Le système met à jour les connaissances nécessaires à l’induction de concepts :

- une liste de « mots fonctionnels» (mots non porteurs : articles, préposition etc.),
- une liste de mots liés (variation morphologique des mots simples),
- une liste de mots de schémas qui traduisent la structure interne des concepts,
- et surtout le "bootstrap" (une liste d’environ 50 concepts) d’initialisation du processus de découverte de nouveaux concepts.

Le deuxième module effectue la découverte de nouveaux concepts en utilisant les connaissances précédemment sélectionnées.

Dans tous les cas les nouveaux concepts ne sont retenus que s'ils ont un nombre d'occurrences suffisant dépassant des seuils numériques fixés à l'avance. Le programme reprend les schémas les plus productifs dans le corpus. ANA construit un réseau de termes de façon entièrement automatique. L'apprentissage effectué ici est appelé apprentissage « endogène ».
1.4.2.2 MANTEX

MANTEX, (Oueslati, 1999) et (Frath et al., 2000), est fondé sur le principe des segments répétés qu’il repère dans un corpus. Les phrases du corpus sont traitées l'une après l'autre. Les mots pris en compte sont des versions partiellement lemmatisées des mots de la phrase. Cette lemmatisation se fait simplement en supprimant les marques du pluriel s et x sur les mots dont le radical de quatre lettres est possédé par un autre mot du corpus et si ce mot possède la même dernière lettre que le mot en question, s et x mis à part.

Un des plus gros problèmes de cette approche, que l'on retrouve d'ailleurs dans toutes les approches numériques, est que les termes n'ayant qu'une seule occurrence dans le corpus (hapax) ne peuvent pas être acquis. Or, les hapax sont une source importante de connaissances. L'auteur remarque que la plupart des segments répétés retenus sont des syntagmes nominaux.

1.4.2.3 LIKES

LIKES (Frath et al., 2000) est une station de travail pour l'acquisition de connaissances à partir de textes, écrite par François Rousselot. Elle permet la création de terminologie et d’ontologies. LIKES a été réalisée au départ pour analyser des corpus en vue de rechercher des candidats termes, et travaille sans aucun dictionnaire pour trouver des segments répétés. Cette technique est facilement transportable d’une langue à l’autre pour toutes les langues non agglutinantes.

LIKES reprend et améliore les caractéristiques de MANTEX et en comprend d'autres. Par exemple, LIKES produit une liste de segments composés, c'est-à-dire des segments proches. Les segments répétés fournissent des candidats termes qui sont ensuite examinés et éventuellement promus au rang de termes.

1.4.3 Les méthodes mixtes

La tendance actuelle consiste à combiner des approches linguistiques avec des approches statistiques.

Généralement la partie essentielle de la méthode d’extraction est statistique, la partie « linguistique » consistant à filtrer les termes en fonction de leur catégorie syntaxique et parmi les outils qui utilisent cette approche nous avons ACABIT (Daille, 94), ASIUM (Faure D., Nédellec C., 1998,1998a), et XTRACT (Smadja F. 1993, 1996).

1.4.3.1 ACABIT (Automatic Corpus Based Acquisition of Binary terms)

ACABIT, de Béatrice Daille (Daille, 94) travaille à partir d'un corpus pré-étiqueté et effectue une analyse syntaxique suivie d'un traitement statistique.

ACABIT commence d’abord par collecter des schémas syntaxiques de termes simples (N N, N à Det N, etc.) et des mécanismes de variation permettent d'obtenir des termes plus complexes. Le système utilise un ensemble de transducteurs pour extraire les composés formés selon les schémas précédents et ne conserve que les formes lemmatisées.
ACABIT repose sur l'utilisation de diverses mesures statistiques qui retiennent le mieux les candidats-termes sans être sensibles aux fréquences.

1.4.3.2 ASIUM

ASIUM, de David Faure (Faure D., Nédellec C., 1998, 1998a) apprend des concepts faits de l'agrégation de mots simples. ASIUM propose de manière automatique des classes de termes\footnote{ASIUM est vu plutôt par d’autres auteurs comme outil de structuration de termes, étant donné qu’il propose des classes de termes qui seront validées par un expert, nous le classons avec les outils d’extraction.}. Il repère dans les textes des associations récurrentes de verbes et de noms, apprend des cadres de sous-catégorisation de verbes et une ontologie en utilisant le résultat d’un analyseur syntaxique et un algorithme de regroupement conceptuel pour détecter les associations entre un verbe et les noms têtes de ses sujets, compléments d'objet direct et autres compléments reliés aux verbes par une préposition.


1.4.3.3 XTRACT

XTRACT, de Franck Smadja (Smadja, 1993) est d'abord un extracteur de cooccurrences mais peut être utilisé en acquisition de terminologie. Il utilise une technique statistique pour détecter les cooccurrences et une analyse syntaxique pour les classifier.

XTRACT effectue une extraction de collocations (avec une visée générale), il a recours à des calculs inspirés de l’information mutuelle assortis d’un marquage des fonctions des mots (MC. L’homme 2000). C’est un outil plus générique de repérage des collocations et pas seulement des termes. Il effectue :

- Un repérage des couples de mots co-occurrents fréquemment dans le corpus ;
- Une recherche des collocations entre ces couples et d’autres mots pour repérer des collocations de plus de deux mots ;
- Ces collocations sont étiquetées par un analyseur morpho-syntaxique et une classification est ensuite faite.

Pour un mot donné, et pour chacune de ses occurrences, la détection se fait en notant le nombre d'occurrences de chacun de ses voisins dans une fenêtre de onze mots (dont cinq à gauche et cinq à droite du mot cible).

Les cooccurrences dont la fréquence dépasse d'une manière statistiquement significative la fréquence due au hasard sont sélectionnées. Cette recherche est relancée récursivement sur les phrases comportant ces cooccurrences significatives pour extraire des cooccurrences de plus de deux mots.
Les cooccurrences collectées à cette étape représentent plusieurs phénomènes différents comme :
- verbe support et nom prédicatif ("make decision", prendre une décision),
- verbe et objet typique ("reach an agreement"),
- ou encore syntagmes figés ("stock market").

Pour pouvoir différencier ces relations, le système utilise un analyseur qui prend en entrée des phrases dont les mots sont pré-étiquetés, puis il utilise des filtres de niveau de plus en plus élevé pour corriger les erreurs éventuelles des niveaux précédents et marquer de nouveaux phénomènes syntaxiques (syntagmes nominaux puis verbaux, sujet et prédicat).

Il est vrai que pour obtenir une évaluation significative, il faut disposer d'un corpus suffisamment étendu pour que tous les phénomènes désirés apparaissent suffisamment souvent.

Nous n'avons trouvé dans nos recherches aucune évaluation comparée du coût global des divers systèmes, certainement parce qu'aucun n'est réellement entré en phase de commercialisation active et que cette évaluation elle-même aurait un coût important. Ce sont d'ailleurs souvent des résultats de coopérations entre industriels ayant un besoin particulier et des équipes de recherche.

Dans le cas de ces outils d’extraction automatique, il faut noter que les candidats termes sont filtrés et validés par l’expert humain.

### 1.5 Tableau récapitulatif

Dans la section 1.4 nous avons présenté les outils principaux d’extraction automatique de terminologie, dans le tableau 1.1 suivant (inspiré du tableau présenté dans (Roche M. 2004) que nous avons étendu avec la colonne caractéristique et l’ajout d’autres systèmes), nous présentons un état récapitulatif des outils d’extraction automatique de terminologie avec leurs caractéristiques et des références bibliographiques.

Dans ce tableau "L" signifie "méthode linguistique", "S" : méthode statistique et "M" : méthode mixte.

<table>
<thead>
<tr>
<th>Systèmes</th>
<th>Caractéristiques</th>
<th>L</th>
<th>S</th>
<th>M</th>
<th>références</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOMINO (anc.) TERMINO</td>
<td>extraction de termes complexes français et catégorisation des autres mots pleins.</td>
<td>X</td>
<td></td>
<td></td>
<td>David et Plante 1990</td>
</tr>
<tr>
<td>ANA</td>
<td>extrait automatiquement des concepts pour produire un réseau sémantique</td>
<td></td>
<td>X</td>
<td></td>
<td>Enguehard 1993</td>
</tr>
<tr>
<td>LEXTER</td>
<td>Syntagmes nominaux maximaux</td>
<td></td>
<td></td>
<td>X</td>
<td>Bourigault 1993</td>
</tr>
<tr>
<td>TermPlus (anc. Notions)</td>
<td>extraction de termes complexes anglais et français avec exploration des termes reliés formellement</td>
<td></td>
<td></td>
<td></td>
<td>Drouin et Ladouceur 1994</td>
</tr>
<tr>
<td>Outil</td>
<td>Description</td>
<td>X</td>
<td>Année</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTRACT</td>
<td>Extraction de collocations (avec une visée générale)</td>
<td>X</td>
<td>Smadja 1993</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACABIT</td>
<td>Extraction de composés simples et complexes</td>
<td>X</td>
<td>Daille 1994</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERMS</td>
<td>extraction de termes anglais</td>
<td></td>
<td>Justeson et Katz 1995</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Système INTEX utilisé</td>
<td>Extraction de termes et identification des variations</td>
<td>X</td>
<td>San Juan I. 2001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLARIT</td>
<td>Extracteur de groupes nominaux</td>
<td>X</td>
<td>David A. Evans et Zhai 1996</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C/NC VALUE</td>
<td>Les termes retenus ont la plus grande C-value</td>
<td>X</td>
<td>Frantzi et al. 2000, Frantzi 1998</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERMIGHT</td>
<td>Les groupes nominaux simples et complexes ordonnés selon la fréquence du mot de tête</td>
<td>X</td>
<td>Dagan et Church 1997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIKES</td>
<td>produit une liste de segments composés</td>
<td>X</td>
<td>Frath et al., 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MANTEX</td>
<td>repère les segments répétés</td>
<td>X</td>
<td>Frath et al. 2000 Oueslati, 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNTAX</td>
<td>Extraction de syntagmes nominaux, verbaux</td>
<td>X</td>
<td>Bourigault et Fabre 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTEX</td>
<td>acquérir des termes anglais de l’informatique de LEXPRO DataBank</td>
<td>X</td>
<td>Savary, 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WASSPENCH</td>
<td>repèrent des patrons reflétant les constructions recherchées et épurent ces premiers résultats au moyen de calculs probabilistes</td>
<td>X</td>
<td>Kilgariff et Tugwel 2001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIPS</td>
<td>Éxtraction de collocations</td>
<td>X</td>
<td>Nerima et al. 2003, 2006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESATEC</td>
<td>Extraction de termes complexes combine des filtres numériques (approche bayésienne + N-grams de mots) et linguistiques</td>
<td>X</td>
<td>Biskri et al. 2004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTERM</td>
<td>Repérage des candidats termes à l'aide de patrons syntaxiques</td>
<td>X</td>
<td>Cerbah F. 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERMINAE</td>
<td>Construction de terminologies et ontologies</td>
<td>X</td>
<td>Szulman S. 1999</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tableau 1-1: Récapitulatif des outils d’extraction.**

### 1.6 Conclusion

Dans ce chapitre nous avons présenté l’extraction de terminologie. Nous nous intéresserons plus particulièrement à l’informatique. Notre objectif est la construction d’un dictionnaire de terminologie informatique qui sera vue au chapitre 5.

L’extraction manuelle de terminologie demande beaucoup de temps, les outils d’extraction automatique sont une aide au terminologue mais la plupart de ces systèmes nécessitent l’intervention humaine pour la validation des termes extraits. Nous constatons l’existence d’une grande variété d’outils d’extraction, mais la plupart ne sont ni commercialisés ni d’utilisation libre. Quand ils sont téléchargeables librement, leur utilisation n’est pas simple.
Chapitre 2 :
Dictionnaires électroniques et bases de données terminologiques
2 Les dictionnaires électroniques et les bases de données terminologiques

2.1 Introduction

Dans ce chapitre nous allons examiner quelques dictionnaires électroniques d’informatique, de spécialité et des bases de données terminologiques.

2.2 Le système DELA

2.2.1 Introduction

Les travaux effectués au LADL depuis 1980 ont permis de développer une famille d’outils de traitement automatique de textes en langue naturelle avec un ensemble de dictionnaires, de lexiques grammairais et le système Intex (Silberztein M. 1993) qui est un logiciel d’interrogation de textes.

La description des langues naturelles est formalisée dans le système DELA par des dictionnaires électroniques et des grammaires représentées par des ensembles de graphes. (Silberztein M. 1993)

Le système DELA est organisé en plusieurs modules. La répartition des données dans chaque module est effectuée d’après la forme des mots d’entrée et la nature des informations qui les accompagnent. La configuration générale du système se compose des ensembles suivants :

- Les dictionnaires de mots simples, DELAS et DELAF. Le DELAF contient les formes fléchies des mots simples du DELAS ;

- Les dictionnaires de mots composés, DELAC et DELACF. Le DELACF contient les formes fléchies des mots composés du DELAC ;

- Les dictionnaires phonémiques, DELAP et DELAPF ;

- Le dictionnaire des verbes DELAV ;

- Les tables syntaxiques regroupées en lexique-grammaire ;

- Les graphes et automates.

À ces ensembles sont associés des outils de traitements informatiques :

- Des programmes de génération de formes fléchies définis dans INTEX ;

- Un logiciel d’interrogation de textes élaboré par Max Silberztein diffusé sous le nom d’INTEX, où sont intégrées de nombreuses fonctionnalités (Silberztein M. 1993).
2.2.2 Les dictionnaires des mots simples DELAS

C’est le dictionnaire des mots simples du français (Courtois B. 1990a). Ce dictionnaire (Courtois B., Silberztein M. 1990) contient plus de 90000 mots simples sous leur forme canonique, à laquelle sont associées diverses informations linguistiques y compris des codes flexionnels. La flexion du DELAS opère suivant plus de 350 paradigmes différents, dont 150 verbaux, pour produire le DELAF. Les mots du DELAS sont mis sous leur forme canonique, dans l’ordre alphabétique. L’introduction d’un nouveau mot dans le DELAS doit être conforme à une syntaxe bien précise.

Le DELAF est le dictionnaire électronique des formes simples fléchies du français. Chaque forme d’entrée est identifiée par la forme canonique, son code morphologique, et des codes représentant :

- Genre et nombre pour les noms, adjectifs, mots grammaticaux de forme variable ;
- Personne, genre, nombre pour les pronoms personnels ;
- Mode, temps, personne, nombre pour les verbes.

Exemples :

- préceptrice, précepteur.N36(Hum) :fs,
- légaux, légal.A76 :mp,
- ils,.PRO(PpvIL) :3mp.

A partir du DELAS, le dictionnaire DELAF est construit automatiquement par un programme de génération de formes fléchies. Le code morphologique de chaque mot est utilisé pour retrouver la classe flexionnelle correspondante qui permet d’engendrer toutes les formes fléchies. Le DELAF comporte près de 750 000 formes simples fléchies.

2.2.3 Le dictionnaire des mots composés DELAC

Le DELAC est le dictionnaire électronique des mots composés avec leur morphologie. Un nom composé est constitué de mots simples. Les entrées sont donc des unités lexicales formellement composées. Des codes précisant les variations de formes ainsi que des indications de traits sémantiques accompagnent les entrées. Ce dictionnaire contient plus de 100 000 mots composés (90 000 noms, 15 000 constructions « être Prép N », 8000 adverbes, 500 conjonctions). (Silberztein M. 1990) a construit le dictionnaire électronique des mots composés DELAC.

---

5 Dictionnaire Electronique du LAld pour les mots Simples.
6 Dictionnaire Electronique du LAld pour les mots Composés.
2.2.3.1 Exemples d’entrées

Voici quelques exemples d’entrées du DELAC :

\[
cousin/germain,\text{un/N32/A32/ms} ;++ ;\text{Hum NA}
\]
\[
pomme/\text{de//terre},\text{une/N21/fs} ;-- ;\text{Conc NDN}
\]
\[
actualités/télévisées,\text{les/fp} ;-- \text{NA}
\]

Dans la première ligne, on a l’entrée cousin germain, le déterminant est un, le 1er composant a pour code morphologique N32, pour le deuxième c’est A32. C’est un mot masculin singulier (ms), humain (Hum), et c’est un NA (Nom Adjectif).

2.2.3.2 Les différentes classes de noms composés

Les noms composés constituant les entrées du DELAC appartiennent chacun à une classe syntaxique, trois catégories permettent un regroupement selon la longueur :

- longueur deux avec deux mots pleins,
- longueur trois avec trois mots pleins,
- longueur plus grande, avec quatre, cinq ou six mots pleins.

2.2.3.2.1 Noms composés binaires (longueur 2)

(Gross Gaston, 1986) a établi la typologie des noms composés de longueur deux, elle comprend sept classes : AN, NA, NN, NDN, NAN, PN, VN. D’autres classes ont été recensées (Courtois B. et al. 1997).

Les deux classes NA et NDN comptent le plus grand nombre de représentants. Au total 160 000 mots composés environ sont rassemblés et codés dans le DELAC (Silberztein M. 1990). Les classes syntaxiques sont recensées dans le tableau 2.1.

<table>
<thead>
<tr>
<th>Classe syntaxique</th>
<th>Signification</th>
<th>Exemple</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>Adjectif Nom</td>
<td>grand-père</td>
</tr>
<tr>
<td>NA</td>
<td>NA Nom Adjectif</td>
<td>carte orange</td>
</tr>
<tr>
<td>NAN</td>
<td>Nom à Nom</td>
<td>scie à métaux</td>
</tr>
<tr>
<td>NDN</td>
<td>Nom de Nom,</td>
<td>donnée de base</td>
</tr>
<tr>
<td>NPN</td>
<td>Nom Prep Nom</td>
<td>adressage par octets</td>
</tr>
<tr>
<td>NN</td>
<td>Nom Nom</td>
<td>carte mère</td>
</tr>
</tbody>
</table>
Tableau 2-1 : Liste des classes syntaxiques de longueur 2 avec exemples.

2.2.3.2.2 Noms composés ternaires

Les classes des composés ternaires recensées par (Gross M., Courtois B., 1998) sont décrites dans le tableau 2.2.

<table>
<thead>
<tr>
<th>Catégorie syntaxique</th>
<th>Composants</th>
<th>Exemple</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAA</td>
<td>Nom Adjectif Adjectif</td>
<td>Moyenne centrée réduite</td>
</tr>
<tr>
<td>AAN</td>
<td>Adjectif Adjectif Nom</td>
<td>ancien premier ministre</td>
</tr>
<tr>
<td>ANA</td>
<td>Adjectif Nom Adjectif</td>
<td>ancienne puissance coloniale</td>
</tr>
<tr>
<td>ANPN</td>
<td>Adjectif Nom prep Nom</td>
<td>abominable homme des neiges</td>
</tr>
<tr>
<td>NPAN</td>
<td>Nom Prep Adjectif Nom</td>
<td>accord de libre échange</td>
</tr>
<tr>
<td>NPNA</td>
<td>Nom Prep Nom Adjectif</td>
<td>abandon du domicile conjugal</td>
</tr>
<tr>
<td>NPNPN(^7)</td>
<td>Nom Prep No Prep Nom</td>
<td>abus de droit de réquisition</td>
</tr>
<tr>
<td>NXA</td>
<td>Nom ADV Adjectif</td>
<td>accotement non stabilisé</td>
</tr>
<tr>
<td>NXN</td>
<td>Nom X Nom</td>
<td>action anti-tabac</td>
</tr>
<tr>
<td>NAPN</td>
<td>Nom Adjectif Prep Nom</td>
<td>Local réservé au sommeil</td>
</tr>
</tbody>
</table>

Tableau 2-2 : Liste des classes syntaxiques de longueur 3 avec exemples.

2.2.3.2.3 Plus longs

Les termes de cette catégorie sont moins nombreux que ceux des classes précédentes, nous avons les termes :

- de longueur quatre (classe NX4) avec quatre mots pleins avec par exemple :
  accès à une base de donnée distante,

- de longueur cinq, avec 5 mots pleins comme :
  accusé de réception d’unité de commande pas prête,

\(^7\) La préposition P peut être vide.
et de longueur six, avec 6 mots pleins tels que :
dispositif extérieur d’arrêt de l’admission du combustible gazeux.
Cet exemple est extrait de (Domingues C. 2001).

### 2.2.4 Le dictionnaire des mots composés fléchis DELACF

Le DELACF est le dictionnaire électronique qui contient toutes les formes composées fléchies. Le fonctionnement de la flexion est dépendant de la classe du composé. Dans (Silberztein M. 1990), le pluriel des noms composés est abordé pour les principales classes. Nous avons le genre et le nombre des noms composés dans le tableau 2.3.

<table>
<thead>
<tr>
<th>Classe</th>
<th>Genre et nombre des noms composés</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA, AN</td>
<td>Genre et nombre identiques à ceux du nom et de l’adjectif</td>
</tr>
<tr>
<td>NDN</td>
<td>Genre et nombre identiques à ceux du 1er constituant N1</td>
</tr>
<tr>
<td>PN</td>
<td>Genre et nombre identiques à ceux du 1er constituant</td>
</tr>
<tr>
<td>NN</td>
<td>1er nom au pluriel, les deux au pluriel 2ème nom au pluriel, aucun</td>
</tr>
<tr>
<td>VN</td>
<td>Genre et nombre indépendants de celui du nom constituant La plupart sont masculins</td>
</tr>
<tr>
<td>NAN</td>
<td>Genre et nombre du 1er constituant</td>
</tr>
</tbody>
</table>

Tableau 2-3 : Genre et nombre des noms composés (Silberztein M. 1990).

Nous avons l’exemple de flexion du nom composé « aide sociale » :

- Un aide social, au masculin singulier ;
- Une aide sociale, au féminin singulier ;
- Des aides social(e)s, est ambigu, pour le féminin pluriel et le masculin pluriel.

### 2.2.5 Le Lexique-grammaire

#### 2.2.5.1 Introduction

C’est une approche formelle, transformationnelle et empirique de la linguistique qui met en avant le caractère fondamental du lexique. L’objectif est de recenser exhaustivement et systématiquement l’ensemble des comportements syntaxiques des phrases simples. Cette méthode, élaborée à partir des années 70 au LADL (Laboratoire d’Automatique Documentaire et Linguistique, CNRS) par Maurice Gross et son équipe, met l’accent sur la nécessité d’établir des inventaires descriptifs systématiques des faits linguistiques, à l’opposé de la démarche « chomskyenne » d’élaboration d’un modèle abrégé et universel du langage, où la syntaxe se pose comme entièrement autonome de la lexicologie.
2.2.5.2 Le lexique-grammaire des verbes du français

Le lexique-grammaire des verbes comprend quatre principaux types de constructions de phrases : complétives, transitives, intransitives et locatives. Les tables syntaxiques, dûes à (Gross M. 1975, 1968), (Boons J.P.1976), (Guillet A.et Leclère C. 1981), constituent une description complète des emplois des verbes français. Actuellement, la classification est répartie en 81 tables, contenant plus de 31 000 emplois, dont près de 20 000 phrases figées.

Les travaux de Maurice Gross (Gross M. 1975) concernant l'étude de la syntaxe du verbe répertorient les comportements syntaxiques (sujet, nombre et types de compléments admis,...) de 5.000 verbes simples du français, environ 15.000 emplois différents sont obtenus, et encodés dans de grandes matricies (les tables du lexique-grammaire), regroupées selon leur structure définitionnelle. Les lignes de ces matrices sont les emplois de verbes, et les colonnes les propriétés syntaxiques. Un signe '+' indique que l'unité lexicale accepte cette propriété, et '-' indique qu'elle ne l'accepte pas. Nous présentons un extrait de la table 32 CL dans la figure 2.1.

| N.V. | Acco || Brac || Empr || Croc || Coup || Coum || Crau || Embr || Empoi || Eplon || Epré || Étrei |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
| Max  | saccroch | une branche avec son hameçon |
Selon Maurice Gross, les verbes du français possèdent des comportements syntaxiques quasiment uniques, et il devient donc impossible de les prédire à partir de règles générales. Seule l’accumulation systématique, au sein d’un lexique, des constructions syntaxiques possibles pour un prédicat déterminé est à même de rendre compte de la réalité de la langue et permettre un traitement automatique précis de celle-ci.

2.3 Les dictionnaires de terminologie et bases de données terminologiques

Selon (El Hanachi M. 2007), de nombreuses ressources terminologiques électroniques multilingues sont mises à la disposition des usagers et en particulier des professionnels de l'information. Ces ressources, matérialisées sous forme de dispositifs techniques électroniques, servent d'outils de médiation entre les usagers et l'information à traiter. Ils deviennent de fait un dispositif d’appropriation des savoirs.

2.3.1 IATE

Au départ, en 1975, cette base de données s’appelait EURODICAUTOM, puis elle a migré vers IATE\(^8\) en 2007. IATE (“Inter-Active Terminology for Europe”) est la base de données interinstitutionnelle de l’UE (Union Européenne).

Nous trouvons dans la figure 2.2, les préférences de recherche avec IATE, pour le terme "mémoire ", dans le domaine "informatique et traitement des données".

Figure 2-2 : Préférences de recherche avec IATE pour le terme "mémoire".

IATE a été utilisée dans les agences et institutions de l’UE depuis 2004 pour la collecte, la diffusion et la gestion partagée de la terminologie propre à l’UE. Le projet a été lancé en 1999 avec l’objectif de fournir une infrastructure basée sur internet pour toute les ressources terminologiques de l’UE en améliorant la disponibilité et la normalisation de l’information. Elle contient actuellement environ 1,4 million d’entrées multilinguales.

La figure 2.3 nous permet de voir une partie des résultats de la recherche pour le terme « mémoire » avec une traduction en anglais:
2.3.2 Le dictionnaire SensAgent

On notera qu’il peut être édité par ALEXANDRIA (voir §3.3.14). Il s’agit d'un dictionnaire multilingue en ligne. Il donne accès à 22 dictionnaires unilingues et permet de rechercher des termes dans n’importe quelle combinaison de langue parmi ces dictionnaires.

Dans un dictionnaire unilingue, on trouve la définition complète des termes, de nombreux synonymes, ainsi que de nombreux contextes. Les recherches bilingues donnent plusieurs traductions possibles et équivalents dans la langue cible. Les deux versions (uni- et bilingue) comprennent également une "arborescence analogique" approfondie pour le terme, ainsi que des anagrammes, des conjugaisons, etc. Dans la figure 2.4, on trouve, sur le site "memodata.com", la page d’accueil pour accéder au dictionnaire en ligne "SenSagent".

---

9 http://dictionary.sensagent.com/

10 http://www.sensagent.com/fr/
Les résultats de la recherche effectuée pour le terme "mémoire" sont donnés avec un extrait concernant les locutions trouvées dans divers domaines, dans la figure 2.5.

![Figure 2-5 : Résultats de la recherche du mot "mémoire".](image)
2.3.3  Le glossaire OSINET\textsuperscript{11}

Il contient 5748 entrées qui sont des termes, des acceptations particulières. La recherche d’un terme selon son contexte (OS, périphériques, presse du monde des TIC, XML…).

En consultant ce dictionnaire, nous avons obtenu par exemple, la définition suivante du terme "mémoire cache" :

« Espace disque alloué au navigateur (browser), et lui permettant de remplir, sur l'ordinateur de l'utilisateur, et sauf paramétrage contraire par ce dernier, le même office qu'un serveur proxy. Les effets sont identiques à ceux des proxies : absence de comptabilisation de l'audience par le site principal, sauf procédure particulière ». 

2.3.4  Le dictionnaire électronique LVF


Pour J. Dubois et F. Dubois-Charlier, « La classification des verbes français repose sur l'hypothèse qu'il y a adéquation entre les schèmes syntaxiques d'une langue et l'interprétation sémantique qu'en font les locuteurs de cette langue : à la différence syntaxique entre venir à Paris et venir de Paris correspond une différence entre la destination et l'origine. » Les auteurs de LEXVALF (Lexique des valences verbales du français) ont produit pour chaque entrée verbale une masse de données importante et fortement structurée. A partir d'une liste préalablement établie de tous les patrons de compléments acceptés par l'ensemble des verbes, sont établies, pour chaque verbe, toutes les formules de compléments acceptées. Leur description est enrichie par l'intégration de tous les phénomènes de sous-catégorisation grammaticale et de restrictions grammaticales pertinentes, le tout assorti d'au moins un exemple extrait d'un des corpus des différentes ressources.

LEXVALF se présente partiellement comme un équivalent formalisé d'un dictionnaire usuel, à cela près que l'essentiel de l'effort est consacré à la description de particularités syntaxiques de constructions des unités lexicales verbales : caractérisation syntaxique des arguments et des propriétés de construction particulière (pronominale, impersonnelle, etc.). LEXVALF ne développe pas une caractérisation sémantique de chaque emploi verbal identifié par une formule de complément ; il se borne à associer à chaque réalisation de patron de complément décrite l'emploi sémantique correspondant dans Les Verbes français (LVF) de (Dubois J. et Dubois-Charlier F. 1997). Dans LEXVALF les relations de sélection lexicales entre le verbe et chacun des arguments sont mentionnées - voire entre les arguments - au

\textsuperscript{11} http://www.riff.org/glossaire
moyen de notations de sémantique référentielle très générales (humain, non humain, abstrait, concret) ou plus rarement de "classes d'objets", comme les "noms partie du corps".

2.3.5 **Dictionnaire de l'informatique et de l'internet Dicofr.com**

Dans ce dictionnaire, on a en entrée le terme. Il contient les termes officiels de l'informatique. Pour chaque terme, une définition est introduite ainsi qu'une traduction en anglais. La liste des termes est triée par ordre alphabétique. Le dictionnaire contient 4383 définitions. Nous avons dans la figure 2.6 les résultats de la recherche pour le terme « mémoire ».

![Figure 2-6 : Recherche pour le terme "mémoire".](http://www.Dicofr.com)

Nous pourrons aussi rechercher une définition de terme précis, ici en recherchant le terme "mémoire tampon", nous avons obtenu la définition suivante :

« La mémoire tampon est une mémoire volatile qui a pour but de permettre un flux continu de données lors d'une transmission ».

---

2.3.6 Le répertoire terminologique 2000

Le site RÉTIF\(^\text{13}\) a pour ambition de permettre la diffusion et la promotion d'un vocabulaire français pour l'informatique (au sens large, comprenant une partie des réseaux et Internet).

Le glossaire informatique des termes a été publié au Journal Officiel français par la Commission générale de terminologie et de néologie le 22 septembre 2000. Le site RÉTIF (Philippe Deschamp) a pour ambition de permettre la diffusion et la promotion d'un vocabulaire français pour l'informatique (au sens large, comprenant une partie des réseaux et Internet).

Des abréviations sont utilisées pour certaines entrées et sont données dans le tableau 2.4 :

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{adj.} & \text{adjectif} & \text{loc. adj.} & \text{localisation adjective} & \text{n.f.} & \text{nom féminin} \\
\hline
\text{adv.} & \text{adverbe} & \text{loc. adv.} & \text{localisation adverbiale} & \text{n.m.} & \text{nom masculin} \\
\hline
\text{pl.} & \text{pluriel} & \text{loc.} & \text{locution} & \text{n.} & \text{nom épicène ou noms} \\
\hline
\text{v.intr.} & \text{verbe} & \text{v.} & \text{verbe intransitif} & \text{v.tr.} & \text{verbe transitif} \\
\hline
\end{array}
\]

\text{Tableau 2-4 : Liste des abréviations}\(^\text{2}\).

Pour chaque entrée nous avons les rubriques suivantes :

- le terme (entrée),
- le domaine,
- une définition,
- les synonymes s’il y a lieu,
- la traduction en anglais.

Ceci donne sur un exemple :

accès direct,

domaine : Informatique,

definition : Mode d'écriture ou de lecture de données se faisant au moyen d'adresses qui repèrent l'emplacement de ces données,

anglais : direct access, random access.

\(^{13}\) http://www-rocq.inria.fr/qui/Philippe.Deschamp/RETIF/
2.3.7 UNBIS

UNBIS\(^{14}\) est un thesaurus multidisciplinaire multilingue, utilisé par les différents organismes des Nations Unies\(^{15}\) pour indexer leurs documents. Ce thesaurus propose les équivalences dans les six langues officielles des Nations-Unies (anglais, arabe, chinois, espagnol, français et russe). L’utilisateur peut choisir\(^{16}\) la langue source soit par une recherche soit par exploration des listes thématique (micro-thesaurus) ou alphabétique mais il ne peut pas choisir de visualiser une langue cible parmi les six officielles. Les informations sur les équivalences dans les six langues sont affichées en même temps sur un même écran.

Une première version en ligne, temporaire, fut mise en ligne en novembre 2001. Dans la figure 2.7, nous trouvons une recherche avec UNBIS, pour le terme "mémoire", la recherche est simple et est effectuée pour trouver tous les mots.

![Figure 2-7 : Recherche avec UNBIS.](http://unhq-appspub-01.un.org/LIB/DHLUNBISThesaurus.nsf/$$searchf?OpenForm)

UNBIS contient la terminologie utile pour l'analyse thématique des documents et autres publications relatifs aux programmes et activités des Nations Unies. Il est utilisé comme fichier d'autorité thématique du Système d'information bibliographique de l'ONU (UNBIS) et a été incorporé en tant que liste de sujets dans le Système de diffusion électronique des


\(^{15}\) [www.Unesco.org](http://www.Unesco.org)


2.3.8 Le GDT (Grand dictionnaire de la terminologie)

Sur le site de l’Office québécois de la langue française on trouve le GDT grand dictionnaire de la terminologie dans divers domaines. Dans la recherche effectuée pour le terme « mémoire », plusieurs fiches sont affichées, nous en donnons un extrait dans la figure 2.8.

Figure 2-8 : Résultats de la recherche du terme "mémoire" dans le GDT.17

2.3.9 WORDNET 18

Le projet Wordnet (réseau de mots) est un projet américain, de l’université de Princeton. Wordnet est une vaste base de données lexicale développée sous la direction de George A. Miller.

Les noms, les verbes, adjectifs sont regroupés en ensembles de synonymes cognitifs (‘synsets’) exprimant chacun un concept distinct. Les ‘synsets’ sont reliés entre eux par des relations sémantico-conceptuelles et lexicales, Wordnet en fait un outil utile pour la linguistique computationnelle et le traitement du langage naturel. Dans la figure 2.9, nous trouvons une recherche avec WordNet pour le terme "memory" qui est un nom, "noun", pour chaque sens du mot, une explication est associée.

17 http://www.granddictionnaire.com/BTML/FRA/r_Motclef/index800_1.asp
18 http://wordnet.princeton.edu/
Nous constatons que les termes composés associés sont aussi affichés (« computer memory », « computer storage », « memory board »).

Figure 2-9 : Recherche WordNet pour le terme ‘mémoire’.

2.3.10 Genoma

Genoma\(^{19}\) est présentée comme une « banque termino-ontologique » sur le génome humain. C’est une ressource de type termino-ontologique qui propose une définition pour chaque terme, la catégorie morphologique et le genre d’un mot, les informations flexionnelles et dérivationnelles avec des étiquettes. L’interrogation de la base de données peut se faire dans trois langues : anglais, espagnol et catalan.

Elle contient des informations conceptuelles et terminologiques, des textes et des documents sur le génome humain. Dans la figure 2.10, nous présentons la page d’accueil de Genoma.

\(^{19}\) http://genoma.iula.upf.edu:8080/genoma/index.jsp
2.3.11 Le Dictionnaire des développeurs\textsuperscript{20}

Il est organisé en thésaurus des termes de l’informatique, les termes ayant des relations sémantiques sont liés. Le tri est fait par thèmes, alphabétique ou arborescent. On a une ontologie du domaine.

« dico.developpez.com » propose un thésaurus des termes de l'informatique, à utiliser comme un dictionnaire très organisé puisque les termes ayant des relations sémantiques sont liés. Dans la figure 2.11, nous avons la page d’accueil du dictionnaire des développeurs, elle permet de naviguer par tri alphabétique ou par thème (Business Intelligence, Conception, culture, etc).

\textsuperscript{20} http://dico.developpez.com
2.3.12 UMLS (Unified Medical Language)

2.3.12.1 Introduction

Le programme de recherches et de développement a été lancé par la Bibliothèque Nationale de la Médecine des États-Unis (NLM pour l'acronyme américain) pour établir des sources de connaissances afin de faciliter le développement des systèmes qui aident les professionnels de santé pour rechercher et intégrer une information biomédicale. Les sources de connaissance peuvent être employées pour lier les systèmes d'information hétérogènes pour surmonter des problèmes de récupération provoqués par des différences dans les terminologies employées et la dispersion d'information appropriée à travers beaucoup de bases de données.

Le but de l’UMLS (Unified Medical System Language), système unifié de langage médical de la NLM (National Library of Medicine) est de faciliter le développement de systèmes informatiques qui se comportent comme s’ils « comprenaient » le sens de la langue de la biomédecine et la santé. À cette fin, NLM produit et distribue les sources de connaissances UMLS (bases de données) et des outils logiciels associés (programmes) pour une utilisation par les développeurs de systèmes, dans la construction ou l'amélioration des systèmes d'information électroniques. Ces systèmes créent, traitent, récupèrent, intègrent et / ou agrègent les informations et données biomédicales et de la santé, aussi bien qu’en recherche informatique. De par leur conception, les sources de connaissances UMLS sont multi-usages.

Elles peuvent être appliquées dans les systèmes qui exécutent une série de fonctions impliquant un ou plusieurs types d'informations, par exemple, les dossiers des patients, la
littérature scientifique, les données de santé publique. Les outils associés UMLS aident les développeurs de logiciels dans la personnalisation ou l'utilisation des sources de connaissances UMLS à des fins particulières. Les outils lexicaux travaillent plus efficacement en combinaison avec les sources de connaissances UMLS, mais peuvent également être utilisés indépendamment.

NLM et de nombreuses autres institutions appliquent les ressources UMLS dans une grande variété d’applications y compris la recherche d'information, le traitement du langage naturel, la création du patient et des données de recherche et le développement de l'entreprise à l'échelle des services de vocabulaire.

2.3.12.2 Terminologie utilisée dans CISMeF

La terminologie utilisée dans CISMeF (Catalogue et Index des Sites Médicaux de langue Française) est basée sur le thesaurus MeSH, ‘Medical Subject Headings’ (version 2010), réalisé par la National Library of Medicine NLM. Il s'agit de l'outil de référence pour l'indexation et la recherche d'informations médicales. Les mots clés et les qualificatifs en français sont ceux du MeSH bilingue réalisé par le département de l'Information scientifique et de la communication de l'INSERM. A cela s’ajoutent les types de ressources et les métatermes.

Les mots clés (ou descripteurs) servent à définir les thèmes traités. Les définitions proposées proviennent du MeSH, du Vidal de la famille, de la base de données Orphanet, des glossaires de la FNCLCC ou de la Ligue suisse contre le cancer. Les traductions françaises des définitions MeSH sont réalisées par l’équipe CISMeF. Les qualificatifs sont des concepts généraux, qui peuvent être affiliés à un mot clé pour en préciser le sens.

Les types de ressources sont utilisés pour décrire la nature des informations d'un site ou d'un document. Il s’agit d’une généralisation des types de publication du MeSH, adaptée aux ressources présentes sur internet.

Les actions pharmacologiques (ap) sont des concepts qui permettent de réunir les substances possédant cette action. Ces substances peuvent correspondre à des mots clés ou à des "concepts chimiques supplémentaires" (supplementary concept records). Ce concept permet donc d'effectuer une recherche sur tous les termes MeSH ayant une même action pharmacologique.

Les métatermes sont utilisés pour regrouper les mots clés, qualificatifs et types de ressources qui correspondent à une spécialité biologique ou médicale. Ils permettent de retrouver l’ensemble des documents d'une spécialité, la cancérologie par exemple.

Les "concepts chimiques supplémentaires" (sc) sont des termes MeSH qui n'ont pas le statut de mot-clé (descripteur) et qui correspondent à des noms de substances chimiques. Contrairement aux mots-clés, ils ne sont pas organisés en arborescence (pas de relation hiérarchique). Par contre, grâce au « mapping » proposé par le MeSH, ils peuvent être reliés à des mots-clés. Enfin, les "concepts chimiques supplémentaires" peuvent être aussi reliés à des actions pharmacologiques. A ce jour, il existe environ 186 700 concepts chimiques supplémentaires.
2.3.12.3 Sources de connaissances UMLS

Il existe trois sources de connaissances UMLS :

- Le Metathesaurus ;
- Le Réseau sémantique ;
- Le Lexique spécialiste.

2.3.12.3.1 Le Metathesaurus

Le métathesaurus est une très grande base de données, multi-usages et de vocabulaire multi-lingual qui contient des informations sur les concepts biomédicaux et de santé, leurs noms différents, et les relations entre eux. Il est construit à partir des versions électroniques de plusieurs thésaurus différents, des classifications, des jeux de codes et des listes de termes contrôlés utilisés dans les soins aux patients, des services de santé de facturation, statistiques de santé publique, d'indexation et de catalogage de la littérature biomédicale, et / ou fondamentale, clinique, et de la santé services de recherche de la santé. Dans cette documentation ils sont référencés comme les «vocabulaires sources» du métathesaurus. Dans le métathesaurus, tous les vocabulaires sources sont disponibles dans un seul format de base de données entièrement spécifié.

Le métathesaurus est organisé par concept ou sens. Son but est de relier les noms alternatifs et vues du même concept ensemble et d'identifier les relations utiles entre différents concepts. Tous les concepts dans le metathesaurus sont affectés à au moins un type sémantique du réseau sémantique. Cela fournit une catégorisation consistante de tous les concepts dans le metathesaurus à un niveau relativement général, représenté dans le réseau sémantique. Beaucoup de mots et des expressions multi-mots qui apparaissent dans les noms de concept ou de chaînes dans le métathesaurus apparaissent également dans le lexique SPECIALIST. Les outils lexicaux sont utilisés pour générer le mot, le mot normalisé, et des index de chaîne normalisée au métathesaurus. MetamorphoSys doit être utilisé pour installer toutes les sources de connaissances et UMLS est l'outil logiciel recommandé pour la personnalisation du métathesaurus.

2.3.12.3.2 Le lexique SPECIALIST

Le lexique SPECIALIST a été développé pour fournir les informations lexicales pour le système de traitement automatique du langage naturel SPECIALIST. C’est un lexique général de l’anglais qui contient beaucoup de termes médicaux. L’entrée du lexique contient les informations syntaxiques, morphologiques, orthographiques, nécessitées par le système SPECIALIST NLP.

Les outils lexicaux ‘The Lexical Tools’ sont utilisés pour adresser le haut niveau de variabilité des termes et mots du langage naturel. Les mots ont souvent plusieurs formes fléchies qui doivent être considérées comme des instances du même mot. Une entrée lexicale génère une variété de formes (chaînes) comprenant toutes les formes flexionnelles de chaque variante orthographique, comprenant toutes les formes flexionnelles de chaque variante orthographiée. La ponctuation et les espaces sont considérés significatifs. La catégorie
La syntaxe de l’entrée lexicale peut être soit un nom, un adjectif, un adverbe, un pronom, un verbe, une préposition, un déterminant, un auxiliaire ou une conjonction.

Les entrées du lexique peuvent être des termes multi-mots constitués d’autres mots, si le terme multi-mot est déterminé comme étant une entrée lexicale, par sa présence comme un terme dans des dictionnaires médicaux ou généraux ou dans un thesaurus médical tel que MeSH. Les expansions d’acronymes et d’abréviations utilisées généralement sont aussi considérées comme des termes multi-mots.

2.3.12.3.3 Le réseau sémantique

Le but du réseau sémantique est de fournir une catégorisation consistante de tous les concepts représentés dans le Metathesaurus et de fournir un ensemble de relations utiles entre ces concepts. Toutes les informations spécifiques sont trouvées dans le Metathesaurus.

2.3.13 EUROVOC

Eurovoc est un thésaurus multilingue couvrant tous les domaines de l'activité de l'Union européenne. Désormais consultable et téléchargeable gratuitement, Eurovoc est disponible en 22 langues officielles de l'Union européenne (bulgare, espagnol, tchèque, danois, allemand, estonian, grec, anglais, français, italien, letton, lituanien, hongrois, maltais, néerlandais, polonais, portugais, roumain, slovaque, slovène, finnois et suédois). Eurovoc permet d'indexer les documents dans les systèmes documentaires des institutions européennes et de leurs utilisateurs.

Toutes les versions linguistiques ont le même statut: à chaque terme préférentiel (descripteur) dans une langue correspond obligatoirement un terme préférentiel dans chacune des autres langues. Chaque concept du thésaurus comporte par défaut tous les équivalents linguistiques du terme préférentiel.

Il n’existe cependant pas nécessairement d'équivalence entre les termes non préférentiels (non descripteurs) dans les différentes langues. Chaque langue se distingue par sa richesse et ses différences sémantiques et culturelles et un concept dans une langue n’est pas toujours représenté dans une autre langue. La page de recherche EUROVOC est donnée dans la figure 2.12.

---

La figure 2.13 indique les résultats de la recherche pour le terme "mémoire", deux concepts sont trouvés dans deux domaines différents :

- Technologie et réglementation technique ;
- Electronique et électrotechnique.
2.3.14 ALEXANDRIA

Dominique Dutoit est le concepteur d'Alexandria, il a cofondé avec Patrick de Torcy la société Memodata en 1989, société qui propose aujourd'hui le logiciel Alexandria. Alexandria est à la croisée des chemins entre un dictionnaire multilingue et une base sémantique multilingue, à la WordNet. En effet, elle a été réalisée en s'appuyant sur la structure de WordNet pour la partie anglaise et sur le Dictionnaire Intégral de la société Mémodata pour la partie française.

C’est une ressource de type lexicographique propose une définition pour chaque terme, la catégorie morphologique et le genre d’un mot, les informations flexionnelles et dérivationnelles avec des étiquettes. Alexandria est davantage une base sémantique lexicale (réseau sémantique) structurant les concepts du domaine général dans une langue et établissant les équivalences dans une deuxième langue Alexandria, qualifiée de dictionnaire électronique, recouvrant dans son contenu à la fois un dictionnaire de définitions, un dictionnaire de synonymes, un dictionnaire de traductions vers 22 langues, des entrées analogique et onomasiologique («dictionnaire allant des idées vers les mots»), des dérivés sémantiques, la navigation dans un réseau lexical à la WordNet et des éléments d’ontologie.


Alexandria affiche les différents sens connus du terme ainsi qu’une liste de pluritermes le contenant (relation d’inclusion lexicale). Alexandria ne propose que des exemples de phrase pour illustrer tel ou tel sens d’un terme, tel un dictionnaire de langue.

22 www.SensAgent.com
Alexandria n'édite pas uniquement les contenus dictionnaire de Memodata (dictionnaire SensAgent). C'est un média orienté vers les dictionnaires. Memodata travaille depuis 15 ans plutôt sur un réseau conceptuel. 950.000 nouvelles fiches ont été ajoutées pour le Français, fin 2005. La figure 2.15 contient la page d’accueil pour l’utilisation des dictionnaires en ligne.

![Figure 2-15 : Dictionnaires en ligne.](image)

Les extraits des résultats affichés pour cette recherche sont les suivants :
Le dictionnaire fondamental de l’informatique et de l’internet : DicInfo

Le DiCoInfo (Dictionnaire fondamental de l’informatique et de l’Internet) est une base de données lexicales contenant des termes fondamentaux appartenant aux domaines de l’informatique et de l’Internet. Il s’appuie sur les principes théoriques et méthodologiques de la Lexicologie explicative et combinatoire (LEC).

Les unités lexicales retenues dans le dictionnaire, sont des noms, verbes, adjectifs et adverbes.

Exemples :

- **Noms** : ordinateur, mémoire, virus ;
- **Verbes** : configurer, formater, activer ;
- **Adjectifs** : compilé, programmable, binaire ;
- **Adverbes** : numériquement, dynamiquement, virtuellement.

Les unités retenues renvoient, par exemple à :

- **des objets concrets** : microprocesseur, mémoire, carte, bus ;
- **des représentations** : caractère, instruction, données ;
- **des activités** : configurer, naviguer, traitement ;
- **des propriétés** : compatibilité, asynchrone, lisibilité ;
- **des animés** : internaute, programmeur, webmestre ;
- **des unités de mesure** : mégahertz.

Les entrées au nombre de 1500 sont pour la plupart des mots simples. Les entrées du dictionnaire définissent un réseau lexical du domaine. La recherche se fait dans des fiches. Les termes reliés sont organisés en respectant l’ordre suivant (L’homme M.C. 2005) :
- Les quasi-synonymes, les sens voisins et les génériques, par exemple discussion relié à dialogue et blogage ;

- Les antonymes, contraires et contrastifs, par exemple installer relié à désinstaller ;

- Les liens donnant souvent lieu à des termes reliés morphologiquement par exemple programmer est relié à programmation et programme ;

- Les « sorte de » sous forme de spécifiques ou de combinaisons composées du terme et d’une modification par exemple fichier relié à ~ source, ~ viral, ~ XML, ~ pièce jointe ;

- Les collocatifs verbaux et dérivés nominaux et adjectivaux de ces verbes par exemple fichier est relié à créer un ~, modifier un ~, génération d’un ~.

Le DiCoInfo est un dictionnaire en construction et l’état d’avancement de la rédaction varie d’un article à l’autre.

**2.3.16 TermSciences**

C’est un portail terminologique offrant la consultation de terminologies scientifiques multilingues (lexiques, dictionnaires, thésaurus, etc.).

TermSciences est présenté comme un « portail terminologique multidisciplinaire » contenant les termes contrôlés issus de nombreuses autres ressources dont les deux lexiques (PASCAL, FRANCIS) de l’INIST, les termes issus du thésaurus MESH (Medical Subject Heading), un dictionnaire de l’INRA (L’Institut national de la recherche agronomique) sur les biotechnologies et le thesaurus de la banque de données en santé publique (BDSP). C’est une ressource de type termino-ontologique.

Pour le terme « mémoire », 167 fiches sont trouvées dans divers domaines, le concept accès mémoire est traduit en anglais, en français et en espagnol. Dans la figure 2.17, on trouve les résultats de cette recherche pour toutes les sources.

---

23 www.TermSciences.fr

24 Institut de l’information scientifique et technique (INIST).
2.3.17 Le dictionnaire du NEF

Il a été créé en 2007, l’auteur est Marie Lebert. Le dictionnaire contient un ensemble de définitions détaillées. Le classement est fait par ordre alphabétique. Les 700 entrées sont des termes, sigles et nom propres en rapport avec l’internet. En faisant une recherche pour le terme « mémoire », nous avons obtenu plusieurs entrées, nous donnons ci-dessous trois exemples :

- « Mémoire (ordinateur) : la mémoire d'un ordinateur comprend en général une mémoire vive ou mémoire RAM (random access memory), qui permet de lire et écrire des données mais ne conserve pas leur contenu, et une mémoire morte ou mémoire ROM (read only memory) qui permet la conservation des données lorsque l'appareil est hors tension. La mémoire vive se mesure en mégaoctets (Mo) ou gigaoctets (Go). [Voir aussi: octet, ordinateur] » ;


- « Mémoire SDRAM (synchronous dynamic random access memory) : la mémoire SDRAM (synchronous dynamic random access memory) est une mémoire vive

http://www.etudes-francaises.net/dico/final.html
dynamique synchrone destinée aux cartes graphiques et aux cartes mères. Elle va à la vitesse du processeur, ce qui permet de réduire ou d’éliminer le temps d’attente. [Voir aussi: mémoire, processeur.]

2.3.18 Le DicoduWeb

Nous avons la page d’accueil dans la figure 2.18, la page d’accueil du dictionnaire.

![Figure 2-18 : Page de recherche du DicoduWeb pour la liste « M ».](http://www.olecorre.com)

Le classement est fait par ordre alphabétique. Une définition explique simplement chaque terme. On y trouve une partie pour la microinformatique avec 7000 termes environ. Il a maintenant 12 ans d’existence. Dans la figure 3.18, nous avons dans la page de recherche du DicoduWeb, nous avons la liste « M » qui contient 388 définitions.

2.4 Conclusion

La plupart des ressources sondées, notamment TermSciences, Alexandria et Genoma, offrent des fonctionnalités hybrides mêlant celles d’un dictionnaire, d’un thésaurus, d’une ontologie et des outils d’exploration automatique de corpus, si bien qu’il faut aujourd’hui inventer de nouveaux qualificatifs pour les décrire.

D'ailleurs, les dénominations de ces ressources le montrent : « portail terminologique » pour TermesSciences, « base de connaissances termino-ontoligique» pour Genoma et « outils d’aide contextuelle, dictionnaire électronique multilingue » ou encore « agent intelligent » pour Alexandria. UNBIS et Eurovoc restent très clairement des thésaurus, Alexandria est ni un dictionnaire électronique, ni une ontologie, ni une base sémantique lexicale, elle est les trois à la fois.

Les bases de connaissances telles que WordNet sont trop générales pour des applications dans des domaines spécialisés. Ces ontologies bien que très complètes ne conviennent pas pour le traitement des langues techniques.
Chapitre 3 :
Le système NooJ
3 Le système NooJ

3.1 Introduction

Nous avons présenté dans le chapitre 2, le système DELA, issu des travaux effectués au LADL. L’ensemble de dictionnaires, de lexiques grammaires et le système Intex (Silberztein M. 1993) développés permettent le traitement automatique de textes en langue naturelle. Cependant, l’approche INTEX étant limitée, son auteur (Silberztein M.) a développé le système NooJ.

NooJ, tout comme son prédécesseur INTEX, est un environnement de développement linguistique utilisé comme outil de formalisation des langues naturelles et de développement d’applications de traitement automatique des langues (TAL). Actuellement, il existe des modules linguistiques NooJ pour une douzaine de langues. Une demi-douzaine d’applications informatiques du TAL ont été construites avec NooJ. Dans sa thèse (Mesfar S. 2008) intègre les modules NooJ pour la langue arabe.

3.2 D’INTEX à NooJ

(Silberztein M. 2005) met en avant les limitations de l’approche INTEX qui sont principalement les suivantes :

- INTEX utilise 5 dictionnaires avec pour chacun un formalisme particulier ;
- Les méta-graphes ne peuvent pas être utilisés avec d’autres grammaires syntaxiques ;
- La séparation DELAS-F et DELAC-F n’est pas sans conséquences : un grand nombre de termes ont leurs variantes orthographiques décrites dans le DELAS et d’autres dans le DELAC ;

De plus INTEX est incapable de calculer les flexions des composés automatiquement.

Les dictionnaires de NooJ (Silberztein, 2003) sont basés sur les dictionnaires DELA-type d’INTEX ainsi que les lexiques-grammaires. Les dictionnaires de NooJ semblables aux dictionnaires DELAS-DELAC peuvent représenter les variantes terminologiques ou orthographiques des entrées lexicales. Les dictionnaires NooJ sont une version unifiée des dictionnaires DELAS, DELAC, DELAV.

Dans NooJ (Silberztein, 2003), tous les dictionnaires sont représentés dans un format unique ; Les dictionnaires par opposition à ceux d’INTEX, contiennent la description complète des flexions et des dérivations pour leurs entrées. C’est pour cela que NooJ ne nécessite pas de dictionnaires DELAF et DELACF.

3.3 NooJ : une plateforme de développement linguistique

NooJ est un environnement linguistique de développement, issu de douze années d’expérience de l’auteur (Silberztein, 1993) comme utilisateur et concepteur du système INTEX au LADL.


Le nouveau moteur linguistique du logiciel NooJ a été réécrit à partir de son prédécesseur INTEX28 pour apporter plus de facilité de manipulation et répondre aux besoins de la communauté qui s’est constituée autour de lui.

NooJ est utilisé pour la construction à grande couverture de descriptions formalisées des langues naturelles dans le but de les appliquer en temps réel à de gros corpus. Les descriptions des langues naturelles sont formalisées dans des dictionnaires électroniques par des grammaires représentées par des ensembles de graphes. NooJ fournit des outils pour décrire la morphologie flexionnelle et dérivationnelle, les variations terminologiques et orthographiques, le vocabulaire (mots simples, mots composés, expressions figées), la syntaxe et la sémantique. Les principales caractéristiques de NooJ sont présentées dans les sections qui suivent.

3.3.1 Architecture intégrée

Dans NooJ, toutes les connaissances linguistiques sont séparées de l’algorithme d’analyse lui-même. L’architecture globale de NooJ est basée sur un ensemble de modules linguistiques : orthographique, flexionnel, morphologique, dérivationnel et syntactico-sémantique. Cette structure est garnie par un ensemble d’outils tels que :

- un éditeur de graphes,
- un concordancier,
- un débogueur de grammaires,
- des outils statistiques,
- etc.

Cet ensemble en interaction permet de répondre aux besoins les plus pointus des utilisateurs de NooJ et notamment des spécialistes du traitement des langues naturelles.

3.3.2 Architecture orientée objet

NooJ est développé sous une architecture orientée objets (Silberztein M. 2004). Les composants du système (objets) intègrent les données ainsi que les routines nécessaires pour leurs traitements. La communication et la coordination entre les objets sont réalisées par un mécanisme interne. Cette architecture repose souvent sur les trois piliers : encapsulation, héritage et polymorphisme. Elle présente les avantages suivants :

- Eviter la redondance dans le code source grâce au concept d’héritage. Ceci rend le code source plus lisible et gérable ;

- Accéder directement à toutes les méthodes publiques dans NooJ, au lieu d’avoir à sélectionner un ensemble de sous-programmes. Ceci apporte une flexibilité et une utilisabilité supplémentaires au système ;

- Pouvoir ajouter d’autres fonctionnalités supplémentaires ou des méthodes spécifiques à des langues sans avoir à modifier l’architecture globale de l’application.

3.3.3 Utilisation de la technologie à états finis

NooJ contient des outils permettant d’éditer, de tester, de déboguer et de gérer des ensembles importants de graphes à états finis. Tous les objets traités par NooJ sont ou peuvent être transformés sous forme de transducteurs à nombre fini d'états. Toutes les opérations sur les textes, grammaires et dictionnaires se ramènent ainsi à des opérations sur des transducteurs. NooJ utilise aussi des outils plus puissants comme les graphes récursifs qui sont équivalents à des grammaires hors-contexte. Lorsque ses transducteurs récursifs à variables sont utilisés en cascade, ils donnent à NooJ la puissance d’une machine de Turing (Silberztein M. 1999).

Toutefois, l’application impose toujours au linguiste une approche ascendante, et en largeur : chaque phénomène doit être traité localement par un ou plusieurs graphes le plus souvent à états finis, ces graphes sont ensuite réutilisables pour traiter d’autres phénomènes plus généraux. Cette approche en cascade met en lumière la complexité de la variation lexicale et la faiblesse de toute généralisation.

NooJ ne traite pas que des graphes d’états finis, avec NooJ on peut aussi créer :

- Des grammaires algébriques ;
- Des graphes récursifs (RTN).
3.3.4 Développement de ressources linguistiques à large couverture

Les différentes ressources linguistiques sont décrites dans des structures de données autonomes (Silberztein, 2005). En l’occurrence, elles sont représentées soit dans des formats textuels lisibles et accessibles par un non-informaticien (comme pour les dictionnaires et les fichiers des paradigmes flexionnels et dérivationnels), soit sous forme de graphes facilement accessibles et compréhensibles qui peuvent aisément être contrôlés et modifiés sans avoir à maîtriser l’ensemble du programme (comme pour les grammaires flexionnelles, morphologiques et syntaxiques).

3.3.5 Moteur linguistique robuste

Etant donné le nombre étendu de langues prises en charge dans NooJ, son moteur linguistique est assez robuste. En effet, analyser des textes écrits en français, allemand, thaïlandais, chinois et arabe exige divers algorithmes d’analyse pour traiter :

- L’absence d’accentuation pour les lettres majuscules en français ;
- L’absence de vocalisations dans les textes arabes standards ;
- La taille importante de l’alphabet chinois ;
- Le nombre potentiellement illimité des mot-composés dans les mots en allemand ;
- Etc.

De façon similaire, presque chaque langue a un système de tri différent ; par exemple, en espagnol, les lettres doubles telle que ‘ll’ sont considérées comme une seule lettre quand il s’agit d’effectuer un tri dans l’ordre lexicographique. Malgré toutes ses spécificités et divergences, NooJ ne propose à ses utilisateurs qu’une seule et unique application quelque soit le traitement envisagé.

3.3.6 Traitement de corpus

NooJ peut traiter directement des ensembles potentiellement importants de documents. Ces documents peuvent être codés dans plus d’une centaine de formats :

- tous les formats des types ASCII, EBCDIC, Unicode (UTF-8, UTF-16, etc.),
- les formats standard tels que HTML (format par défaut des textes disponibles en ligne), XML (format de bases de données textuelles), PDF (format de documents portables), RTF (format de textes enrichis), Microsoft WORD, etc.

3.3.7 Construction, édition et gestion de concordances sophistiquées

NooJ permet de construire et de gérer des concordances sophistiquées. Il permet de combiner plusieurs requêtes, filtrer manuellement les résultats incorrects et de sauvegarder la
concordance résultante. Les outils proposés pour la gestion des concordances permettent d’aborder certains problèmes linguistiques de façon incrémentale.

Par exemple, il est envisageable de commencer son travail avec NooJ par l’identification de certains termes ou expressions dans un corpus, tout d’abord de façon naïve (par exemple, rechercher tous les mots finissant par –ment pour identifier des adverbes, ou rechercher tous les noms pour identifier des dates). Puis, au vu de la concordance résultante, il est possible de raffiner progressivement la caractérisation des phénomènes, d’une part en améliorant les requêtes (par ex. en tenant compte du contexte des mots), d’autre part en classifiant les résultats obtenus (par ex. pour distinguer les adverbes en –ment, par exemple "régulièrement", des noms en "–ment", par ex. "investissement"). Les résultats des requêtes peuvent être facilement contrôlés.

3.3.8 Annotation interactive de corpus

NooJ permet d’appliquer des grammaires représentées sous forme de graphes ou d’expressions rationnelles augmentées à un corpus. Le résultat est affiché dans une table de concordances où la colonne du milieu contient le mot ou la séquence de mots reconnus. L’utilisateur peut valider ou éliminer chaque entrée de cette concordance. La concordance ainsi filtrée peut, ensuite, être utilisée pour annoter le corpus.

A ce propos, nous notons que l’annotation d’un corpus à partir d’une concordance permet aux utilisateurs de vérifier, adapter et corriger les résultats d’un traitement automatique dont les sorties ne sont, pratiquement, jamais parfaites. Ceci est très important surtout lors d’une étape de préparation d’un corpus.

3.4 Les dictionnaires NooJ

Les dictionnaires NooJ (Silberztein M. 2003) contiennent indistinctement des mots simples et des mots composés qui peuvent représenter des variantes terminologiques ou phonétiques des entrées lexicales.

3.4.1 Les ALUs (Atomic Linguistic Units)

Les unités atomiques linguistiques sont les plus petits éléments qui forment une phrase. Ce sont des "mots" dont la signification ne peut pas être calculée ou prédite, on doit les apprendre pour être capables de les utiliser. On doit décrire explicitement toutes les propriétés syntaxiques et sémantiques de ces mots afin de pouvoir les analyser.

D’un point de vue formel, NooJ sépare les ALU en quatre classes :

- Les mots simples, ALU orthographiés sous forme de mots, tels que "pomme" ou "voiture" ;

- Les affixes, qui sont des ALU orthographiés comme des séquences de lettres, habituellement à l’intérieur des formes de mots, telles que dans "dé-" et "-isation" dans "dé-central-isation" ;
- Des mots composés, ALU orthographiés par des séquences de formes de mots, tels que "table ronde", "pomme de terre" ;

- Des expressions figées, ALU orthographiés comme des séquences discontiguës de formes de mots.

Afin de reconnaître les mots simples et les affixes, NOOJ effectue une recherche dans les dictionnaires. Les autres types seront identifiés en découplant les formes de mots en plus petites unités. Les quatre formes sont reconnues par NOOJ en appliquant des grammairies syntaxiques.

3.4.2 Ressources pour reconnaître les unités linguistiques atomiques

En ayant choisi la langue ici Fr, dans NooJ, la commande

```
Info → Preferences → Lexical Analysis
```


![Figure 3-1: Ressources "morphology" et "dictionary".](image)
3.4.2.1 Les Dictionnaires

La zone "Dictionary" contient toutes les ressources lexicales qui sont utilisées pour reconnaître les mots simples et les mots composés.

Les dictionnaires NooJ sont des fichiers ".nod" qui sont compilés à partir de fichiers source ".dic". Techniquement les fichiers ".nod" sont des transducteurs d’états finis, ils proviennent de fichiers de type texte ".dic" qui sont compilés en utilisant la commande

```
Lab → Dic → compile
```

3.4.2.2 La Morphologie

La zone "Morphology" (voir figure 3.2) affiche toutes les grammaires morphologiques qui sont utilisées pour reconnaître les formes de mots à partir de leurs composants (préfixes, affixes et suffixes). Ces ressources morphologiques sont des fichiers ".nom" constitués d’ensembles structurés de graphes qui décrivent des paradigmes morphologiques.

![Figure 3-2 : la zone "Morphology".](image)
3.4.3 Outils pour décrire la morphologie


3.4.3.1 Descriptions flexionnelles et dérivationnelles

Ce sont des fichiers ‘.nof’ organisés en règles libres de contexte qui décrivent les propriétés de chaque catégorie.

3.4.3.2 Grammaires flexionnelles et dérivationnelles

Elles sont représentées par des ensembles structurés de graphes qui décrivent des paradigmes morphologiques stockés dans des fichiers ‘.nof’.

3.5 Format des dictionnaires NooJ

Généralement, le dictionnaire d’un langage donné contient tous les lemmes du langage et les associe avec un code morpho-syntaxique, des codes sémantiques et syntaxiques possibles, des paradigmes flexionnels et dérivationnels.

3.5.1 Exemples d’entrées

Voici quelques exemples d'entrées :

cheval,N+FLX=Cheval+Anim
adhérent,N+FLX=Cousin+Hum,
ordinateur,N+FLX=Crayon+Conc
table,N+FLX=Table+Conc
voiture,N+FLX=Table+Conc

- La première entrée "cheval" est un nom, (catégorie "N"), elle se fléchit :
  ("cheval → chevaux") selon le modèle "cheval" qui est la valeur de la propriété "FLX", le trait sémantique est animal ("Anim").

- La troisième entrée "ordinateur" est un nom, (catégorie "N"), elle se fléchit :
  ("ordinateur → ordinateurs"), selon le modèle "crayon" qui est la valeur de la propriété "FLX", le trait sémantique est concret ("Conc").
3.5.2 Informations linguistiques

Dans les dictionnaires NooJ, toutes les informations linguistiques telles que les codes syntaxiques tels que "+tr" (transitif) et les codes sémantiques tels que "+conc" (concret) et "+abst" (abstrait) doivent être précédés du caractère "+".

3.5.3 Codes d’information spéciaux

Le code "+ NW " (Non-Word) est utilisé pour décrire des entrées lexicales abstraites qui n’apparaissent pas dans les textes et qui ne devraient pas être analysées comme des formes réelles de mots. Cette forme est utile pour la construction d’un dictionnaire dans lequel des entrées sont des stems ou des racines de mots.

Exemples d’entrées :
Fortiori,NW,LATIN
Priori,NW,LATIN

Le code "+UNAMB" (unambigous ou mot non ambigu) indique à NOOJ de stopper l’analyse de la forme du mot avec soit les ressources lexicales soit l’analyseur morphologique. Par exemple sur les entrées suivantes l’analyse s’arrête dès la rencontre de "UNAMB" pour le mot table :
onde,A
table ronde,N+UNAMB+Abst

Le code "+FLX" (Paradigme flexionnel) signifie que l’entrée se fléchit comme table. Par exemple dans la troisième ligne de l’exemple précédent où Table est associé avec la propriété +FLX=Table, Table est un paradigme flexionnel.

Table = <E>/singulier + s/pluriel

Ce qui indique que "Table" a pour singulier "table" et pour pluriel "tables".

Le code "+DRV" (Paradigme dérivationnel) permet aux lexicographes d'associer chaque mot avec les variantes dérivationnelles correspondantes dans les versions NooJ des dictionnaires DELA. Des dérivation morphologiques seront ensuite décrites avec précision pour chaque entrée, par exemple, soient les entrées lexicales suivantes :
Lier des formes dérivées dans le dictionnaire permettra d'avoir un dictionnaire moins volumineux, par exemple, si la fonction "+ DRV = RE" est associée avec le verbe "donner", Il devient inutile d'ajouter une entrée lexicale pour "redonner". "Monter" se dérive en "remonter", qui se conjugue de façon identique. "Voler" se dérive en "volaible", qui se fléchit comme "Artiste". "Mangeable" se fléchit comme "Artiste".

3.5.4 Propriétés lexicales

Non seulement des formes, mais aussi des propriétés peuvent être associées aux entrées lexicales. Une propriété lexicale a un nom et une valeur exprimés sous la forme "+nom=valeur". Par exemple maison a pour paradigme flexionnel "+FLX=TABLE" et se fléchit selon le modèle TABLE. Les paradigmes flexionnels dérivationnels sont décrits au chapitre 7.

3.5.5 Variantes lexicales

Les dictionnaires NooJ ne sont plus limités à un seul champ : ils peuvent contenir des entrées liées à un «super-lemme», qui peut être une variante orthographique, la traduction dans une autre langue, une entrée synonyme ou hyperonyme. Considérons par exemple les entrées lexicales suivantes :

ONU, Organisation des nations unies
czar, tsar, N + Hum+ FLX = Crayon

La première entrée ("ONU") est associée à un super-lemme "Nations Unies", il ne peut pas être fléchi. La deuxième entrée ("czar") est associée au super-lemme "tsar", il se fléchit selon le paradigme "Crayon" (c'est-à-dire qu'il prend un "s" au pluriel).

NooJ offre la capacité d'associer des mots avec des super-lemmes, c'est-à-dire des lemmes qui ne correspondent pas nécessairement à leur lemme de flexion ("czar" est le lemme de czars, et non pas "tsar").

3.6 Fichiers de définition des propriétés "DEF"

Dans le dictionnaire, l'instruction suivante indique où sont définies les associations catégories, propriétés et leurs valeurs :

```
donner,V+Tr+FLX=Aimer+DRV=RE+DRV=ABLE:
monter,V +tr+FLX=Aimer+DRV=Re
voler,V+tr+FLX=Aimer+DRV=Re+DRV=Able:Artiste
manger,V+tr+FLX=Manger+DRV=Re+DRV=Eable:Artiste
```

Dans le dictionnaire, l'instruction suivante indique où sont définies les associations catégories, propriétés et leurs valeurs :
Les utilisateurs peuvent créer de nouveaux codes d’information tels que "+Info" "+Politiqu" et peuvent les ajouter à des entrée lexicales. On peut aussi utiliser ces codes d’informations qui sont ajoutés sous forme de propriété avec une valeur, telle que "+Domain=Info".

Exemples :
- V_Nb = s|p signifie que le nombre du verbe est singulier ou pluriel ;
- V_Pers = 1|2|3 signifie que la personne du verbe est 1ère, 2ème ou 3ème.

3.7 Représentation formelle des dictionnaires électroniques

Les automates d’état finis (FSA) sont utilisés dans beaucoup d’applications de traitement du langage naturel (NLP). En particulier, ils fournissent un stockage et une récupération efficaces des ensembles finis de chaînes sur un alphabet fini, et peuvent donc être utilisés pour représenter le vocabulaire d'une langue.

L'utilisation des automates peut être étendue à l’utilisation des transducteurs d’état fini (FST) qui permet aux utilisateurs d'associer à chaque élément d'un vocabulaire des informations, telles que des informations morpho-syntactiques (genre, nombre, etc.), des informations syntaxiques et sémantiques (par exemple, transitif, humain, etc.), ou plus complexes, telles que la traduction des termes dans d'autres langues, un ensemble d'expressions synonymes, etc.

Le but d'une analyse lexicale d'un texte est de mettre en correspondance les termes de ce texte avec ceux d'un vocabulaire généralement décrits dans un dictionnaire. Dans la plupart des langues, les tokens sont des formes de mots fléchis et / ou dérivés qui nécessitent d’être associés au lexème correspondant (entrée de dictionnaire) et quelques informations linguistiques. Ainsi, les transducteurs sont bien adaptés pour l’exécution d’analyses automatiques lexicales. Les FSA peuvent être facilement étendus en FST pour produire des informations associées aux tokens acceptés : la sortie du transducteur est tout simplement mise en affixe aux états acceptés correspondants du FSA. (Roche E. 1993) utilise les transducteurs pour l’analyse transformationnelle du français.

Les unités linguistiques de NooJ sont reconnues en effectuant une recherche dans les dictionnaires, et aussi bien en utilisant des grammaires morphologiques pour analyser les formes du mot, ainsi qu’en utilisant les grammaires syntaxiques des phrases à analyser.

Sur le plan linguistique, l’intérêt des graphes ou des automates est de regrouper dans une même description des éléments équivalents du point de vue du sens, mais différents au niveau
de la forme, par exemple les variantes graphiques de mots dont l'orthographe n'est pas normalisée :

- (bistrot | bistro),
- (tsigane | tzigane),

ou bien des variantes d'expressions de temps :

(avec le recul du temps | avec le recul des années).

### 3.8 Conclusion

Les dictionnaires électroniques sont construits en vue du traitement automatique des langues naturelles.

Le système NooJ est un environnement de développement linguistique, doté d’un moteur linguistique robuste, pouvant prendre en charge une douzaine de langues variées. C’est un outil puissant, la plate-forme conviviale permet le développement et la mise au point de diverses applications :

- linguistiques,
- d’aide à l’enseignement des langues,
- d’aide à la traduction,
- d’extraction de terminologie.
Chapitre 4 :
Aspects linguistiques de la terminologie
4 Aspects linguistiques de la terminologie

4.1 Introduction

Lors de l’extraction de terminologie qu’elle soit manuelle ou automatique, il faut examiner si les candidats termes extraits peuvent constituer des entrées de dictionnaires. La tâche est délicate, dans ce chapitre nous examinons divers travaux sur les approches qui pourraient aider le terminologue lors l’extraction et de la validation des termes, afin de distinguer un mot composé d’un groupe "nominal libre". Les critères recensés restent complexes et parfois difficiles à appliquer.

4.2 Définitions

4.2.1 Terminologie

Selon ISO\textsuperscript{29} (1990), la "terminologie" est définie comme l’étude scientifique des notions et des termes en usage dans les langues des spécialistes.

Dans (Fotopoulo A. 1995), on trouve qu’«une langue spécialisée est l’usage d’une langue qui permet de rendre compte techniquement de connaissances spécialisées. Ces connaissances spécialisées sont dénommées linguistiquement par des termes qui sont des mots, des groupes de mots».

Selon (Larousse) la terminologie est un ensemble des termes propres à une technologie, une Science.

Dans le dictionnaire de Linguistique Larousse (Dubois J. et al. 1973, 1994), nous trouvons que toute discipline, et à plus forte raison toute science, a besoin d’un ensemble de termes, définis rigoureusement, par lesquels elle désigne les notions qui lui sont utiles: cet ensemble de termes constitue sa terminologie.

4.2.2 Mot

Un "mot" est un son ou groupe de sons servant à désigner un être, une idée.

4.2.3 Terme

"Terme" est la désignation d’une notion sous forme de lettres, de chiffres, de pictogrammes ou d’une combinaison quelconque de ces éléments ISO (1990).

\textsuperscript{29} International organization for standardization (organisation internationale de standardisation).
Selon (Larousse), un terme est un mot qui a un sens strictement délimité à l’intérieur d’un système de notions donné.


4.2.4 Forme simple (définition orthographique)

Une forme simple est une séquence consécutive non vide de caractères de l’alphabet apparaissant entre deux séparateurs.

4.2.5 Mot simple

Un mot simple de la langue est une forme simple qui constitue une entrée d’un dictionnaire des mots simples fléchis de cette langue.

Exemples :
- pomme,
- chat,
- table,
- mange.

4.2.6 Forme composée

Une forme composée est une séquence consécutive d’au moins deux formes simples et de blocs de séparateurs.

4.2.7 Mot composé

Un "mot composé" est une forme composée qui constitue une entrée du dictionnaire des mots composés fléchis de cette langue.

Exemples :
- Pomme de terre,
- avion à réaction,
- voiture de course.

Remarques :
- Cette définition nécessite d'examiner les problèmes linguistiques de la notion de composition et de définir quel composé constitue une entrée du dictionnaire à construire.
- Les mots composés peuvent être des adjectifs composés, verbes composés, noms composés, conjonctions composées ou adverbes composés. Dans ce qui suit nous
utiliserons indifféremment les termes mots composés et noms composés, la plupart des mots composés d'informatique étant des noms composés.

4.2.8 Synapsie

Certains noms composés complexes ont été baptisés "symapsies" par (Benveniste E. 1966). Une synapsie consiste en un groupe entier de lexèmes, reliés par divers procédés, et formant une désignation constante et spécifique. On trouve le noyau initial dans des exemples déjà anciens comme : "pomme de terre", "robe de chambre", "clair de lune", "plat à barbe". Le fait nouveau et important est que ce type de composition prend aujourd'hui une extension considérable et qu'il est appelé à une productivité indéfinie : il sera la formation de base dans les nomenclatures techniques.

Les synapsies sont des locutions, ou groupes de mots construits sur des substantifs ou des verbes. Elles sont utilisées pour effectuer le dépouillement terminologique d'un texte, pour cerner un paradigme particulier (celui construit autour du "mot d'entrée"). Les synapsies ont en effet la propriété de saturer le sens d'un vocable. Comparez "santé économique" et "système de santé". Dans la première synapsie, "santé" est métaphorisé alors qu'il ne l'est pas dans la seconde.

4.3 Mots composés et notions de composition

4.3.1 Mots composés


Exemples :
- Adjectifs composés: aigre-doux, aigres-doux, aigre-douce, aigres-douces;
- Verbes composés: copier-coller.

4.3.2 Notion de composition

Il n'existe pas de définition unique d’un nom composé mais un certain nombre de propriétés communes si ce n'est qu'un terme peut être simple s'il ne contient qu'un seul mot, ou composé s'il en contient plus d'un et qu'un mot composé est construit à partir de mots simples (Daille B. 1994).

La notion de composition nominale, très controversée a été abordée dans divers travaux ((Cadiot P. 1992), (Corbin D. 1992), (Habert et al., 1993), (Bauer L. 1988), (Grévisse M. 1988)). Parmi ces travaux (Levi J. 1978) énonçait qu’il n’était pas possible de distinguer les

On remarque que tous les travaux de recherche réalisés tentent de définir la notion de composition d’un nom composé mais ne fournissent pas de méthode opérationnelle pour leur reconnaissance (Daille B. 1994). (Gonzales Rey M., M. Lopez Diaz. 2003) traitent de l’opacité des séquences figées. La principale problématique associée aux noms composés est de pouvoir détecter si un groupe de mots constitue un terme pour la science ou discipline retenue.

Le problème est complexe pour le linguiste, aussi l’aide du spécialiste de la terminologie étudiée est souvent d’un grand secours. Nous allons examiner les mots composés - plus particulièrement les noms composés - et la distinction entre groupe nominal libre et nom composé. Nous verrons par exemple les critères définis par Gaston Gross et ceux de Max Silberztein, ils constituent une aide pour distinguer un nom composé d’un groupe nominal libre.

4.4 Degré de figement des noms composés

On notera tout d’abord que la majorité des mots composés sont des noms composés. Dans (Gross G. 1990), des critères permettant de calculer le degré de figement des noms composés de type "N de N" sont présentés. Les groupes "N de N" sont analysés dans (Gross G. 1991).

Différents critères sont retenus pour décider si on a un nom composé que l’on pourra rajouter dans un dictionnaire ou un groupe nominal libre. Ils sont principalement fondés sur l'analyse des relations entre les deux noms. Nous exposons ces critères ci-dessous.

4.4.1 "Il n'y a pas de relation syntaxique entre les 2 noms"

On n’a pas de figement si dans un groupe N De N, on peut établir une relation syntaxique entre les deux noms. Il y a deux cas :

- N1 est un concret

Exemples :

Le briquet de Luc.

Luc a un briquet.

Un pied-de-biche.

La biche a un pied.* , ici on a un nom composé, pied de biche

- N1 est un substantif prédicatif : "N1 de N2"

Exemple :

Le voyage de Luc.
Diverses transformations sont possibles :

- **Nominalisation**: Luc a fait un voyage en Italie;
- **Relativation**: Le voyage que Luc a fait en Italie;
- **Réduction du verbe support**: Le voyage de Luc en Italie.

Dans ces cas, N2 est sujet du substantif prédicatif N1, dans les cas suivants, N2 peut représenter l’objet.

Exemple:
L’aménagement du territoire. *On a procédé à l’aménagement du territoire.*

S’il y a une relation syntaxique entre les 2 substantifs d’un GN, chacun des éléments est susceptible de certaines transformations, dont la pronominalisation.

### 4.4.2 Pronominalisation

S’il y a une relation syntaxique entre les 2 substantifs d’un GN, chacun des éléments est susceptible de certaines transformations, dont la pronominalisation, nous avons deux cas, la pronominalisation de N2 et la pronominalisation de N1 :

- **Pronominalisation de N2** :

  N2 peut être pronominalisé par un possessif

  Exemples :

  La voiture de Luc.
  *Sa voiture.*

  Je connais les résultats de cette expérience.
  *J’en connais les résultats.*

- **Pronominalisation de N1** :

  Dans les groupes nominaux libres, N1 peut être pronominalisé, en particulier dans le cas de la coordination. L’absence de pronominalisation est un indice de figement.

  J’ai un livre d’histoire et de géographie.
  *Mon livre d’histoire et de géographie.*

### 4.4.3 Figement partiel

Deux cas sont possibles : le figement partiel du 1er nom ou le figement partiel du 2ème nom.
Figement partiel du 1er nom :

Exemple :

Luc a versé un nuage de lait.

$\textit{Luc a versé du lait.}$

$\textit{*Luc a versé un nuage.}$

Figement partiel du 2ème nom :

Exemple :

Luc a une fièvre de chacal.

$\textit{*Cette fièvre est de chacal}$

4.5 Les groupes nominaux productifs et les noms composés lexicalisés

(Silberztein M. 1993a) a défini des critères utilisés pour définir un nom composé. L’apport sémantique des composants du GN est examiné, dans le cas des mots composés généralisés (séquences figées) les constituants sont des entrées dans le dictionnaire des mots simples DELAS.


4.5.1 L’atomicité sémantique

Afin de décider si un groupe de mots est un mot composé ou une séquence libre, il faut comparer les propriétés syntaxiques et sémantiques des séquences « candidates au statut de mot composé » aux propriétés générales des séquences libres.

Si toutes les propriétés syntaxiques et sémantiques d’une séquence peuvent être déduites des propriétés de ses constituants, on n’a pas besoin de les décrire dans un dictionnaire, la séquence est libre.

En revanche si au moins une propriété syntaxique ou sémantique ne peut pas être calculée, il faut associer explicitement cette séquence à ses propriétés et donc la placer dans le dictionnaire : la séquence est figée.

Exemples :

Une carte ancienne. – $\textit{Une carte qui est ancienne.}$

Une carte bancaire. – $\textit{*Une carte qui est bancaire.}$

On ne peut pas la paraphraser avec la structure à verbe être.
4.5.2 L’institutionnalisation de l’usage

4.5.2.1 Termes institutionnalisés

Ces termes sont institutionnalisés bien que semblant syntaxiquement et sémantiquement analysables.

Un groupe nominal plus ou moins figé (terme) désigne de nombreux éléments de notre environnement.

Exemples :
- Des chefs d’état ;
- Un employé de banque ;
- Une machine à écrire ;
- Une torche électrique.

On ne dira pas des "chefs de nation", des "chefs de pays", un "employé bancaire", une "machine d’écriture", ou une "torche à électricité".

4.5.2.2 Termes inexistants

On ne peut désigner un certain nombre d’éléments de notre entourage que par des groupes nominaux libres ou des paraphrases, car les termes n’existent pas.

Exemples :

Il a mis une chemise qui a un motif à voitures.* Il a mis une chemise à voitures.

4.5.3 Restrictions distributionnelles

Dans le dictionnaire DELAC, on pourra avoir des entrées telles que :

(NA) un produit (actif+adhésif+agricole+alimentaire+apicole…)

Ces groupes nominaux seront reliés à d’autres groupes nominaux par des transformations. (Gross G. 1991, Monceaux A. 1993, 1997) :

(NA) (une production+exportation+excédent+ …. ) Adj = : une production agricole.

Afin d’effectuer ces transformations, des classes distributionnelles sont à construire, par exemple les classes :

85
- AdjProduit (adjectifs de produits tels que agricole, alimentaire, apicole, laitier …etc.),
- Nprof (noms de profession).

On se reportera à (Gross M. 1986), pour plus de détails sur leur construction.

### 4.5.4 Analyse transformationnelle

Par exemple avec les classes NA et NDN nous aurons :

- Un accord monétaire. – *un accord sur des monnaies*,
- Un tube de verre. – *Un tube en verre*.

### 4.6 Variantes terminologiques

Plusieurs catégories de variantes existent (parmi elles, les abréviations et les sigles qui seront vus au chapitre 5). Nous considérons ici, pour les noms composés, les variantes terminologiques avec :

#### 4.6.1 La surcomposition

Un composé se voit rajouter un modifieur ou une tête pour créer un terme plus complexe,

Exemple :

réseaux de microstations → connexion de [réseaux de microstations] ;

#### 4.6.2 Les insertions

Un modifieur est inséré à l’intérieur d’un composé :

- insertion d’un adjectif :

  Exemple :

  réseau d’ordinateurs → réseau [local] d’ordinateurs

- insertion d’un nom :

  Exemple :

  protocole TCP/IP → protocole [internet] TCP/IP
4.6.3 La coordination

Un terme est coordonné avec un autre terme par la mise en facteur de la partie commune. (Domingues C. 2001) analyse les problèmes de reconnaissance des termes coordonnés.

Exemples :

Puissances de calcul et de stockage → puissance de calcul et puissance de stockage.

Elévateurs et abaisseurs de fréquence → Elévateurs de fréquence | abaisseurs de fréquence.

4.7 Conclusion

L’aspect linguistique de la terminologie pour la constitution de dictionnaires spécialisés est examiné dans ce chapitre. Toute élaboration de dictionnaire de terminologie (dans notre cas de l’informatique) nécessite de bien penser comment extraire et sélectionner les termes valides. Nous avons présenté diverses approches pour "détecter" un terme en utilisant les notions de composition et de degré de figement. Cependant le problème reste entier car aucune approche ne peut être reconnue comme étant une méthode. Nous noterons que la notion de composition nominale a été très controversée.

La diversité des domaines de spécialité et leur complexité font que chaque domaine devra être traité à part. Une collaboration entre terminologue, linguiste, spécialiste du domaine serait idéale pour la construction de tout dictionnaire de terminologie.

Nous remarquerons que les noms composés peuvent être des abréviations, des sigles, des emprunts à d’autres langues et des onomatopées, ils seront mis dans le dictionnaire des noms composés ou à part. Une étude leur est consacrée au chapitre 5.
Chapitre 5 :
Elaboration des dictionnaires électroniques de la terminologie informatique
5 Elaboration des dictionnaires électroniques de la terminologie informatique

5.1 Introduction

Selon (Gross M. 1999) trois sortes de données lexicographiques sont disponibles sous forme électronique :

- Les dictionnaires édités, ils peuvent être monolingues ou bilingues. Ils sont disponibles sur papier, les éditeurs fournissent quelquefois une version électronique sur CD-ROM, ainsi ils peuvent être consultés et être imprimés. Dans certains cas, il est permis de télécharger des dictionnaires et les fichiers peuvent être traités par un logiciel de traitement de texte. Les données doivent obligatoirement être codées avant d’être ajoutées aux dictionnaires électroniques ;

- Les dictionnaires électroniques, ces dictionnaires sont construits pour une utilisation par des programmes, leur contenu est constitué de codes alphanumériques représentant les données grammaticales ;

- Les corpus, des textes (littérature, journaux, revues) sont publiés sur CD/ROM. L’Internet offre actuellement une grande source de corpus téléchargeables.

En terminologie, les unités lexicales représentent les concepts d’un domaine auquel appartiennent les textes dont elles sont issues. Nous allons décrire les dictionnaires électroniques de terminologie informatique que nous avons construits. Des dictionnaires décrivant une douzaine de langues naturelles existent dans NOOJ. Plus particulièrement, les dictionnaires et ressources pour la langue française ont été présentés dans le chapitre 2. Les mots simples d’informatique français se trouvent en partie dans le DELAS.

Les dictionnaires spécialisés des mots composés d’informatique en français sont inexistants dans NOOJ. Nous nous intéressons à la construction des dictionnaires d’informatique, adaptés au traitement automatique, en vue d’analyser des textes techniques.

Les dictionnaires électroniques sont, avant tout, construits pour être utilisés par des programmes informatiques d’analyse linguistique. Pour réduire au maximum les situations d’échec, leur élaboration doit être minutieuse avec un contenu aussi complet que possible. Les dictionnaires électroniques doivent contenir suffisamment d’informations rattachées aux entrées lexicales et utilisables pour l’analyse automatique (Mesfar S. 2008).

La mise en œuvre des dictionnaires électroniques dans NooJ se base sur quatre parties essentielles :
- Une liste d’entrées lexicales : la transcription de cette liste sera détaillée dans la section suivante ;

- Une liste de propriétés : ces codes désignent l’ensemble des informations flexionnelles, morphologiques, syntaxiques et sémantiques convenablement choisies par le concepteur pour être rattachées à aux différentes formes du lexique ;

- Des descriptions formelles des règles de flexion et dérivation automatique des lemmes : ces paradigmes sont associés de façon univoque à des codes alphanumériques permettant de les désigner ;

- Des opérateurs de transformations morphologiques et phonologiques : ils sont employés par les routines de flexion et de dérivation automatiques, élaborées sur la base des algorithmes de flexion. Nous pouvons distinguer aussi bien des transformations génériques que des transformations spécifiques à la langue du dictionnaire.

5.2 Dictionnaires usuels et dictionnaires spécialisés

Dans les dictionnaires usuels les descriptions informelles sont loin d’être suffisantes pour l’analyse automatique de textes. Les dictionnaires spécialisés ajoutent à ces descriptions un certain nombre de termes techniques qui peuvent aider un utilisateur humain mais pas un programme d’informatique qui doit avoir accès à des listes exhaustives de termes comprenant leurs variantes et abréviations, dans le but de faire correspondre le contenu des textes (Gross M. 1999). Aussi une quantité plus grande d’informations que celles contenues dans les dictionnaires usuels est nécessaire pour les applications telles que la recherche d’informations et la traduction automatique.

Selon (Hadouche F., L’Homme M.C., Lapalme G., Le Serrec A. 2009), les dictionnaires spécialisés et les bases de données terminologiques, quoique riches en information de nature conceptuelle, fournissent en général très peu de renseignements sur les propriétés linguistiques des termes ou sur leur comportement en langue.


Les bases de données lexicales sont de plus en plus sollicitées dans de nombreuses applications en traitement des langues naturelles, aussi, il devient nécessaire de fournir une description explicite et formelle de l’ensemble des propriétés linguistiques des unités lexicales qu’elles soient générales ou spécialisées. Afin de constituer les dictionnaires électroniques qui seront utilisés par des programmes informatiques, il faut recenser les termes et les introduire sous forme codifiée dans les dictionnaires.

30 http://id.erudit.org/iderudit/008735ar
5.3 Extraction des termes

L'extraction de terminologie consiste à identifier des termes potentiels dans un texte spécifique ou un ensemble de textes (corpus) ainsi que les informations pertinentes liées à l'emploi de ces termes. L'extraction de terminologie s'accompagne manuellement ou automatiquement (Aoughlis F. 2005). L'extraction automatique est faite à l'aide d'outils d'extraction terminologique examinés au chapitre 1.

Pour construire notre dictionnaire "Info_comp.dic", nous avons dans un premier temps extrait les termes manuellement à partir de textes. En utilisant N00J, nous avons constaté qu'il peut être utilisé pour l'extraction automatique, à l'aide de patrons syntaxiques appropriés. Nous effectuons aussi l'extraction de terminologie à l'aide de N00J (sera vue plus tard au chapitre 8).

Les termes d'informatique sont sous forme de mots simples, de mots composés, sigles et abréviations. Les mots composés sont regroupés dans le dictionnaire "Info_comp.dic", séparément des sigles et abréviations qui sont enregistrés dans des dictionnaires séparés "sigles.dic" et "abréviations.dic".

5.4 Dictionnaire des mots composés "Info_comp.dic"

Un terme candidat extrait ne devient une entrée du dictionnaire qu’après avoir été examiné. Le spécialiste du domaine doit décider, si oui ou non il constitue une entrée. On pourra tenir compte des critères présentés dans le chapitre 4, pour décider. Les entrées du dictionnaire sont les termes retenus après l'extraction.

Dans le cas d'entrée valide, on effectue l'acquisition, c'est-à-dire la codification manuelle de l'entrée dans le dictionnaire.

Nous étudions les déterminants possibles pour le terme, d’après l’emploi fait dans les textes et les mots constituant le terme. La notion de tête peut être utilisée pour déduire le(s) déterminant(s) et aussi le genre et le nombre.

Le dictionnaire électronique des mots composés d'informatique que nous avons construit est décrit dans (Aoughlis F., Métais E. 2006).

5.4.1 Caractéristiques des composants des noms composés (notion de tête)

Selon (Savary A., 2000), la tête d’un nom composé est constituée de composants qui ont les mêmes formes morphologiques que le mot composé lui-même. Parfois il est difficile de décider lequel est la tête. Pour les longs noms composés (longueur 3 et plus) il est nécessaire d’étudier le sens pour trouver la tête.
### Structure

<table>
<thead>
<tr>
<th>Nom Adjectif</th>
<th>Adjectif Nom</th>
<th>Nom1 de Nom2</th>
<th>Nom1 à Nom2</th>
<th>Nom1Préposition Nom2</th>
<th>Préposition Nom</th>
<th>Nom1 Nom2</th>
<th>Verbe Nom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tête</td>
<td>Exemples</td>
<td>Carte mère</td>
<td>Grand-mère</td>
<td>Partage de temps</td>
<td>Arrière-plan</td>
<td>Mémoire partagée.</td>
<td>Pas de tête</td>
</tr>
</tbody>
</table>

**Tableau 5-1: Tête des mots composés.**

### 5.4.2 Les déterminants possibles pour les noms composés

Les déterminants des noms composés sont recensés dans (Silberztein M. 1993). Neuf formes sont possibles (avec m: masculin, p: pluriel, f: féminin, s: singulier):

- E associé à un nom propre sans déterminant, par exemple Action/directe, E (NA), *toujours au masculin/féminin singulier*,

- le, un & et de&le, *le nom composé est ms*,

- la, une & et de&la, *le nom composé est fs*,

- les, de, *le nom composé est mp ou fp*.

**Exemples :**

- Masculin singulier :
  **Le programme translatable, un programme translatable ;**

- Féminin singulier :
  **La mémoire auxiliaire, une mémoire auxiliaire ;**

- Masculin pluriel :
  **Les programmes translatables, des programmes translatables ;**

- Féminin pluriel :
  **Les mémoires auxiliaires, des mémoires auxiliaires ;**

### 5.4.3 Notion de mot vide, mot plein

Dans les systèmes de recherche automatique de documents, on appelle mots-vides les mots qui ne sont pas indexés, soit parce qu'ils ne sont pratiquement pas porteurs d'information (exemple : un article le, la, les,…), soit parce qu'ils sont d'un usage si courant qu'on est certain
de les retrouver dans tous les documents (exemple : les verbes être et avoir et leurs conjugaisons), soit parce qu'ils appartiennent au langage argotique ou familier.

Dans son ouvrage, après une critique en règle de la classification traditionnelle des mots en parties du discours, (Tesnière L. 1969) 31 divise les mots de la langue en *mots pleins*, qui sont chargés d'une fonction sémantique, d'un sens, et en *mots vides*, qui ne le sont pas. Seuls les premiers sont susceptibles de constituer un nœud (élément qui a sous sa dépendance un autre élément): ce sont les verbes, les substantifs, les adjectifs qualificatifs, certains pronoms et adverbes suffisamment autonomes (*moi, aujourd'hui*). Les mots vides sont interdits de position de nœud: ce sont les déterminants, les pronoms et les adverbes non autonomes (*je, très*).

(Vergne J. 2004) reprend la typologie des mots venant de (Tesnière L. 1969) qui définit deux types de mots :

- les mots pleins, "chargés d'une fonction sémantique" (noms, verbes, adjectifs, adverbes),
- les mots vides, "simples outils grammaticaux" comprenant tous les autres mots (déterminants, prépositions, pronoms),

afin de développer un système d’extraction de candidats termes dans des corpus bruts de langues non identifiées par étiquetage mot vide - mot plein.

Dans notre cas, les mots considérés comme vides n’entrent pas en jeu dans la longueur du mot composé. Nous avons par exemple la préposition qui fait partie du composé mais dont la longueur ne compte pas.

Exemples :

- Mémoire de masse ;
- Programme de traitement ;
- Carte à puce ;
- Décalage à droite.

Dans les exemples précédents, les mots "de" et "à" sont des mots vides, les autres sont des mots pleins.

---

5.4.4 Etude et classification syntaxique des termes

5.4.4.1 Codes grammaticaux

Nous retiendrons dans le tableau 5.2, les codes grammaticaux (Silberztein M. 2003) pouvant apparaître dans les mots composés d’informatique :

<table>
<thead>
<tr>
<th>Code</th>
<th>Catégorie</th>
<th>Exemples</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adjectif</td>
<td>Auxiliaire, translatable</td>
</tr>
<tr>
<td>ADV</td>
<td>Adverbe</td>
<td>Mieux</td>
</tr>
<tr>
<td>CONJC</td>
<td>Conjonction de coordination</td>
<td>Et, ou, donc</td>
</tr>
<tr>
<td>N</td>
<td>Nom</td>
<td>Mémoire, disque, registre</td>
</tr>
<tr>
<td>PFX</td>
<td>Préfixe</td>
<td>Pseudo</td>
</tr>
<tr>
<td>PREP</td>
<td>Préposition</td>
<td>De, à, en</td>
</tr>
<tr>
<td>PRO</td>
<td>Pronom</td>
<td>Je, vous,</td>
</tr>
<tr>
<td>V</td>
<td>Verbe</td>
<td>Ecrire, lire, afficher</td>
</tr>
</tbody>
</table>

Tableau 5-2 : Codes grammaticaux utilisés

A une entrée du dictionnaire des termes d’informatique est associée une catégorie syntaxique, nous examinons pour chaque terme retenu ses composants syntaxiques.

Un mot composé est construit à partir de mots simples, aussi, la catégorie syntaxique du composé est construite avec les catégories syntaxiques des mots simples qui le constituent.

Exemple :

- le terme "adressage absolu" contient deux mots simples, "adressage" qui est un nom noté "N" et "absolu" qui est un adjectif, noté "A", la classe du composé est "NA" pour "Nom adjectif", il est de longueur 2.

5.4.4.2 Longueur 2

Nous trouvons ci-dessous les classes de noms composés de longueur deux que nous avons recensés. Ce sont les plus nombreux.

<table>
<thead>
<tr>
<th>Classe</th>
<th>Forme</th>
<th>Exemple</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>Nom Adjectif</td>
<td>Adressage absolu</td>
</tr>
<tr>
<td>NN</td>
<td>Nom1 Nom 2</td>
<td>Mémoire tampon</td>
</tr>
<tr>
<td>NPN</td>
<td>Nom Préposition Nom</td>
<td>Adressage par octets</td>
</tr>
<tr>
<td>NDN</td>
<td>Nom1 de Nom2</td>
<td>Formatage de données</td>
</tr>
<tr>
<td>AN</td>
<td>Adjectif Nom</td>
<td>Arrière-plan</td>
</tr>
<tr>
<td>PN</td>
<td>Préposition Nom</td>
<td>Sous programme</td>
</tr>
</tbody>
</table>
Par exemple, dans la classe NA (Nom Adjectif), nous trouvons aussi les termes :

Adressage direct,
Adressage indirect,
Adressage indexé,
Programme translatable,
Programme absolu …,

**5.4.4.3 Longueur 3**

Nous trouvons ci-dessous les composés de longueur trois que nous avons recensés.

<table>
<thead>
<tr>
<th>classe</th>
<th>Forme</th>
<th>Exemple</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAPN</td>
<td>Nom1 Adjectif Préposition Nom2</td>
<td>Allocation dynamique de mémoire</td>
</tr>
<tr>
<td>NPNPN(^{32})</td>
<td>Nom1 Préposition1 Nom2 Préposition2 Nom3</td>
<td>Allocation de ressources en batch (La préposition peut être vide)</td>
</tr>
<tr>
<td>NPAAN</td>
<td>Nom1 Préposition1 Adjectif1 Adjectif2</td>
<td>Animation en fondus chaînés</td>
</tr>
<tr>
<td>NPNPN</td>
<td>Nom1 Préposition Adverbe Adjectif</td>
<td>Algorithmme du mieux adapté</td>
</tr>
<tr>
<td>NdePN</td>
<td>Nom1 Préposition1 Préposition2 Nom2</td>
<td>Appel de sous programme</td>
</tr>
<tr>
<td>NPNA</td>
<td>Nom1 Préposition Nom2 Adjectif</td>
<td>Architecture à mémoire partagée</td>
</tr>
<tr>
<td>NPAN(^{32})</td>
<td>Nom1 Préposition Adjectif Nom</td>
<td>Bit de demi-retenu</td>
</tr>
<tr>
<td>NNXA</td>
<td>Adverbe Nom Adjectif</td>
<td>Interruption non masquable</td>
</tr>
<tr>
<td>NAA</td>
<td>Nom Adjectif1 Adjectif2</td>
<td>Circuit intégré décimal</td>
</tr>
<tr>
<td>XNPN</td>
<td>Adverbe Nom1 Préposition Nom2</td>
<td>Non retour à zéro</td>
</tr>
<tr>
<td>NACA</td>
<td>Nom Adjectif1 conjonction Adjectif2</td>
<td>Unité arithmétique et logique</td>
</tr>
</tbody>
</table>

\(^{32}\) La préposition peut être vide.

**Tableau 5-3 : Description des classes syntaxiques pour les composés de longueur 2.**

**Tableau 5-4 : Description des classes syntaxiques pour les composés de longueur 3.**
5.4.4.4 **Longueur 4 (termes contenant 4 mots pleins)**

En ce qui concerne les composés de longueur 4, ils sont moins nombreux.

- Classe NX4

Exemple :

Algorithm pr\'eemptif à priorité dynamique.

Les mots pleins sont algorithme, préemptif, priorité et dynamique.

5.4.4.5 **Longueur 5 (termes contenant 5 mots pleins)**

Ces composés ne sont pas nombreux.

- Classe NX5

Exemple :

Traitement différé à flot de travail unique.

Les mots pleins sont traitement, différé, flot, travail, unique.

5.4.5 **Format d’une entrée**

Toute entrée lexicale dans NooJ est constituée d’un ensemble de données (Mesfar S. 2008), nous citons :

- Un lemme : considéré comme forme de base ;
- Une étiquette : qui en indiquera la catégorie grammaticale d’appartenance ;
- Une liste optionnelle d’informations syntactico-sémantiques ;
- Une liste éventuelle de codes alphanumériques désignant les modèles flexionnels et dérivationnels à y appliquer.

En dépit de la présence indispensable des deux premières parties de l’entrée, nous signalons que l’apparition des deux dernières dépend des propriétés de l’entrée lexicale.

Nous donnons, figure 5.1, les extraits du dictionnaire "Info\_comp-v3.dic" et du fichier "composés-Flex.nof" que nous avons élaborés.
Par exemple, pour le terme retenu "accès aléatoire" (random access) nous avons pour l’entrée extraite du dictionnaire Info_comp-v3.dic (Fig. 6.1) :

\[
\text{accès aléatoire, N+NA+info+FLX=AccesAccordé+Hild93}
\]

- le lemme "accès aléatoire" est un terme,
- la catégorie est N ou NA : "N+NA" ; Nom ou Nom adjectif,
- "+info" est une information sémantique indiquant que l’entrée du dictionnaire est un terme d’informatique,
- "+FLX" donne le nom du modèle flexionnel, ici "AccesAccordé", défini dans le fichier des descriptions flexionnelles composes-flex.nof,
- Hild93 est la référence bibliographique d’où a été extrait le terme.

En appliquant le modèle flexionnel AccesAccordé (extrait du fichier "composes-flex.nof" de la figure 5.2) :

\[
\text{AccesAccordé} = \langle E \rangle /m+s + s/m+p;
\]
À l’entrée “accès aléatoire” du dictionnaire, on a deux formes fléchies :

- m+s → accès aléatoire masculin (m) singulier (s)
- m+p → accès aléatoires masculin (m) pluriel (p)

### 5.4.6 Les informations sémantiques


La classe initiale est + info (informatique), actuellement, nous avons recensé 10 domaines codés “+xxx”, xxx étant l’abréviation du domaine.

#### 5.4.6.1 Systèmes d’information

"+sinf" avec :
- Systèmes client serveur : "+cliserv",
- Bases de données, "+dbase",

Exemples :
SGBD interne, SGBD externe, système de gestion de fichiers, base de données répartie, base de données relationnelle.
- Technologies objet : "+otech",

Exemples :
Programmation orientée objet, diagramme de classes, diagramme d'interactions.
- Génie logiciel : "+softg",

Exemples :
Cycle de vie du logiciel, diagramme de modélisation, cycle de développement itératif.

5.4.6.2 Compilation: "+comp"

Exemples :
Code intermédiaire, analyseur montant, analyse déterministe, analyse descendante.

5.4.6.3 Algorithmique : "+algo"

Exemples :
Déclaration des variables, programmation structurée, Programmation non structurée, ligne de commentaires.

5.4.6.4 Langages (programmation) : "+lang"

Exemples :
Langage C, langage de bas niveau, langage évolué, langage assembleur.

5.4.6.5 Architecture des ordinateurs : +arch"

Exemples :
Mémoire centrale, disque dur, unité arithmétique et logique.

5.4.6.6 Systèmes d'exploitation : "+expl"

- Systèmes temps partagé : "+tshare",

Exemples :
Quantum de temps, interruption de fin de quantum.
- Gros et moyens systèmes : "+lmsyst",
- Unix : "+unix",

**Exemples :**
Système d’exploitation multitâche, script shell, système de fichiers.

- Linux : "+linux"

**Exemples :**
Gestionnaire de fenêtres, environnement KDE, explorateur de fichiers, serveur Radius.

- Windows NT : "+wint",

**Exemples :**
Système d’exploitation 32 bits, système multitâche préemptif, système de fichiers NTFS.

- Windows XP : "+winxp",

**Exemples :**
Windows XP professionnel, Windows XP pour Tablet PC.

- Systèmes distribués : "+ Dists",

**Exemples :**
Algorithme distribué, réseau physique de machines, système de fichier distribué.

- Systèmes temps reel : "+ rts",

**Exemples :**
Système d’exploitation temps réel, temps réel strict.

**5.4.6.7 Réseaux et télécommunications : "+ rest"**

**Exemples :**
Paquet d’acquittement, protocole de routage, point d’accès vidéo texte.

**5.4.6.8 Internet and groupware : "+intn"**

- Programmation internet : "+ intp".

**Exemples :**
Page HTML, page web dynamique, programmation serveur avec PHP, langage ASP.net.

- Internet : "+int",

Exemples :
Protocole de communication IP, fournisseur d’accès à internet.

- groupware & messagerie : "+gmess",

Exemples :
Gestionnaire de tâche et d’agenda collaboratif, logiciel de travail collaboratif, groupware Suse OpenExchange.

- internet & entreprise : "+intf",

Exemples :
Réseaux d’entreprises, stratégie internet, paiement en ligne, gestion des paiements sécurisés.

5.4.6.9 Informatique appliquée : "+ iapl"

- graphisme : "+graphic",

Exemples :
Logiciel de retouche d’images, images de synthèse.

- multimédia : "+mult",

Exemples :
Environnement d’apprentissage multimédia, conception assistée par ordinateur.

- graphisme orienté objet : "+oog",

Exemple :
Programme d’interface graphique.

- Bureautique : "+off",

Exemples :
Visionneuse fichiers textes avec image, archivage de documents.

- Publication assistée par ordinateur (PAO) : "+dskpubl",

Exemples :
Document d’impression, mise en page publicitaire, modifier une publication, créer une publication.

5.4.6.10 Intelligence artificielle : "+iart"

- Systèmes experts : "+exps",

   Exemples :
   Moteur d’inférence, chaînage avant, chaînage arrière.

- Compréhension automatique du langage naturel : "+nlp",

   Exemples :
   Analyse automatique de phrases, grammaire de cas, analyse du discours.

- Apprentissage automatique : "+mlearn",

   Exemples :
   Apprentissage par analogie, apprentissage par essai erreurs.

- Théorie des jeux : "+gtheor",

   Exemples :
   Jeux coopératifs, jeux non coopératifs, jeu en forme stratégique.

- Résolution de problèmes : "+pbsolv",

   Exemples :
   Espace de recherche, classe NP, parcours en profondeur.

- Représentation des connaissances : "+krepr",

   Exemples :
   Graphes conceptuels, réseaux sémantiques, règles de production.

- Systèmes intelligents : "+isyst",

   Exemples :
   Vision artificielle, systèmes de robotique.

- Démonstration de théorèmes : "+thprov",

   Exemples :
   Raisonnement par induction, raisonnement par déduction, modus ponens.
Vision "+vision",

Exemple :
Reconnaissance des formes.

Par exemple, l’entrée "algorithme de gestion mémoire" peut contenir toutes les informations sémantiques : "+info", "+expl" et "+unix".

5.5 Variantes


Diverses variantes existent, les abréviations, les variantes orthographiques, les variantes terminologiques (vues au chapitre 4) et morphosyntaxiques. Nous nous intéressons pour nos dictionnaires aux abréviations de termes et aux abréviations d’organisations, de compagnies, etc., appelées sigles. Elles seront mises dans des dictionnaires à part.

5.5.1 Définitions

Abréviation, sigle et acronyme désignent une forme de mot, indépendant de sa nature. Il peut en efft s’agir de noms, mais aussi d’adjectifs, de verbes, ou de locutions.

5.5.1.1 Abréviations

Une abréviation est la réduction d'un mot ou d'un groupe de mots, à certaines de ses lettres. Non souhaitée dans un texte littéraire, on y recourt toutefois régulièrement dans les notes qui l'accompagnent, dans les documentations scientifiques, juridiques, commerciales, etc.

Exemple : MEGA pour mégaoctets.

L'abréviation peut se construire à partir de l'initiale du mot, écrite soit en minuscule, soit en majuscule suivie d'un point abréviatif. Ces abréviations sont traitées exactement comme des mots normaux : étymologie, prononciation (le cas échéant), définition, etc.

Selon (Daille B. 1994), nous avons les abréviations de termes et les abréviations de compagnies appelées sigles. L’abréviation d’un terme est obtenue en prenant la première lettre de chaque unité pleine du mot composé. Les abréviations dépendent non seulement du domaine technique mais plus encore du texte technique ou elles ont été introduites.

Les abréviations sont utilisées comme les unités lexicales dans le corpus et gardent les propriétés syntaxiques, sauf l’accord. Une abréviation peut être utilisée à la place du terme et elle est de longueur 1.

103
5.5.1.2 Acronymes

L'acronymie est l'abréviation d'un groupe de mots formée par la ou les premières lettres de ces mots dont le résultat, nommé acronyme, se prononce comme un mot normal, on parle aussi de lexicalisation. L'acronyme peut également être formé en abrégeant les mots par leur syllabe initiale comme Bénélux. Ils s'écrivent généralement en lettres capitales et suivent les mêmes règles que les sigles : pas de point abréviatif, pas de trait d'union, pas d'accent.

Il peut être synonyme de sigle lorsque les lettres sont épelées, comme « A.S.S.E.D.I.C. »); mais il se distingue de l'abréviation, qui est prononcée en long, comme etc.

Exemple :
UNICEF United Nations children's fund - Fonds des Nations Unies pour l'enfance,

Il s'écrit parfois en lettres minuscules, éventuellement avec capitale initiale :

Exemples :
Radar (Radio detection and ranging)
Laser (Light amplification by stimulated emission of radiation)
Cedex (Courrier d'entreprise à distribution exceptionnelle)

5.5.1.3 Sigles

Un sigle est un ensemble de lettres initiales formant un mot servant d'abréviation. Si un sigle peut se prononcer comme un mot ordinaire, comme Unesco, alors c'est aussi un acronyme. Dans les autres cas, on l'épelle : SNCF [ɛs.en.se.'ɛf].

Certains sigles sont aussi écrits tels qu'on les prononce, devenant ainsi des noms communs (et s'accordant donc en genre et en nombre), par exemple : une bédé, des bédés, un cédérom, des cédéroms. Les sigles sont invariables en français et ne prennent pas la marque du pluriel. Les sigles, français ou étrangers, s'écrivent en lettres capitales sans point abréviatif, ni espace, ni trait d'union entre les lettres.

Exemples :
HEC (École des hautes études commerciales)
IBM (International business machines)

5.5.2 Différences entre les trois notions

Il s'agit en fait de trois notions imbriquées :

Une abréviation est le raccourcissement d'un mot ou d'un groupe de mots.

Exemple :
etc. : et cætera.

Un sigle est l'abréviation d'une locution qui n’en garde que les initiales (en général).
Exemple :
JO jeux olympiques

Un acronyme est un sigle dont l’enchâinement des lettres se lit comme un mot simple (il est prononçable au lieu de devoir être épelé) ; parfois ces acronymes deviennent des mots ordinaires, processus qui commence par leur passage en minuscules.

Exemple :
ONU se lit /o.ny/ et non /o.en.y/.

### 5.6 Dictionnaires des sigles et abréviations d’informatique

Pour construire nos dictionnaires nous avons recensé manuellement les sigles et abréviations (mais pas les acronymes) sur Internet, dans des magazines d’informatique tels que "Le monde informatique" et dans des ouvrages spécialisés.

#### 5.6.1 Dictionnaire des sigles d’informatique

##### 5.6.1.1 Le corpus

L'INRIA\(^{33}\) (Institut national de recherche en informatique et en automatique) diffuse la liste des sigles relatifs aux nouvelles technologies. Des sigles sont en anglais d’autres sont en français. Une liste par ordre alphabétique est fournie.

Dans la figure 5.3, nous avons quelques entrées pour la lettre "A".

---

\(^{33}\) [http://www.inria.fr/publications/infoweb/sigles](http://www.inria.fr/publications/infoweb/sigles)
Liste des sigles
(par ordre alphabétique de sigle)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y

AAAI -- American Association for Artificial Intelligence
Adresse : 445 Burgess drive
Localisation : Menlo Park, California 94025-3442 USA
Url : <urlhttp://www.aaai.org>

AAAS -- American Association for Advancement of Science
AACC -- American Automatic Control Council
ADEBUG -- International Workshop on Automated and Algorithmic Debugging
AAECC -- International symposium on applied Algebra, Algebraic Algorithms and Error-Correcting Codes
AAES -- American Association of Engineering Societies
AAIM -- American Association of Industrial Management
AARP -- AppleTalk Address Resolution Protocol

Figure 5-3 : Liste des sigles INRIA.

Exemples :
- ADELIE, Association pour le Développement et l'Etude de Logiciel Interactif pour l'Enseignement ;
- ADEMIR, Association pour le Développement dans l'Enseignement de la Micro-Informatique et des Réseaux ;
- ADERSA, Association pour le Développement de l'Enseignement et de la Recherche en Systématique Appliquée.

5.6.1.2 Le dictionnaire des sigles d’informatique "sigles.dic"

Nous avons tout d’abord recensé dans le corpus cité au ¶5.6.1.1, les sigles en français puis nous avons codifié chaque entrée selon le format Nooj.

On notera que deux entrées par terme sont nécessaires, la première décrit le terme et la seconde fait le lien entre le sigle et le terme.
Exemple :

Association Canadienne des Communications entre l'Homme et l'Ordinateur, N+NX5+sigle+Inria06

ACCHO, Association Canadienne des Communications entre l'Homme et l'Ordinateur, N

La première entrée contient le terme composé "Association Canadienne des Communications entre l'Homme et l'Ordinateur", elle indique :

- nous avons un nom "N",
- c’est nom composé de 5 mots pleins "NX5",
- c’est un sigle "+sigle" comme information sémantique,
- la référence bibliographique "+Inria".

La deuxième entrée c’est le sigle, elle fait le lien entre le sigle "ACCHO" et le terme "Association Canadienne des Communications entre l'Homme et l'Ordinateur". Dans la figure 5.4, nous avons un extrait du dictionnaire des sigles d’informatique "sigles.dic", pour les entrées "A" :

![Image](sigles.dic.png)

Figure 5-4 : Un extrait du dictionnaire des sigles d’informatique "sigles.dic".
5.6.2 Dictionnaire des abréviations d’informatique

5.6.2.1 Recensement des abréviations

Les abréviations en anglais et en français sont listées dans Wikipédia, des ouvrages d’informatique, des articles dans divers journaux (Le Monde informatique, 01 informatique). Nous donnons en exemple, un extrait des abréviations de Wikipedia, dans la figure 5.5.

Figure 5-5 : Extrait de la liste des abréviations d’informatique Wikipedia, lettre "A".

Exemples :

- ACRONYM (acronyme) : Abbreviated Coded Rendition Of Name Yielding Meaning
- ADMX : ADMinistrative XML template file (Stratégies de groupe)
- ADO : ActiveX Data Object de Windows

34 http://fr.wikipedia.org/w/index.php?title=Abr%C3%A9viations_en_informatique_A&oldid=49652113
35 http://www.01net.com/
36 http://www.01net.com/rub/01net-pro/10538/01net-pro/01-informatique/
- ADPCM : Adaptive Differential Pulse Code Modulation,
- ADSI : Active Directory Service Interface
- ADSL : ligne d'abonné numérique à débit asymétrique (asynchronous digital subscriber line).

5.6.2.2 Le dictionnaire des abréviations d’informatique "abréviations.dic"

Dans la figure 5.6, nous avons un extrait du dictionnaire des abréviations d’informatique "abréviations.dic", pour les entrées "A" :

![Extrait du dictionnaire des abréviations d'informatique](image)

Nous avons tout d’abord recensé dans le corpus cité au ¶5.6.2.1, les abréviations en français puis nous avons codifié chaque entrée selon le format Nooj.
On notera que deux entrées par terme sont nécessaires, la première décrit le terme et la seconde fait le lien entre l’abréviation et le terme.
Exemple :

Soient les entrées :

Analogique-Analogique-Digital,N+Abrev+wiki
AAD,Analogique-Analogique-Digital,N

La première entrée contient le terme composé ‘Analogique-Analogique-Digital’, elle indique :
- nous avons un nom "N",
- c’est une abrévéiation "+Abrev" comme information sémantique,
- la référence bibliographique "+wiki" pour Wikipedia.

La deuxième entrée c’est l’abrévéation, elle fait le lien entre l’abrévéation "AAD" et le terme "Analogique-Analogique-Digital".

5.7 Conclusion

Nous avons construit trois dictionnaires des termes d’informatique :

- Le dictionnaire des mots composés "Info_comp.dic" qui contient actuellement 10250 termes ;

- Le dictionnaire des sigles d’informatique "sigles.dic" qui comprend 300 entrées ;

- Le dictionnaire des abréviations d’informatique ‘abréviations.dic’ qui contient 524 entrées.

Actuellement 30000 termes composés ont été recensés manuellement, ils seront codifiés et rajoutés au dictionnaire "info_comp", d’autres abréviations et sigles sont recensés et seront rajoutés dans les dictionnaires correspondants.

Les dictionnaires élaborés permettent d’effectuer l’analyse automatique de textes d’informatique avec NOOJ. Ils pourront être complétés en vue d’effectuer une traduction automatique des termes composés vers l’anglais par exemple.
Chapitre 6 :
Morphologie
6 Morphologie

6.1 Introduction

Contrairement à INTEX, NooJ peut effectuer la flexion des composés. Le module flexionnel est déclenché en ajoutant la propriété spéciale "+FLX" à l’entrée du dictionnaire de mots simples ou composés (Silberztein M. 2003). Dans ce chapitre nous présentons le modèle utilisé dans NooJ pour la flexion des mots simples et composés. Nous élaborons et présentons les descriptions flexionnelles des termes composés d’informatique retenus comme entrées du dictionnaire "Info-comp".

6.2 Exemples d’entrées de dictionnaire et paradigme "+FLX"

Les deux entrées lexicales suivantes :

- table, N + FLX = Table+Conc
- voiture, N + FLX = Table+Conc

se fléchissent de la même manière (elles prennent un "s" au pluriel). Par conséquent, elles sont toutes deux associées à la même classe flexionnelle, "Table ", indiquée dans la propriété "+FLX =". La classe Table est définie par l’expression suivante :

Table = <E> / singulier + s/ pluriel;

qui stipule que si l’on n’ajoute rien à l’entrée lexicale ("E" est la chaîne vide), on obtient la forme du singulier ("singulier"), et si l’on ajoute un "s" à la fin de l’entrée lexicale, on obtient la forme du pluriel ("pluriel"), ainsi :

Table → tables au féminin singulier ;
Voiture → voitures au féminin pluriel.

6.3 Opérateurs (commandes)

Le système de flexion de NooJ peut traiter à la fois la morphologie flexionnelle des mots simples et composés d’une manière unifiée.
Le moteur flexionnel de NooJ utilise des commandes ou opérateurs par défaut qui opèrent sur le suffixe des entrées lexicales, comme suit :

- Commandes génériques :

  - **<B>** : supprime le caractère précédent ;
  - **<D>** : duplique le caractère courant ;
  - **<E>** : chaîne vide ;
  - **<L>** : positionne le curseur à gauche de xx positions ;
  - **<N>** : aller à la fin du mot suivant ;
  - **<P>** : aller à la fin du mot précédent ;
  - **<R>** : positionne le curseur à droite ;
  - **<S>** : supprime le caractère courant ;

- Arguments pour les commandes **<B>**, **<L>**, **<N>**, **<P>**, **<R>**, **<S>** :
  
  - xx nombre : répéter xx fois ;
  - W : mot entier.

Exemples :

- **<R3>** positionne le curseur à droite de 3 positions ;
- **<LW>** indique aller au début du mot ;
- `cheval[<B> ux]` ➔ `chevaux`
  
  **<B>** supprime le caractère "I", on obtient "cheva", "ux" est rajouté à "cheva", on obtient "chevaux" ;
- **Avoir** [**<B5> ont]** ➔ **ont**
  
  **<B5>** supprime 5 caractères précédents on obtient " ", on rajoute " ont", on obtient " ont".
- **Lever** [**<L3><B> è <R2><S>nt]**] ➔ `lèvent`
  
  **<L3>** positionne le curseur à gauche de 3 positions, "lever", **<B>** supprime le caractère précédent "Ivent", rajoute è, "lèver", **<R2>** positionne le curseur 2 positions à droite "lèver", **<S>** supprime le caractère courant "ève" rajoute nt, on obtient "lèvent".
Opérateurs pour les langues spécifiques :

- \(<A>\) : enlever accent grave ;
- \(<Á>\) : ajouter accent aigu ;
- \(<À>\) : ajouter accent grave.

Outre la fourniture d'opérateurs prédéfinis, NooJ permet aux linguistes de modifier ces opérateurs ou d’ajouter leurs propres opérateurs pour chaque langue au moteur de flexion.

6.4 Morphologie flexionnelle

NooJ fournit deux outils équivalents pour décrire les paradigmes flexionnels : les graphes et les règles. Les descriptions sont compilées en FST (transducteurs d'états finis). Ce sont des fichiers ayant pour extension ".nof".

6.4.1 Description graphique des flexions

Le dictionnaire doit obligatoirement contenir au moins une ligne contenant la commande :
#use NomGrammaireflexionnelle.nof

Par exemple dans l’entrée :

voiture,N + FLX = Table+Conc

La propriété "+FLX" permet de préciser le paradigme de flexion, ici, "Table". Cela suppose qu’il existe dans la grammaire flexionnelle du dictionnaire, un graphe nommé "Table" (Figure 6.1) qui permet de fléchir "voiture" en "voitures". "Table" est le nom du paradigme et correspond aux graphes inclus dans les grammaires flexionnelles/dérivationnelles.

Figure 6-1 : Graphe du paradigme TABLE (Silberztein M. 2003).
En entrée du graphe on indique les suffixes à ajouter à l’entrée pour obtenir chaque forme fléchie. En sortie, on a les codes flexionnels correspondants.

Dans le graphe de la figure 6.1, en prenant le premier chemin, <E> est un suffixe vide, s est le code flexionnel du singulier, nous obtenons "table" au singulier.
Dans le deuxième chemin, "s" est le suffixe, "p" est le code flexionnel du pluriel, nous obtenons "tables".

![Figure 6-2 : Graphe du paradigme flexionnel "cousin" (Silberztein M. 2003).](image)

Le graphe de la figure 6.2 contient deux sous-graphes : "Genre" qui contient le suffixe du genre (ajouter un e pour le féminin) et "Nombre" qui donne le suffixe du nombre (ajouter un s pour le pluriel). Les sous graphes Genre et Nombre sont donnés ci-dessous dans la figure 6.3.

![Figure 6-3 : Sous Graphes "Genre" et "Nombre" (Silberztein M. 2003).](image)
6.4.2 Description textuelle des flexions

Le dictionnaire doit obligatoirement contenir au moins une ligne contenant la commande :

```
#use NomDescriptionflexionnelle.nof
```

Par exemple dans l’entrée :

```
Chaise, N + FLX = Table+Conc
```

La propriété "+FLX" permet de préciser le paradigme de flexion, ici, "Table". Cela suppose qu’il existe dans la description flexionnelle des entrées du dictionnaire une règle nommée "Table" qui permet de fléchir "chaise" en "chaises". La règle "Table" est donnée ci-dessous :

```
Table = <E> / singulier + s/ pluriel ;
```

Nous pouvons utiliser des sous-règles dans les paradigmes des règles. Soient les règles flexionnelles suivantes :

- Genre = <E>/m | e/f ;
- Nombre = <E>/s | s/p ;
- Table = <E>/f ;Nombre ;
- Cousin = :Genre ;Nombre.

Les deux règles Genre et Nombre sont auxiliaires (sous-règles) et sont partagées entre les deux paradigmes Table et Cousin.

Elles permettent d’obtenir :

- *table,* en appliquant la troisième règle, <E>/f indique qu’on ne rajoute rien "<E>" au féminin "f", ":Nombre" fait appel à la deuxième règle Nombre qui contient "<E>/ s", indiquant qu’on ne rajoute rien "<E>" au singulier "s";

- *tables* en appliquant la deuxième partie de la règle Nombre "s/p" qui indique ajouter "s" au pluriel "p";
- Cousin ;
- Cousine ;
- Cousins ;
- Cousines ; (en appliquant les règles et sous-règles définies).
6.5 Flexion des mots composés

6.5.1 Modes de flexion

Le mode de flexion est dépendant de la classe du composé. Dans (Silberztein M. 1990), le pluriel des noms composés est abordé pour les principales classes. Nous avons le genre et le nombre des noms composés dans le tableau 6.1 suivant :

<table>
<thead>
<tr>
<th>Classe</th>
<th>Genre et nombre des noms composés</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA, AN</td>
<td>Genre et nombre identiques à ceux du nom et de l’adjectif</td>
</tr>
<tr>
<td>NDN</td>
<td>Genre et nombre identiques à ceux du 1er constituant N1</td>
</tr>
<tr>
<td>PN</td>
<td>Genre et nombre identiques à ceux du 1er constituant</td>
</tr>
<tr>
<td>NN</td>
<td>1er nom au pluriel, les deux au pluriel</td>
</tr>
<tr>
<td></td>
<td>2ème nom au pluriel, aucun</td>
</tr>
<tr>
<td>VN</td>
<td>Genre et nombre indépendants de celui du nom constitutif</td>
</tr>
<tr>
<td></td>
<td>La plupart sont masculins</td>
</tr>
<tr>
<td>NAN</td>
<td>Genre et nombre du 1er constituant</td>
</tr>
</tbody>
</table>

Tableau 6-1 : Genre et nombre des noms composés (Silberztein M. 1990).

Dans les exemples suivants :

- Un aide social,
- Une aide sociale,
- Des aides social(e)s,

dans la première ligne, « un » indique le masculin singulier, "une", le féminin singulier, mais "des " est soit masculin pluriel ou féminin pluriel, aussi, le pluriel d’aide social est ambigu.

6.5.2 Cas où un élément est à fléchir

La flexion opère sur le premier composant. Ces unités sont fléchies de la même manière que les mots simples.

Exemples :

- Avion à réaction → avions à réaction ;
- Homme d’honneur → hommes d’honneur.
6.5.3 Cas où plusieurs éléments sont à fléchir

6.5.3.1 Il n’y a pas d’accord entre les composants fléchis

Exemple :
- Homme d’action ➔
  Hommes d’action, homme d’actions, hommes d’actions

6.5.3.2 Il y a accord en genre ou en nombre

Les règles de flexion :

Cousin = <E>/mas + sin | s/mas+plur | e/fem+sin | es/fem+plur;
CousinGermain = :Cousin <PW> :Cousin;

permettent d’obtenir :
cousin germain,   cousins germains,
cousine germaine, cousins germaines.

6.6 Morphologie dérivationnelle

Les grammaires flexionnelles (règles ou graphes) peuvent contenir des paradigmes dérivationnels.

"+DRV" est utilisé pour indiquer un paradigme dérivationnel (présenté à la section 3.5.3).

Soient la description dérivationnelle :

RE = re/V

et l’entrée du dictionnaire :

manger,V+tr+FLX=Manger+DRV=Re

Pour ce verbe transitif, le paradigme de flexion est "manger ", "+DRV=RE" permet de dériver "remanger" qui est un verbe.

6.7 Descriptions flexionnelles des mots composés d’informatique

Pratiquement absente des modèles de flexions pour les mots composés du DELAC, la description des flexions des noms composés d’informatique du dictionnaire "Info-comp" reste à faire, pour cela, nous étudions pour chaque entrée ses formes fléchies.
Après l’extraction des termes d’informatique, l’acquisition consiste à introduire le terme retenu, comme entrée, dans le dictionnaire. La flexion des composés dépend des constituants, nous avons examiné pour chaque entrée ses formes de flexions afin d’établir la règle de flexion qui correspond. Une règle de flexion peut être commune à plusieurs entrées.

Les entrées lexicales dans les dictionnaires NooJ peuvent être associées à un paradigme qui formalise leur flexion. Dans notre cas, les entrées de notre dictionnaire “Info-comp” contiennent un champ “+FLX” qui indique le nom du paradigme flexionnel pour l’entrée. Le but de la description de la flexion d’une entrée lexicale est de permettre de relier ensemble automatiquement toutes ses formes de mots.

Pour chaque nom composé nous indiquons s’il se met au singulier et/ou au pluriel, au féminin ou au masculin. Il faut aussi décrire comment le fléchir, en décrivant les modèles de flexions.

Pour permettre l’analyse automatique linguistique de textes d’informatique, toutes les flexions des composés doivent être décrites, en effet dans les textes, les composés peuvent apparaître sous toutes leurs formes possibles.

6.7.1 Elaboration des règles de flexions des mots composés du dictionnaire

Nous élaborons les règles de flexions des mots composés du dictionnaire "info_comp.dic" que nous avons construit.

Soient les termes au masculin singulier, “algorithme de gestion mémoire”, “programme de test machine” et “programme de gestion canal”, ils se fléchissent au masculin pluriel selon :

algorithme de gestion mémoire → algorithmes de gestion mémoire;

programme de test machine → programmes de test machine;

programme de gestion canal → programmes de gestion canal.

Ils ont la même catégorie syntaxique NPNPN, aussi, dans notre dictionnaire "Info-comp" dont un extrait est présenté dans la figure 6.4,
Figure 6-4 : Extrait du dictionnaire "Info_comp".

on voit que les entrées lexicales "algorithme de gestion mémoire", "programme de test machine" et "programme de gestion canal" :

algorithme de gestion mémoire,N+NP PN+info+FLX=AccorDeBase+OS
programme de test machine,N+NP PN+info+FLX=AccorDeBase+Hild98
programme de gestion canal,N+NP PN+info+FLX=AccorDeBase+Hild98

ont la même description flexionnelle définie par le paradigme "AccorDeBase" décrit par la règle textuelle :

\[\text{AccorDeBase} = \langle E \rangle/m+s + \langle PW \rangle s/m+p;\]

"\langle E \rangle/m+s" : stipule que si l'on n'ajoute rien à l'entrée lexicale ("E" est la chaîne vide), on obtient la forme du masculin singulier ("m+s").

"\langle PW \rangle s/m+p" : stipule que si l'on ajoute un "s" à la fin du mot précédent "PW", de l'entrée lexicale, on obtient la forme du masculin pluriel ("m+p").

Dans le cas de l'entrée "programme de gestion canal", nous avons les formes fléchies:

- m+s → programme de gestion canal;
- m+p → programmes de gestion canal.
Prenons le terme recensé, "accès principal au périphérique", il est au masculin singulier, examinons les pluriels suivants :

accès principal aux périphériques,
accès principaux au périphérique,
accès principaux aux périphériques.

Nous ne retenons que le pluriel "accès principaux aux périphériques", et on obtient la règle suivante :

\[ \text{AccesPauPeriph} = <E>/m+s + s<PW><N><B>ux<N>x/m+p; \]

6.7.2 Grammaire textuelle des mots composés d’informatique

Les règles textuelles décrivant les paradigmes précisés dans "+FLX" sont regroupées dans le fichier 'composes-flex.nof' dans la grammaire textuelle. Dans la figure 6.5, nous avons un extrait de la grammaire décrivant les règles de flexions des composés d’informatique.

Dans la figure 6.5, la première ligne contient :

\[ \text{AccesAccordé} = <E>/m+s + s/m+p; \]

Cette règle flexionnelle stipule que si l’on n’ajoute rien à l’entrée lexicale («E» est la chaîne vide), on obtient la forme du masculin singulier («m+s»), si l’on ajoute un «s» à la fin du mot de l’entrée lexicale, on obtient la forme du masculin pluriel («m+p»), on obtient les formes :
accès accordé et accès accordés.

Ainsi, la règle de flexion "AccesAccordé" concerne les entrées (masculin singulier) suivantes du dictionnaire et fournit leurs pluriels:

<table>
<thead>
<tr>
<th>Entrée au singulier</th>
<th>Pluriel</th>
</tr>
</thead>
<tbody>
<tr>
<td>accès accidentel</td>
<td>accès accidentels</td>
</tr>
<tr>
<td>accès accordé</td>
<td>accès accordés</td>
</tr>
<tr>
<td>accès aléatoire</td>
<td>accès aléatoires</td>
</tr>
<tr>
<td>accès aléatoire à la mémoire</td>
<td>accès aléatoires à la mémoire</td>
</tr>
<tr>
<td>accès anonyme</td>
<td>accès anonymes</td>
</tr>
<tr>
<td>accès arbitraire</td>
<td>accès arbitraires</td>
</tr>
<tr>
<td>accès arborescent</td>
<td>accès arborescents</td>
</tr>
<tr>
<td>accès complet</td>
<td>accès complets</td>
</tr>
<tr>
<td>accès concurrentiel</td>
<td>accès concurrentiels</td>
</tr>
<tr>
<td>accès conflictuel</td>
<td>accès conflictuels</td>
</tr>
<tr>
<td>accès cyclique</td>
<td>accès cycliques</td>
</tr>
</tbody>
</table>

D’autres entrées ont pour modèle de flexion "FLX" la règle "AccesAccordé", ici, nous n’avons présenté qu’une partie de ces entrées.

Dans la figure 6.5, la sixième ligne contient :

AdrAbsolu = <E>/m+s + s<P>s/m+p;

Cette règle flexionnelle stipule que si l’on n’ajoute rien à l’entrée lexicale ("E" est la chaîne vide), on obtient la forme du masculin singulier ("m+s"), si l’on ajoute un "s" au mot de fin de l’entrée lexicale, puis un "s" au mot précédent "P", on obtient la forme du masculin pluriel ("m+p"), on obtient les formes :

adressage absolu et adressages absolus.

Ainsi, la règle de flexion "AdrAbsolu" concerne les entrées (masculin singulier) suivantes du dictionnaire, et fournit leurs pluriels:

<table>
<thead>
<tr>
<th>Entrée au singulier</th>
<th>Pluriel</th>
</tr>
</thead>
<tbody>
<tr>
<td>adressage abrégé</td>
<td>adressages abrégés;</td>
</tr>
<tr>
<td>adressage absolu</td>
<td>adressages absolus;</td>
</tr>
<tr>
<td>adressage aiguilleur</td>
<td>adressages aiguilleurs;</td>
</tr>
<tr>
<td>adressage basé</td>
<td>adressages basés;</td>
</tr>
<tr>
<td>adressage direct</td>
<td>adressages directs;</td>
</tr>
<tr>
<td>adressage immédiat</td>
<td>adressages immédiats;</td>
</tr>
<tr>
<td>adressage implicite</td>
<td>adressages implicites;</td>
</tr>
<tr>
<td>adressage indexé</td>
<td>adressages indexés;</td>
</tr>
<tr>
<td>adressage indirect</td>
<td>adressages indirects;</td>
</tr>
<tr>
<td>adressage relatif</td>
<td>adressages relatifs.</td>
</tr>
</tbody>
</table>
6.7.3 Tableau des paradigmes flexionnels

Dans le tableau suivant nous indiquons quelques règles de formes fléchies que nous avons élaborées, avec des exemples de flexions de termes à l’aide des expressions flexionnelles correspondantes.

<table>
<thead>
<tr>
<th>Paradigme</th>
<th>Expression flexionnelle</th>
<th>Termes fléchis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accès Accordé</td>
<td>&lt;E&gt;/m+s + s/m+p</td>
<td>accès accordé, accès accordés</td>
</tr>
<tr>
<td>Accès Min</td>
<td>&lt;E&gt;/m+s + &lt;B&gt;ux/m+p</td>
<td>accès minimal, accès minimaux</td>
</tr>
<tr>
<td>Accès Direct à la Mémoire Évoluée</td>
<td>&lt;E&gt;/m+s + &lt;PW&gt;&lt;N&gt;s/m+p</td>
<td>Accès direct à la mémoire évolué, Accès directs à la mémoire évolués</td>
</tr>
<tr>
<td>Accords de Base</td>
<td>&lt;E&gt;/m+s + &lt;PW&gt;s/m+p</td>
<td>Accord de base, accords de base</td>
</tr>
<tr>
<td>AccèsPrincipal aux Périphériques</td>
<td>&lt;E&gt;/m+s + &lt;PW&gt;s/m+p</td>
<td>Accès principal aux périphériques, Accès principaux aux périphériques</td>
</tr>
<tr>
<td>Abonné Local</td>
<td>&lt;E&gt;/m+s + &lt;B&gt;ux&lt;PW&gt;s/m+p</td>
<td>Abonné local, abonnés locaux</td>
</tr>
<tr>
<td>Adressage Multiniveau</td>
<td>&lt;E&gt;/f+s + &lt;B&gt;ux&lt;PW&gt;s/f+p</td>
<td>Adressage multiniveau, Adressages multiniveaux</td>
</tr>
<tr>
<td>Affichage Virtuel en Cours</td>
<td>&lt;E&gt;/m+s + &lt;PW&gt;s/N&gt;s/m+p</td>
<td>Affichage virtuel en cours, Affichages virtuels en cours</td>
</tr>
</tbody>
</table>

Tableau 6-2 : Extrait explicatif du fichier "composes-flx.nof".

6.8 Conclusion

Les règles de flexions des mots composés ont été élaborées pour les entrées retenues du dictionnaire "info-comp". En recensant d’autres termes, nous examinerons si les règles déjà établies correspondent aux nouvelles entrées, sinon nous élaborerons les règles de flexions correspondantes. Grâce aux règles de flexions que nous avons établies, NooJ pourra reconnaître, lors de l’analyse automatique de textes, les termes composés d’informatique sous leurs différentes formes fléchies rencontrées dans les textes.
Chapitre 7 :
Grammaires locales
7 Grammaires locales

7.1 Introduction


En partant du point de départ, les graphes d’automates à états finis, reconnaissent les unités linguistiques qui se trouvent sur un chemin. Lorsqu’un chemin a été exploré et qu’on arrive au point final, la séquence a été reconnue (Gross M., 1997). Une séquence est reconnue par un automate si elle appartient au langage reconnu par l’automate. Les grammaires locales sont un mécanisme formel représentant, des formes linguistiques. Les combinaisons lexicales complexes d’expressions et leurs régularités nécessitent l’emploi des graphes, qui sont des automates finis, particulièrement adaptés aux expressions linguistiques (Gross M. 1989).

Les règles décrites au sein de ces grammaires permettent de regrouper l’ensemble des termes d’une même famille. En produisant des contraintes syntaxiques, les automates permettent de résoudre de nombreuses ambiguïtés lexicales et structurales.

7.2 Les automates finis

7.2.1 Automates finis et graphes

Les automates finis sont des outils (des définitions théoriques sur les outils formels de base sont présentées en annexe2) qui sont bien adaptés pour représenter les phénomènes observés à des niveaux variés de la description des langues naturelles. Les variations formelles des unités linguistiques (allant des mots aux phrases) peuvent être représentées (Gross M. 1987). Un automate fini est un graphe qui contient des nœuds et des flèches qui relient les nœuds.

Dans la notation utilisée dans (Silberztein M. 1993), les transitions sont représentées dans les nœuds, et les états implicites sont laissés afin d’alléger les graphes. Il y a un seul nœud initial et un seul final. Les nœuds sont étiquetés (sauf le nœud terminal) en leur associant le mot vide <E> ou un symbole. Sur le plan pratique, dans NooJ, les grammaires locales sont représentées par des graphes. Chaque graphe est représenté sous forme d’un ensemble de nœuds connectés et incluant un nœud initial et un nœud terminal.

Un graphe est un ensemble de chemins qui partent d’un état initial (la flèche la plus à gauche) et aboutissent à un état dit final (appelé état terminal, symbolisé par un cercle muni d’une croix). Les graphes se lisent de gauche à droite et chaque chemin comporte des boîtes.
Ces boîtes sont étiquetées soit par des caractères pour une analyse morphologique (figure 8.1 et figure 8.2), soit par des mots pour une analyse lexicale ou syntaxique (figure 8.3). Dans tous les graphes, un chemin représente une suite de mots qui est une expression de la grammaire.

A l’intérieur de ces graphes on trouve :

- Le nœud initial, il est représenté par une flèche droite ;
- Le nœud terminal, il est représenté par une croix dans un rond ;
- Les nœuds intermédiaires, ce sont des boîtes qui contiennent les étiquettes en entrée du transducteur.
- Des étiquettes de sortie éventuelles. On parle d’étiquetage, il sera abordé à la section 7.2.4.1. Une étiquette de sortie sous les nœuds sera utilisée pour associer des informations linguistiques aux séquences reconnues. Dans ce cas, les graphes fonctionnent comme transducteurs à états finis. Un transducteur est un graphe d’état fini qui associe des sorties aux chaînes reconnues (Gross M. 1997).

Les nœuds intermédiaires, sont des boîtes qui contiennent les étiquettes en entrée du transducteur. Ils peuvent contenir :

- Des symboles grammaticaux ;
- Des appels à des sous-graphes qui seront vus à la section 7.2.3.

Les symboles grammaticaux représentent, dans la majorité des cas, des mots au sens linguistique et sont donc très variés. Ces symboles sont utilisés afin de limiter le nombre de transitions dans les graphes.

Exemples :

- \(<N>\) : désigne n’importe quelle unité lexicale ayant été associée à la catégorie grammaticale N (noms simples ou composés) ;
- \(<V+2+m+s>\) : désigne n’importe quel verbe conjugué à la deuxième personne, masculin, singulier.

Notons que dans les graphes, les entrées lexicales sont représentées entre angles et représentent un lemme. Nous avons par exemple, \(<N>\) cité précédemment, et aussi \(<être>\) qui correspond à toutes les formes conjuguées du verbe être telles que “est”, “sont”, “était”, etc. Les catégories grammaticales sont également écrites entre angles, mais en majuscules, par exemple :

\(<ADV>, <N>, <V>, etc.\)

D’autres informations lexicales ou sémantiques peuvent y être ajoutées comme dans \(<N:s>\) qui correspond à des noms au singulier ou \(<N+hum>\) qui correspond à des noms humains. La notion \(<E>\) est utilisée pour la séquence vide.
### 7.2.2 Exemples

Soit le graphe suivant () représentant les variantes du mot tsar:

![Graphe de variantes orthographiques du mot tsar en français.](image)

**Figure 7-1 : Automate à états finis des variantes orthographiques du mot tsar en français.**

Ce graphe (Silberztein M. 1993) représente 4 variantes orthographiques possibles du mot tsar : csar, czar, tsar, tzar.

Dans le graphe de la figure 7.2 (transducteur à états finis), les 16 formes reconnues par le graphe : csar, czar, tsar, tzar, csarine, czarine, tsarine, tzarine, csars, czars, tsars, tzars, csarines, czarines, tsarines, tzarines, sont lemmatisées en ‘tsar’.

![Graphe de grammaire morphologique du mot tsar.](image)

**Figure 7-2: Grammaire morphologique ".Tsar.nom" du mot tsar** (Silberztein M.).

Nous trouvons les informations:

- "N+ Hum" : Nom humain ;
- "+s" : singulier ;
- "+m" : masculin ;
- "+p" : pluriel.
- "+ UNAMB" est utilisé pour que l’analyse s’arrête dès la rencontre d’"UNAMB" pour la chaîne rencontrée.

---

37 Graphe extrait avec NooJ V2.5 b0414 (open grammar -» lexical analysis)
Soit le mot "tsarines" à analyser par cette grammaire, on obtient en sortie :

<tsar, N+Hum+f+p>

- "tsar" est associé aux informations : "+N" pour indiquer que tsarines est un nom, "+f" : féminin, "+p" : pluriel, et "+UNAMB", non ambigu.

### 7.2.3 Graphes et RTN

Les RTN sont des automates dans lesquels on se donne la possibilité d’entrer des nœuds auxiliaires, c’est-à-dire des nœuds qui sont eux-mêmes des RTN.

Les graphes, ayant le formalisme des réseaux de transitions récursifs (RTN, Recursive Transition Networks), (Woods W. 1970), peuvent faire appel à des sous-graphes imbriqués.

Dans les graphes NooJ, les étiquettes non terminales colorées en jaune, représentent des appels à des sous-graphes de même nom (voir le graphe de la figure 7.3).

Par exemple, le graphe de la figure 7.3 est un automate qui décrit les déterminants numéraux de 100 à 999 en français :

![Figure 7-3 : Déterminants numéraux de 100 à 999 en français (Silberztein M.)](image)

Les appels aux sous-graphes sont assurés par le biais du symbole " : " (n’apparaissant pas sur les graphes) qui précède le nom du sous-graphe à appeler. Par exemple, le nœud qui apparait colorié (boîte jaune) dans la Figure 7.3, fait appel à un autre graphe dont l’étiquette est Dnum2-99, et qui décrit à son tour les déterminants numéraux de 2 à 99. La possibilité de référencer dans un graphe d’autres graphes permet d’éviter l’énumération des éléments qui
apparaissent de façon régulière et répétitive dans plusieurs automates. La description linguistique est allégée grâce au formalisme de RTN.

### 7.2.4 Utilisation des grammaires locales dans le TAL

Outre la possibilité de description des règles de flexion et de dérivation et d’analyse morphologique (introduites au chapitre 3 et développées au chapitre 6), les graphes de NooJ peuvent être utilisés dans d’autres applications liées au traitement automatique des langues (Mesfar S. 2008). Parmi ces applications, nous citons :

#### 7.2.4.1 Reconnaissance et étiquetage automatiques

Le système NooJ est un système de traitement automatique de la langue naturelle qui permet d’étiqueter les textes avec les dictionnaires électroniques existants et/ou créés selon les besoins et d’appliquer les grammaires locales aux textes.

Les grammaires locales lexicalisées permettent de reconnaître des séquences composées et d’étiqueter ces dernières de manière très satisfaisante.

Exemple :
Le graphe de la figure 7.4 représente les adverbes de temps tels que : au matin, à la nuit ou dans l’après-midi.

![Figure 7-4: Quelques adverbes de temps en français.](image)

L’étiquetage consiste à rajouter une étiquette de sortie sous les nœuds, pour associer des informations linguistiques aux séquences reconnues. Généralement, ces sorties jouent le rôle

---

38 Traitement automatique du langage naturel.
d’étiquette syntactico-sémantique pour la séquence identifiée. Sur le plan de la formalisation, cet étiquetage est effectué par le biais des informations de sortie écrites en gras sous les boîtes du graphe. En l’occurrence, les étiquettes écrites en-dessous des nœuds du graphe pour annoter chaque séquence reconnue. Toutefois, lorsqu’aucune étiquette n’est représentée en-dessous des nœuds du graphe, les sorties seront vides et les grammaires ne joueront plus le rôle d’étiqueteur.

Le graphe de la figure 7.4 peut servir à étiqueter les expressions qu’il reconnaît dans un texte comme adverbes de temps, à l’aide d’informations indiquées “<ADV+Temps>”. Ainsi, après avoir appliqué ce graphe, les phrases suivantes reconnues :

Luc est arrivé au matin à Lyon
Ils sont partis dans l’après-midi

sont étiquetées comme suit:

Luc est arrivé au matin <ADV+Temps> à Lyon
Ils sont partis dans l’après-midi <ADV+Temps>

7.2.4.2 Extraction de données

L’extraction d’information est l’un des domaines d’actualité, il s’agit notamment de l’extraction de noms propres et aussi de terminologie. De nombreuses études ont été menées sur le français pour extraire automatiquement des noms de personnalités en leur associant une fonction politique ou professionnelle à l’aide de grammaires sous forme de graphes (Senellart J. 1998) ou extraire les noms de gènes dans les corpus en génomique (Poibeau T. 2001). Nous nous intéressons à l’extraction de mots composés d’informatique, notre objectif étant la construction d’un dictionnaire de terminologie informatique.

7.2.4.3 Ambiguités

Une grammaire NooJ peut être utilisée pour lever les ambiguïtés. La levée d’ambiguïté est fondamentale pour le traitement automatique de textes. De nombreuses études pointues ont montré l’intérêt de l’utilisation des cascades de transducteurs pour la désambiguïsation (Silberztein M. 2005a).

En outre, les grammaires locales peuvent être utilisées pour le filtrage d’information, notamment les travaux sur la distribution des dépêches AFP39 (Balvet A. 2000, 2002). Elles peuvent aussi servir comme outil d’aide à la traduction automatique (Mesfar S. 2006).

---

39 AFP : Agence France Presse (site web : http://www.afp.com/).
7.3 Construction des Grammaires locales des mots composés d’informatique

Les grammaires construites sous forme de graphes sont faciles à lire et elles peuvent être appliquées directement sur des textes à l’aide de NooJ. Une grammaire NooJ permet de regrouper les termes par famille, peut contenir plusieurs graphes et être utilisée pour lever les ambiguïtés. (Silberztein M. 1993). Nous nous intéressons aux grammaires locales des termes d’informatique et nous allons, dans ce chapitre, présenter la méthode utilisée pour la construction des graphes.

7.3.1 Construction des graphes : une méthode empirique

La construction des graphes est entièrement empirique, le nombre de sous classes dépend de chaque mot (Gross M. 1999). Le nombre de graphes dépend du nombre de significations et des termes trouvés dans les dictionnaires et les textes. Il varie largement, aussi la décision d’inclure un terme dans un dictionnaire plutôt que dans une grammaire est difficile à prendre, elle est en partie basée sur le nombre de termes observés.

Dans (Gross M. 1999), on trouve une méthode empirique pour la construction des graphes autour d’un mot-clé, de manière équivalente autour d’une unité sémantique. Les graphes peuvent être utilisés directement pour l’analyse syntaxique, la reconnaissance de chaînes significatives. Chaque graphe contient un ensemble de chaînes décrites.

7.3.2 Les grammaires locales des termes d’informatique

7.3.2.1 Construction après recensement manuel (textes écrits)

Une liste des ouvrages, revues, journaux, etc. utilisés pour l’extraction manuelle des termes d’informatique est donnée en annexe 1.

Afin de présenter un exemple, une lecture de textes est faite dans le dictionnaire (Harrap, 1983). Nous faisons l’extraction des termes trouvés dans les pages 17 et 18.

Les termes trouvés sont :

<table>
<thead>
<tr>
<th>Carte SCII Telecom Express</th>
<th>Carte à mémoire flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOISA</td>
<td>Carte binaire par colonne</td>
</tr>
<tr>
<td>Carte son</td>
<td>Carte binaire par rangée</td>
</tr>
<tr>
<td>Carte vidéo</td>
<td>Carte chercheuse</td>
</tr>
<tr>
<td>Carte vierge</td>
<td>Carte de compte</td>
</tr>
<tr>
<td>Carte à volet</td>
<td>Carte de contrôle</td>
</tr>
<tr>
<td>Carte à bande</td>
<td>Carte de décompression</td>
</tr>
<tr>
<td>Carte à Carte</td>
<td>Carte de fin</td>
</tr>
<tr>
<td>Carte à disque</td>
<td>Carte de lancement</td>
</tr>
<tr>
<td>Carte à bords perforés</td>
<td>Carte de lancement de travail</td>
</tr>
<tr>
<td>Carte à encoches marginales</td>
<td>Carte de fin de travail</td>
</tr>
<tr>
<td>Carte à perforations marginales</td>
<td>Carte graphique</td>
</tr>
<tr>
<td>Carte à mémoire</td>
<td>Carte grille</td>
</tr>
</tbody>
</table>
A partir de ces termes nous construisons la grammaire locale "Carte’" (Figure 7.5). Ainsi, ce graphe représente la syntaxe des composés de la famille de mots "carte" à partir de la racine.

Figure 7-5 : Une grammaire locale pour le terme "Carte".
7.3.2.2 Construction après concordance (le texte est un fichier)

Soit le fichier texte "text1-chantal.not" ayant pour titre "Histoire du langage C++", On désire obtenir les concordances pour le "mot-clé" "Opérateur", et examiner le contexte du mot, afin d’extraire les termes composés puis construire la grammaire locale "Opérateur.nog".

Dans le système NooJ, on effectue les opérations suivantes :

- on ouvre le fichier contenant le texte "text1-chantal.not",
- on applique "Linguistic Analysis" (pour effectuer l’analyse lexicale puis syntaxique),
- puis ‘locate pattern’ pour localiser le terme "Opérateur" dans le texte.

Nous obtenons la concordance pour "Opérateur". Dans la fenêtre "Concordance for text1-chantal.not" de la figure 7.6, nous trouvons l’exécution par NooJ de ces opérations.

Figure 7-6 : Concordance dans le fichier "text1-chantal.not" pour le terme "Opérateur".
Dans la colonne du milieu, le terme ‘Opérateur’ est localisé entre cinq mots avant et cinq mots après. En examinant dans chaque ligne, le terme "Opérateur" et ses mots voisins dans la concordance du texte de la figure 7.6, nous retenons les termes ci-dessous énoncés et construisons la grammaire locale de la figure 7.7.

Opérateur d'addition  Opérateur de résolution
Opérateur d'affectation "=" Opérateur de résolution de portée
Opérateur de cast Opérateur de rotation de bit
Opérateur inversible SK Opérateur de soustraction
Opérateur bit-à-bit Opérateur d'incrémentation
Opérateur d'assignation Opérateur inverse SK
Opérateur de multiplication Opérateur logiques
Opérateur de calcul Opérateur sizeof()
Opérateur de comparaison Opérateur ternaire
Opérateur de division

Figure 7-7: Grammaire locale "Opérateur".

Le graphe regroupe les termes ayant pour concept "opérateur". Cette grammaire syntaxique pourra être utilisée pour l’analyse de textes.
7.4 Application d’une grammaire à un texte d’informatique

Une grammaire peut être appliquée à un texte
- soit manuellement, via

Text ➔ Locate puis concordance ➔ Annotate text

Nous avons par exemple, le cas d’annotation du texte "dic-lhomme.not" de la figure 8.8 suivante :

![Figure 7-8 : Annotation1 du texte "dic-lhomme.not".](image)

- soit automatiquement en sélectionnant la grammaire locale dans

Info ➔ Preferences ➔ Syntactic Analysis

Cette commande permet (en ayant choisi la langue “Fr” pour le français) d’afficher les ressources syntaxiques (grammaires locales) qui sont des fichiers “.nog” constitués d’ensembles structurés de graphes qui décrivent les paradigmes syntaxiques des termes composés d’informatique que nous avons retenus.
On choisit parmi ces ressources celle ou celles qui seront appliquées au texte. Dans la figure 7.9, les ressources syntaxiques intégrées “mémoire.nog”, “processeur.nog” etc., sont des graphes décrivant des termes d’informatique que nous avons recensés. Le graphe “programme.nog” est sélectionné.

![Figure 7-9 : Un extrait des ressources syntaxiques pour les textes d’informatique.](image)

Tous les termes du graphe seront annotés comme s’ils avaient été reconnus dans un dictionnaire. Le résultat de l’application “linguistic analysis” de ce graphe au texte “dic-lhomme.not” donne l’annotation du texte, dont un extrait est donné ci-dessous dans la figure 7.10:
7.5 Quelques grammaires locales des termes d’informatique

7.5.1 Recensement des "mots-clés" d’informatique

Nous pouvons constater que certains termes simples appelés ici “mots-clés” sont des concepts porteurs de sens et permettent de regrouper des termes composés dans les graphes. A partir de la lecture de documents d’informatique, nous avons dressé une liste de concepts. Dans un premier temps nous avons recensé les mots-clés suivants :
<table>
<thead>
<tr>
<th>Accès</th>
<th>Décodeur</th>
<th>Numération</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additionneur</td>
<td>Dispositif</td>
<td>Opération</td>
</tr>
<tr>
<td>Adressage</td>
<td>Disque</td>
<td>Ordinateur</td>
</tr>
<tr>
<td>Adresse</td>
<td>Donnée</td>
<td>Organigramme</td>
</tr>
<tr>
<td>Algorithme</td>
<td>Ecran</td>
<td>Page</td>
</tr>
<tr>
<td>Article</td>
<td>Editeur</td>
<td>Panneau</td>
</tr>
<tr>
<td>Assemblage</td>
<td>Enregistrement</td>
<td>Perforation</td>
</tr>
<tr>
<td>Assembleur</td>
<td>Ensemble</td>
<td>Périphérique</td>
</tr>
<tr>
<td>Automate</td>
<td>Entrée</td>
<td>Piste</td>
</tr>
<tr>
<td>Bande</td>
<td>Erreur</td>
<td>Point</td>
</tr>
<tr>
<td>Bascule</td>
<td>Exécution</td>
<td>Processeur</td>
</tr>
<tr>
<td>Base</td>
<td>Exploitation</td>
<td>Processus</td>
</tr>
<tr>
<td>Binaire</td>
<td>Fichier</td>
<td>Programmation</td>
</tr>
<tr>
<td>Bloc</td>
<td>Fonction</td>
<td>Programme</td>
</tr>
<tr>
<td>Branchement</td>
<td>Format</td>
<td>Protocole</td>
</tr>
<tr>
<td>Calcul</td>
<td>Gestion</td>
<td>Puce</td>
</tr>
<tr>
<td>Calculateur</td>
<td>Groupe</td>
<td>Recherche</td>
</tr>
<tr>
<td>Caractère</td>
<td>Horloge</td>
<td>Référence</td>
</tr>
<tr>
<td>Carte</td>
<td>Impression</td>
<td>Registre</td>
</tr>
<tr>
<td>Chargement</td>
<td>Imprimante</td>
<td>Réseau</td>
</tr>
<tr>
<td>Chargeur</td>
<td>Impulsion</td>
<td>Ressource</td>
</tr>
<tr>
<td>Circuit</td>
<td>Index</td>
<td>Segment</td>
</tr>
<tr>
<td>Clavier</td>
<td>Indicateur</td>
<td>Souris</td>
</tr>
<tr>
<td>Clé</td>
<td>Instruction</td>
<td>Sous-programme</td>
</tr>
<tr>
<td>Codage</td>
<td>Interruption</td>
<td>Superviseur</td>
</tr>
<tr>
<td>Code</td>
<td>Langage</td>
<td>Symbole</td>
</tr>
<tr>
<td>Codeur</td>
<td>Lecteur</td>
<td>Système</td>
</tr>
<tr>
<td>Codification</td>
<td>Lecture</td>
<td>Table</td>
</tr>
<tr>
<td>Commande</td>
<td>Logiciel</td>
<td>Tableau</td>
</tr>
<tr>
<td>Compilateur</td>
<td>Longueur</td>
<td>Tampon</td>
</tr>
<tr>
<td>Compilation</td>
<td>Marque</td>
<td>Temps</td>
</tr>
<tr>
<td>Complément</td>
<td>Matériel</td>
<td>Terminal</td>
</tr>
<tr>
<td>Conception</td>
<td>Mémoire</td>
<td>Traitement</td>
</tr>
<tr>
<td>Conversion</td>
<td>Message</td>
<td>Unité</td>
</tr>
<tr>
<td>Cryptage</td>
<td>Mode</td>
<td>Variable</td>
</tr>
<tr>
<td>Cycle</td>
<td>Mot</td>
<td>Voie</td>
</tr>
<tr>
<td>Décimal</td>
<td>Nombre</td>
<td>Zone</td>
</tr>
</tbody>
</table>

Ces concepts sont mis dans le dictionnaire intitulé "concepts_info" que nous avons construit, un extrait du dictionnaire se trouve dans la figure 7.11.
Figure 7-11 : Un extrait du dictionnaire des mots-clés "concepts_info".

Nous trouvons par exemple dans ce dictionnaire, les concepts "carte", "mémoire" et "registre", nous construisons les grammaires locales de "mémoire" et "registre", dans les sections 7.5.2 et 7.5.3.

Ces concepts pourront être utilisés, pour l’extraction automatique de mots composés à partir de textes à l’aide de patrons syntaxiques, ainsi que pour une indexation automatique de textes avec NooJ qui sera présentée au chapitre 8.

Lors de l’extraction manuelle, ces "mots-clés" permettent tout de suite d’associer le "mot-clé" qui est un terme, avec des mots voisins afin de créer des termes de longueur supérieure.

7.5.2 Grammaire locale "mémoire"

A partir de textes, nous effectuons une extraction manuelle de termes issus du concept "mémoire".40, nous désirons construire la grammaire locale des termes recensés suivants :

40 On notera que le terme "accès direct à la mémoire évolué" comprend le mot mémoire, ce terme est classé avec le terme "accès".
<table>
<thead>
<tr>
<th>Mémoire périphérique</th>
<th>Mémoire principale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mémoire à accès aléatoire</td>
<td>Mémoire à lecture non destructive</td>
</tr>
<tr>
<td>Mémoire à accès direct</td>
<td>Mémoire à lecture destructive</td>
</tr>
<tr>
<td>Mémoire à accès immédiat</td>
<td>Mémoire à liste inversée</td>
</tr>
<tr>
<td>Mémoire à accès rapide</td>
<td>Mémoire à mercure</td>
</tr>
<tr>
<td>Mémoire à accès sélectif</td>
<td>Mémoire à propagation</td>
</tr>
<tr>
<td>Mémoire à bulles magnétiques</td>
<td>Mémoire à tambour magnétique</td>
</tr>
<tr>
<td>Mémoire à commande manuelle</td>
<td>Mémoire à tores de ferrites</td>
</tr>
<tr>
<td>Mémoire à feuillets magnétiques</td>
<td>Mémoire à tube cathodique</td>
</tr>
<tr>
<td>Mémoire à liste directe</td>
<td>Mémoire à verrouillage</td>
</tr>
<tr>
<td>Mémoire active</td>
<td>Mémoire adressable par le contenu</td>
</tr>
<tr>
<td>Mémoire adressable</td>
<td>Mémoire volatile</td>
</tr>
<tr>
<td>Mémoire associative</td>
<td>Mémoire asynchrone</td>
</tr>
<tr>
<td>Mémoire auxiliaire de page</td>
<td>Mémoire auxiliaire</td>
</tr>
<tr>
<td>Mémoire bloc-notes</td>
<td>Mémoire cache</td>
</tr>
<tr>
<td>Mémoire centrale</td>
<td>Mémoire circulante</td>
</tr>
<tr>
<td>Mémoire cryogénique</td>
<td>Mémoire cyclique</td>
</tr>
<tr>
<td>Mémoire de manœuvres</td>
<td>Mémoire de masse</td>
</tr>
<tr>
<td>Mémoire de sauvegarde</td>
<td>Mémoire de travail</td>
</tr>
<tr>
<td>Mémoire dynamique</td>
<td>Mémoire électrostatique</td>
</tr>
<tr>
<td>Mémoire externe</td>
<td>Mémoire holographique</td>
</tr>
<tr>
<td>Mémoire ineffaçable</td>
<td>Mémoire interne</td>
</tr>
<tr>
<td>Mémoire magnétique</td>
<td>Mémoire morte</td>
</tr>
<tr>
<td>Mémoire morte programmable</td>
<td>Mémoire morte reprogrammable</td>
</tr>
<tr>
<td>Mémoire optique</td>
<td>Mémoire partagée</td>
</tr>
<tr>
<td>Mémoire programmable</td>
<td>Mémoire réelle</td>
</tr>
<tr>
<td>Mémoire rémanente</td>
<td>Mémoire reprogrammable</td>
</tr>
<tr>
<td>Mémoire secondaire</td>
<td>Mémoire segmentée</td>
</tr>
<tr>
<td>Mémoire séquentielle</td>
<td>Mémoire statique</td>
</tr>
<tr>
<td>Mémoire tampon</td>
<td>Mémoire topographique</td>
</tr>
<tr>
<td>Mémoire virtuelle</td>
<td>Mémoire vive</td>
</tr>
</tbody>
</table>

Nous obtenons la grammaire locale de la figure 7.12.
Figure 7-12 : Grammaire locale du terme "mémoire".

7.5.3 Grammaire locale "Registre"

Nous avons extrait manuellement, à partir d’ouvrages (annexe 1), les termes suivants :

Registre à décalage
Registre adressable
Registre arithmétique
Registre associatif
Registre banalisé
Registre buffer mémoire
Registre d’adresse mémoire
Registre d’échange avec la mémoire

Registre d’adressage
Registre d’adresse
Registre de base
Registre de sélection mémoire
Registre de translation
Registre d’index
Registre double
Registre général
Registre instruction
Registre intégré de configuration
Registre limite

Le concept est "registre", il permet le regroupement des termes dans le graphe suivant:

![Figure 7-13 : Grammaire locale du terme "Registre".](registres.png)

7.6 Conclusion

Les grammaires locales sont des automates à états finis représentés par des graphes, le format retenu est celui de Silberztein. Les graphes sont un outil très puissant de description des règles de flexions, de dérivation en morphologie, ils peuvent aussi être utilisés dans diverses applications telles que :

- la reconnaissance et l’étiquetage automatique,
- la représentation des mots composés,
- la résolution d’ambiguïtés,
- l’extraction de terminologie,

Nous avons construit diverses grammaires locales syntaxiques pour la reconnaissance des termes composés d’informatique. Les termes sont regroupés par famille autour des concepts, définis (en partie) à la section 7.5.1. Des grammaires locales ont été construites pour l’extraction automatique de terminologie, elles seront vues dans le chapitre suivant.
Chapitre 8 :
Analyse automatique de textes
8 Analyse automatique de textes

8.1 Introduction

Une réflexion sur la constitution de corpus spécialisés est un élément nécessaire dans tout projet de construction de dictionnaire terminologique. Le corpus doit vérifier trois types de conditions : signification, acceptabilité et exploitabilité en plus de la pertinence par rapport à un objectif d’analyse. L’ensemble de ces conditions est nécessaire pour sa réutilisabilité. De même, la production de ces corpus doit respecter la règle d’homogénéité : les documents retenus doivent être homogènes, c’est-à-dire obéir à des critères de choix précis et ne pas présenter trop de singularité en dehors de ces critères de choix.

Nous pouvons analyser des textes spécialisés d’informatique, avec le système NooJ, car nous avons construit les ressources linguistiques indispensables (dictionnaires électroniques, règles de flexions et grammaires locales des termes d’informatique), présentées dans les chapitres 5, 6 et 7.

Dans ce chapitre nous élaborons, tout d’abord, des corpus d’informatique. A partir des corpus, nous analysons des textes, et, grâce aux annotations des textes, nous pouvons compléter, si nécessaire, nos dictionnaires et grammaires locales afin d’assurer une bonne couverture lexicale.

Nous pouvons faire de l’extraction automatique de terminologie avec NooJ, en utilisant les des patrons syntaxiques que nous avons développés et exposons dans ce chapitre.

8.2 Annotation automatique de textes d’informatique

8.2.1 Analyse textuelle et annotation automatique

Après la phase de formalisation du lexique et de développement de grammaires morphologiques à large couverture, nous confrontons nos ressources linguistiques à des textes d’informatique. Cette confrontation nous permet de réviser, corriger et compléter toutes nos ressources. En effet, lors de l’application des ressources linguistiques à un corpus, NooJ fournit deux listes : celle des mots reconnus nommée « Annotations » et celle des mots non reconnus désignée par "Unknowns". Les causes des mots non reconnus peuvent être diverses :

- Les erreurs d’orthographe ;

- Les mots corrects mais non recensés dans les dictionnaires, dans la section 8.5, les "Unknowns" seront utilisés pour l’acquisition de terminologie.

Par ailleurs, après toute analyse morpho-lexicale d’un texte, chacun des mots reconnus est associé aux informations qui s’y rattachent indépendamment de son contexte. Ces informations lexicales, affichés sur la liste des "Annotations", sont attribuées de deux façons différentes :
- le programme d’identification de formes simples et composées par consultation de la liste des formes fléchies possibles rattachées aux entrées du dictionnaire,

- la routine d’application de grammaires morphologiques.

Dans cette section, nous nous intéressons à l’analyse linguistique d’une phrase et nous fournissons la liste des annotations contenant toutes les analyses qui auront été associées aux différentes formes reconnues par consultation de dictionnaires ou par tokenisation à l’aide de grammaires morphologiques.

Après avoir défini les flexions, les grammaires, les entrées des dictionnaires, et introduit les préférences pour l’analyse syntaxique (liste de grammaires locales utilisées par NOOJ), et l’analyse lexicale (liste de dictionnaires utilisés), nous pouvons analyser des textes d’informatique.

Dans l’exemple détaillé de la figure 8.1, soit la phrase :

Les algorithmes de gestion mémoire sont programmés.

Cette phrase contient :

- Des formes simples : le déterminant "les", le verbe "sont " et un adjectif "programmés";

- Un mot composé : "algorithmes de gestion mémoire".

Sur le plan pratique, chaque annotation attribuée à un morphème, à un mot simple ou composé est représentée par une flèche (voir figure 8.1). Toute flèche est étiquetée de façon à afficher le lemme, la catégorie grammaticale ainsi que toutes les informations flexionnelles, morphologiques et syntactico-sémantiques qui sont rattachées aux différents constituants des phrases. Nous notons que, par rapport à l’étiquetage de cette phrase, nous rencontrons :
Le mot "les" qui est ambigu, il est reconnu comme étant :

- un déterminant : "DET" qui est "le" au pluriel, avec "les,le,DET+z1+m+p" au masculin pluriel et "les,le,DET+z1+f+p" au féminin pluriel,
- un pronom : "PRO", avec "les,le,PRO+z1+3+m+p" au masculin pluriel et "les,le,PRO+z1+3+f+p" au féminin pluriel ;
- Un verbe : sont, "V", conjugué au présent 3ème personne du pluriel, avec "sont,être,V+z1+PR+3+p" ;
- Un mot composé : "algorithmes de gestion mémoire", il se présente comme alternative à la séquence de trois mots simples pleins. Nous avons :
  "algorithmes de gestion mémoire,algorithme de gestion mémoire,N+NPNPN+info+OS+ALGO+m+p", au masculin pluriel ;
- Le mot "programmé" qui est ambigu, il est reconnu comme étant :
  - Un participe passé : "programmé", avec "Programmer,V+z1+PP+m+p", au masculin pluriel,
  - Un adjectif : "programmé", avec "A,z1+m+p".

La même liste d'annotations est fournie sous forme d'une liste facilement exploitable, comme le montre la figure 8.2.
Figure 8-2 : Liste des annotations regroupées dans le dictionnaire "annotations_phrase.dic".

Ces annotations peuvent être utilisées dans de nombreuses applications du traitement de la langue :

- Elles peuvent être fournies comme entrées d’un analyseur syntaxique ;
- Elles peuvent être fournies comme entrées à un correcteur orthographique ;
- Elles peuvent servir pour l’évaluation de ressources lexicales, les mots non reconnus sont utilisés ;
- Elles peuvent être utilisées dans l’enseignement des langues en fournissant les analyses potentielles à l’apprenant ;
Elles peuvent être intégrées dans des requêtes, sous forme de symboles grammaticaux dans une application de concordance, ainsi qu’un concordancier en ligne (Mesfar S. 2008).

Notons qu’aux annotations morpho-lexicales ainsi obtenues, peuvent s’ajouter des annotations syntaxico-sémantiques attribuées par confrontation du texte à des grammaires locales.

### 8.2.2 Recherches linguistiques et concordances

A ce niveau du traitement, tous les mots reconnus par l’ensemble des ressources lexicales et morphologiques sont associés à un ensemble d’informations linguistiques. Les informations sont attribuées directement à partir d’une consultation du dictionnaire électronique, d’une tokenisation par le biais de grammaires morphologiques. Dès lors, il est possible d’exploiter ces informations pour explorer les phénomènes linguistiques locaux au moyen d’expressions rationnelles ou de graphes.

Outre la recherche des formes fléchies telles qu’elles figurent dans les textes, nous pouvons utiliser les informations morpho-syntaxiques recensées dans le dictionnaire. Celles-ci sont représentées par le biais de symboles grammaticaux ou flexionnels écrits entre des crochets (" < " et " > ").

Par ailleurs, nous utilisons la description formelle des entrées lexicales introduites dans le dictionnaire afin de construire nos symboles grammaticaux :

- Les formes fléchies identifiables avec l’écriture de leurs formes de base ou lemmes entre crochets.
- Les formes associées à une catégorie grammaticale donnée.

En ce qui concerne les formes fléchies identifiables, nous avons, par exemple le symbole `< programmer>` qui représente toutes les formes conjuguées du verbe programmer ainsi que celles qui y sont dérivées "programmeur", "programme", etc. Dans la figure 8.4, nous avons une concordance pour le symbole `< programmer>".
En ce qui concerne les formes associées à une catégorie grammatique donnée, elles peuvent être toutes reconnues par simple écriture de ce code morpho-syntaxique entre crochets. Par exemple, le symbole <V> représente toutes les formes verbales dans le texte ou le corpus, le symbole <ADV> symbolise le sous-dictionnaire des adverbes, etc.

En fait, tous les codes qui apparaissent dans le dictionnaire peuvent être utilisés dans de telles requêtes. Par ailleurs, le programme de recherche n'attache aucune signification aux codes utilisés : une forme ne sera reconnue comme <ADV> que si elle fait partie des mots ayant été préalablement associées au code morpho-syntaxique "ADV" dans une entrée du dictionnaire électronique ou à partir d’une annotation.

Les mêmes symboles peuvent aussi être utilisés pour représenter des séquences grammaticales. Par exemple, les séquences constituées d’un nom suivi d’un adjectif peuvent être représentées dans notre système par l’expression rationnelle : <N><A>.

Dans la figure 8.4, une recherche linguistique faite avec l’expression <N><A>, nous donne la table de concordance avec 100 items.
Les fenêtres de concordances (voir figure 8.5), que nous utilisons pour montrer les résultats des requêtes décrites, offrent plusieurs possibilités de traitement. Nous pouvons citer les possibilités de :

- nettoyer les concordances afin d’exclure le bruit des résultats affichés ;
- ajouter des annotations au texte. Cette fonctionnalité permet une annotation interactive du texte ou du corpus en cours de traitement ;
- colorier les séquences retenues dans le texte ;
- exporter les concordances en XML ;
- exporter les index des formes ou expressions retenues ;
- construire un rapport statistique des concordances.

Figure 8-4 : Fenêtres de recherche linguistique "Locate <N><A>" et table de concordance de la requête.
8.3 Constitution du corpus

8.3.1 Caractéristiques générales du corpus étudié

Nous avons effectué la construction d’un corpus spécialisé d’informatique sous format électronique afin dans un premier temps d’extraire automatiquement de nouveaux termes et enrichir le dictionnaire des termes composés d’informatique "Info-comp" que nous avons construit. Nous avons aussi cherché, sans succès, des corpus d’informatique d’usage libre, sous forme électronique.

Le corpus est constitué de fichiers textes issus de divers fichiers d’informatique au format "HTML", "pdf", "ps", "doc" ou "txt":

- articles de recherche,
- manuels techniques,
- thèses,
- magazines (01 informatique, Le monde informatique, etc.).

Les fichiers ont été transformés en format texte ".txt" puis importés dans NooJ41, pour obtenir des fichiers dans le format ".not" que NooJ peut traiter. Lors de la création d’un nouveau corpus (file → new corpus), NooJ crée un corpus vide dans lequel on peut importer un ensemble de fichiers textes.

Les fichiers constituant le corpus partagent les mêmes paramètres, habituellement le même langage, la structure et le codage. Les fichiers de corpus NooJ sont stockés avec leur information structurelle, indices variés, annotations linguistiques dans chaque “T.A.S” (Text Annotation Structure).

41 Dans la version actuelle de NooJ, il est possible d’importer, dans le corpus, des textes de types différents.
La taille du corpus d’informatique, constitué est de 33,5 Mo et comprend 1066 fichiers textes au format ".not".

![Figure 8-6 : Les fichiers constituant le corpus d'informatique constitué.](image)

En lançant "Linguistic analysis" dans Nooj, L’analyse lexicale est effectuée en 1314,1 secondes pour les 1066 fichiers du corpus de taille 33,5 Mo. Nous trouvons dans la figure 8.7, le temps mis pour l’analyse linguistique du corpus "corpus-info1.noc".
Figure 8-7: Analyse linguistique du corpus d’informatique.

Nous avons constitué un deuxième corpus plus court de quatre textes (figure 8.8), trois seront retenus pour l’extraction automatique, à la section 8.5.2.2.

Figure 8-8 : Les fichiers constituant le corpus2 d’informatique "textes-chantal.noc".
8.3.2 Liste des formes dans le corpus

Avant toute application de ressources linguistiques, nous fournissons quelques informations générales sur le corpus construit. Outre la liste de caractères et de digrammes que nous pouvons afficher (figure 8.8), nous pouvons aussi afficher la liste de toutes les formes (tokens) accompagnées par leurs fréquences d’apparition.

Dans la figure 8.9 ci dessous, nous avons sous forme de table, la liste des formes trouvées dans le corpus2 de la figure 8.8.

![Figure 8-9 : liste des formes dans le corpus.](image)

Cette table réalise différentes fonctionnalités:
- Trier les tokens (mots du texte) selon l’ordre alphabétique ;
- Trier suivant la fréquence d’apparition des tokens ;
- Exporter la liste totale ou partielle des tokens ;
- Afficher la table de concordances d’un ou plusieurs tokens au choix.

En l’occurrence, cette table montre que le token sélectionné "disque" apparaît 11 fois dans notre corpus. Sa sélection permet d’afficher (dans la figure 8.10), sous forme d’une table de concordances toutes les occurrences de ce token entourées de leurs contextes droit et gauche selon le modèle KWIC (Key Word In Context).

### 8.3.3 Table des concordances du token sélectionné

Dans la table 8.10, nous trouvons, les concordances pour le token "disques" trouvé dans la liste des formes trouvées dans le corpus.

![Table des concordances du token sélectionné](image)

**Figure 8-10 : table de concordances du token sélectionné "disques".**

On notera que les concordances sont des listes de courts extraits du texte. Chaque extrait est appelé concordance et montre une partie du texte avant et après un mot particulier.
Par exemple, la concordance du mot "maison" dans un texte donné, est une liste de courts extraits contenant le mot "maison" dans le texte. Avec chaque concordance est indiquée la référence du passage où ce mot a été trouvé.

Dans notre cas, La première colonne de la table de la figure 8.10, fournit le nom du texte source. Nous notons qu’un double clic sur l’une des lignes de cette table permet d’afficher l’occurrence en surbrillance directement à l’intérieur de son texte source. En l’occurrence, un double-clic sur la première ligne provoque l’affichage de la première apparition du token "disques" dans son texte source comme affiché sur la figure 8.11.

### 8.3.4 Affichage du token sélectionné dans le texte source

Dans la figure suivante, nous avons le token "disque", sélectionné dans la première ligne de la table des concordances de la figure 8.10. Le token est affiché dans son texte d’origine "chantal-3.not", donc d’où il a été extrait :

![Figure 8-11 : Affichage, dans son texte source, du token sélectionné "disque".](image)

### 8.4 Evaluation et couverture lexicale

#### 8.4.1 Analyse linguistique de corpus

Le corpus3 suivant (figure 8.12), contenant trois textes est créé et une analyse linguistique est effectuée.
Figure 8-12 : Les fichiers du corpus traité et temps d’exécution de l’analyse linguistique.

L’analyse linguistique a duré 20,4 secondes, les trois textes du corpus ont été analysés et nous pouvons lister les 5887 tokens différents avec leur fréquence d’apparition, les "unknowns", les mots ambigus, etc. Dans le tableau 8.1, nous indiquons les résultats trouvés.

<table>
<thead>
<tr>
<th>Elément du corpus</th>
<th>Nombre total trouvé</th>
<th>Nombre d’éléments différents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>217 308</td>
<td>5887</td>
</tr>
<tr>
<td>Unknowns</td>
<td>855</td>
<td></td>
</tr>
<tr>
<td>Annotations</td>
<td>123 945</td>
<td>10 367</td>
</tr>
</tbody>
</table>

Tableau 8-1 : Résultats trouvés.

8.4.2 Résultats

Les résultats sont basés sur ce corpus. Dans le tableau ci-dessous, nous montrons les résultats des expérimentations, liées à la couverture lexicale, obtenus à l’issue de l’étape de tokenisation et application des dictionnaires électroniques.

Le nombre de mots simples reconnus est égal à :

\[(\text{nombre de tokens} - \text{nombre d’UNKNOWNs}) = 5887 - 855 = 5032.\]

Le nombre de formes composées reconnues et non reconnues est donné dans le tableau 8.2.
<table>
<thead>
<tr>
<th>Nombre de formes composées reconnues</th>
<th>Nombre de formes composées non reconnues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nombre %</td>
<td>Nombre %</td>
</tr>
<tr>
<td>221 58,31</td>
<td>158 41,69</td>
</tr>
</tbody>
</table>

Tableau 8-2 : résultats obtenus.

Nous trouvons ci-dessous un extrait du dictionnaire des UNKNOWNS obtenu après l’analyse linguistique :

- AWK,UNKNOWN
- AX,UNKNOWN
- backplane,UNKNOWN
- basiquement,UNKNOWN
- bdevsw,UNKNOWN
- bca,UNKNOWN
- BE,UNKNOWN
- BCP,UNKNOWN

La liste des « Unknowns » est un bon indice de déficience des ressources développées. Ses entrées permettent aussi bien la mise à jour des dictionnaires (par l’ajout de nouvelles entrées lexicales ou par la description de nouvelles règles de déclinaison ou de dérivation), que l’enrichissement des grammaires (par écriture de nouvelles règles de tokenisation ou de nouveaux types de contraintes lexicales).

La liste des « Unknowns » comporte :

- Des noms corrects mais non recensés dans les dictionnaires ;
- Des mots anglais introduits dans le vocabulaire français d’informatique ;
- Des erreurs d’orthographe : erreurs typographiques, avec entre autres, l’omission de l’espace entre les mots, l’omission d’une lettre, erreur d’accent, omission d’accent ;
- Des abréviations.

8.5 Extraction automatique et acquisition de termes composés

8.5.1 Cas de la coordination : acquisition de nouveaux termes

8.5.1.1 Différents cas de coordination

Dans les textes nous trouvons différents cas de coordination d’où nous pouvons extraire automatiquement plusieurs termes. Dans les travaux de Catherine Domingues (Domingues C. 2001), sur la coordination et liés à la recherche d’informations dans un corpus documentaire, nous retenons les différents cas de coordination suivants :
- "N1 et N2 Mod" qui permet d’obtenir "N1 et (N2 Mod)", par exemple Électronique et calculateur électronique ;

- "N1 et N2 Mod" qui permet d’obtenir "(N1 Mod) et (N2 Mod)", par exemple Sauvegarde et restauration automatiques ;

- "N1 Mod1 et Mod2 " qui permet d’obtenir "(N1 Mod1) et (N1 Mod2) ", par exemple Mémoire de masse et auxiliaire ;

- "N1 Mod1 et MOT*" équivalent à "(N1 Mod1) et (MOT*)", analysé comme "N1 Mod1 et MOT*", équivalent à ",(N1 Mod1) et (N2 Mod2) ", par exemple Diagramme de temporisation et opération d’écriture ;

- " N1 et N2 mod1 et Mod2" équivalent à "(N1 mod1) et (N1 Mod2) et (N2 Mod1) et (N2 Mod2) ", par exemple Clavier et souris sans fil et ergonomiques ;

8.5.1.2 Grammaires locales pour la coordination et extraction de termes

Dans la figure 8.13, nous avons une grammaire locale pour l’extraction de terminologie, liée à la coordination (cas : "N1 Mod1 et Mod2").

Après application de cette grammaire locale (utilisant des variables et des outputs), au texte "mémoire de masse et auxiliaire", nous obtenons en figure 8.14, les sorties ("outputs "), ou termes extraits suivants :

Figure 8-13 : Patron syntaxique pour "N1 Mod1 et Mod2 ".

42 N1 et N2 sont des Noms, "Mod" est un modifieur qui peut être un Nom, un adjectif, un participe passé, etc.
- mémoire de masse,
- mémoire auxiliaire,

figurant dans les "output" de la figure 8.14 :

![Image](image_url)

**Figure 8-14 : Termes extraits ("outputs") pour "mémoire de masse et auxiliaire".**

Nous pouvons constater que le patron syntaxique créé, permet dans le cas de la coordination, d’extraire de nouveaux termes d’informatique.

Dans la figure 8.15, nous trouvons, une grammaire locale, établie pour le cas "(N1 et N2 Mod)".

![Image](image_url)

**Figure 8-15 : Grammaire locale pour "(N1 et N2 Mod)".**

En appliquant cette grammaire locale, au texte « programmes et routines en assembleur », nous obtenons en sortie les termes (figure 8.16) :
- Programme en assembleur ;
- Routines en assembleur.

Nous avons donc extrait automatiquement deux termes.

8.5.2 Extraction de terminologie avec NooJ

8.5.2.1 Extraction à partir d’un texte

Un patron est une expression NooJ ou une grammaire locale. Nous pouvons localiser tout patron dans le texte. Nous utilisons la commande "locate a pattern" pour extraire des noms composés à partir de textes ou de corpus. Par exemple (Aoughlis F. 2007), dans la Figure 8.17, nous trouvons les étapes suivantes :

- une analyse linguistique du texte "uc.not" est faite avec NooJ,
- l’option "locate a pattern" est utilisée pour trouver les termes qui ont la catégorie syntaxique <N><A> ("Nom Adjectif") dans le texte (termes candidats).

Une liste des termes (concordances) trouvés est donnée, il y en a ici 472.
Après l’extraction automatique, l’acquisition de terminologie se fait comme suit :

- A partir de la concordance pour le texte, le linguiste ou terminologue, sélectionne les termes parmi les 472 "candidats-termes", et ne retient que ceux qui sont des entrées pour le dictionnaire d’informatique "Info_comp", nous avons retenu 156 termes ;

- Le terminologue ajoute manuellement (acquisition) les termes dans le dictionnaire, après les avoir codifiés, avec le format donné à la section 5.4.5.

Nous avons fait en parallèle une extraction manuelle et avons extrait 156 termes de catégorie syntaxique "<N><A>". Nous avons les résultats suivants :

<table>
<thead>
<tr>
<th>Extraction automatique avec NooJ</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidats termes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nombre</td>
<td>472</td>
<td></td>
</tr>
<tr>
<td>Termes retenus</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>33,05</td>
<td></td>
</tr>
</tbody>
</table>

**Tableau 8-3 : Résultats obtenus avec NooJ.**

Nous constatons que tous les termes existants (156) ont bien été extraits avec NooJ, cependant un très grand bruit est observé. Le bruit est dû aux ambiguïtés syntaxiques pour les noms.
Exemples :
- unité chargée  Nom Adjectif,
- un certain "un" peut être un Nom ou un DET,
- la prochaine "la" peut être un Nom ou un DET.

Ces candidats-termes ne sont pas retenus car ne sont pas termes d’informatique.

8.5.2.2 Extraction à partir d’un corpus

Trois textes constituent le corpus :
- Texte2-chantal.doc, 228Ko, 53 pages, 20660 mots, traitant de l’histoire du parallélisme ;
- Texte1-chantal.doc, 149 Ko, 45 pages, 12047 mots, traitant de l’histoire du C++ ;
- Chantal-3.doc, 299K0, 96 pages, 16294 mots, regroupant divers textes d’informatique.

Dans la figure 8.18, nous avons les trois textes du corpus.

![Figure 8-18 : Les trois textes testés avec NooJ.](image)

Pour les termes de longueur deux et trois, nous avons le patron syntaxique décrit, dans la figure 8.19.
Cette grammaire locale qui est un patron syntaxique, permet l'extraction des termes de longueur deux et trois avec Nooj. Une analyse linguistique est effectuée, puis une recherche des termes composés est faite, grâce à la commande "Locate Pattern". "Locate pattern" est utilisé avec la grammaire locale, syntaxique "longdeuxtrois.nog.

Les résultats de la recherche sont donnés dans la fenêtre de "concordances". Nous y trouvons le nom du fichier texte où le terme candidat est trouvé, la colonne "seq" qui contient les termes candidats.

Dans la figure 8.20, suivante, nous pouvons voir toutes ces informations.
Dans le tableau 8.4, nous avons les résultats suivants obtenus après l’extraction des termes de longueur deux et trois.

<table>
<thead>
<tr>
<th>Extraction automatique avec NooJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidats termes</td>
</tr>
<tr>
<td>Nombre</td>
</tr>
<tr>
<td>6031</td>
</tr>
</tbody>
</table>

**Tableau 8-4 : résultats obtenus avec NooJ.**

Les termes composés retenus manuellement dans les trois textes, sont au nombre de 1454, bien que les candidats termes soient au nombre de 6031.

#### 8.5.3 Extraction de terminologie à l’aide de la méthode statistique ANA

Le système ANA est une méthode statistique d’extraction automatique, présentée à la section 1.4.2.1 du chapitre "État de l’art".

Trois textes ont été fournis pour être traités avec le système ANA, par Chantal Enguehard (Enguehard C. 1992), ce sont ceux (Texte2-chantal.doc, Texte1-chantal.doc, Chantal-3.doc) utilisés pour créer le corpus de la section 8.5.2.2.

Les 3 textes ont été désaccentués et le système ANA a été initialisé avec les 8 termes suivants (bootstrap) :
DONNEES, FICHIER, FONCTION, MACHINE, MEMOIRE, PROCESSUS, RESEAU, VARIABLE.

473 candidats-termes ont été reconnus sous 7750 formes différentes. Les termes avec leurs variantes sont présentés par le système sous la forme d'un concordancier dans un fichier texte. Par exemple, le candidat-terme "ouverture d'un fichier" (5 occurrences) a été trouvé sous les formes :
- ouverture d'un fichier (3 occurrences),
- ouverture de fichier (1 occurrence),
- ouvertures du fichier (1 occurrence).

Nous avons ci-dessous un extrait des statistiques d’apparition de quelques candidats termes simples ou composés, trouvés par le système ANA :

<table>
<thead>
<tr>
<th>N°</th>
<th>Termes</th>
<th>Nombre</th>
<th>%</th>
<th>N°</th>
<th>Termes</th>
<th>Nombre</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>237</td>
<td>processus</td>
<td>193</td>
<td>43,53</td>
<td>141</td>
<td>donnees*</td>
<td>193</td>
<td>43,53</td>
</tr>
<tr>
<td>139</td>
<td>utiliser</td>
<td>127</td>
<td>43,53</td>
<td>118</td>
<td>fonction*</td>
<td>127</td>
<td>43,53</td>
</tr>
<tr>
<td>109</td>
<td>reseau*</td>
<td>103</td>
<td>43,53</td>
<td>101</td>
<td>systeme</td>
<td>103</td>
<td>43,53</td>
</tr>
<tr>
<td>101</td>
<td>variable*</td>
<td>100</td>
<td>43,53</td>
<td>93</td>
<td>valeur</td>
<td>100</td>
<td>43,53</td>
</tr>
<tr>
<td>86</td>
<td>differente</td>
<td>82</td>
<td>43,53</td>
<td>81</td>
<td>instructions</td>
<td>82</td>
<td>43,53</td>
</tr>
<tr>
<td>80</td>
<td>informations</td>
<td>78</td>
<td>43,53</td>
<td>74</td>
<td>adresses</td>
<td>78</td>
<td>43,53</td>
</tr>
<tr>
<td>73</td>
<td>nombre</td>
<td>70</td>
<td>43,53</td>
<td>3</td>
<td>transfert de donnees</td>
<td>70</td>
<td>43,53</td>
</tr>
<tr>
<td>13</td>
<td>protocoles de routage</td>
<td>13</td>
<td>43,53</td>
<td></td>
<td>type d utilisation</td>
<td>13</td>
<td>43,53</td>
</tr>
<tr>
<td>13</td>
<td>entrees sorties</td>
<td>13</td>
<td>43,53</td>
<td>3</td>
<td>terminaison d’une thread</td>
<td>13</td>
<td>43,53</td>
</tr>
<tr>
<td>3</td>
<td>terminal de controle</td>
<td>3</td>
<td>43,53</td>
<td>3</td>
<td>verrou incompatible</td>
<td>3</td>
<td>43,53</td>
</tr>
<tr>
<td>3</td>
<td>type nom</td>
<td>3</td>
<td>43,53</td>
<td>3</td>
<td>type de serveur</td>
<td>3</td>
<td>43,53</td>
</tr>
</tbody>
</table>

Après étude, d'une part du fichier texte (concordancier) contenant les candidats termes simples, et composés de longueur deux (non dupliqués), fournis par ANA et d'autre part des trois textes (extraction manuelle des termes de longueur deux), nous avons établi, les résultats donnés tableau 8.5.

<table>
<thead>
<tr>
<th>Nombre de formes reconnues de longueur deux</th>
<th>Nombre de formes non reconnues de longueur deux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nombre</td>
<td>%</td>
</tr>
<tr>
<td>165</td>
<td>43,53</td>
</tr>
</tbody>
</table>

Tableau 8-5 : Résultats obtenus avec ANA.

Les termes retenus sont laissés désaccentués.

*signifie que le terme a été retenu dans le bootstrap.
8.5.4 Étude des résultats

Dans le système l’extraction de terminologie est faite après analyse linguistique et utilisation de "Locate pattern" pour trouver les termes selon le patron syntaxique défini pour les termes de longueur deux et trois. Le bruit est très élevé mais tous les termes sont extraits.

Dans le cas du système ANA, les candidats termes composés extraits sont uniquement ceux de longueur deux (sauf quelques rares termes de longueur trois). Il y a peu de bruit, mais tous les termes ne sont pas extraits.

Nous pensons que ces résultats peuvent être améliorés dans les deux méthodes avec :

- Le rajout de termes dans le bootstrap utilisé dans ANA ;
- L’utilisation du dictionnaire de mots-clés, "concepts_info", présenté à la section 7.5.1 afin de réduire le bruit ;
- Le rajout de règles de levée des ambigüités dans les patrons syntaxiques de recherche.

8.5.5 Statistiques

Nous avons appliqué au texte "texte1-chantal.not", le patron syntaxique décrit dans la section 8.2.2.2. Nous avons construit ce graphe afin d’extraire les termes de longueur deux et trois dans les textes d’informatique. Nous obtenons avec les statistiques de la figure 9.21.
Lorsque le SS est inférieur à 2 et supérieur à -2 on ne peut rien dire d'intéressant. Les SS grands correspondent à des "accidents" ou des choses "intéressantes". Les SS petits signifient qu'il n'y a rien de particulier dans la concordance.

On constate, qu'il y a un nombre anormalement fort "d'éléments" reconnus dans la partie 18 de notre corpus.
Les résultats donnés dans la colonne SS (score standard) de la figure 8.21 sont mis sous forme de graphique, dans la figure 8.22.

![Figure 8-22 : Graphique des statistiques obtenues pour la colonne SS.](image)

Les axes de coordonnées sont :
- les ordonnées sont la valeur de SS,
- les abscisses, les parties du texte.

La partie 18 montre bien, un grand nombre de SS, donc pour nous un grand nombre d'éléments, de longueurs deux et trois est extrait.

### 8.5.6 Indexation automatique de textes

#### 8.5.6.1 Indexation automatique

L'indexation consiste, en toute généralité, à substituer au document originel une représentation de ce document sous la forme d'une description abrégée, le plus souvent textuelle. Sa fonctionnalité est de permettre de repérer rapidement au sein d'un ensemble (ou d'un document), les documents (ou les extraits) pertinents en fonction d'une requête donnée.
Cette définition, très générale, est conforme à celle donnée par la norme en vigueur qui a pour objet d'établir des méthodologies valables et cohérentes pour la description et la caractérisation des documents à l'aide de représentations de concepts.

L’indexation est une tâche centrale du traitement documentaire puisqu'elle préside à la constitution d'une description qui sert de substitut au contenu dans les phases de recherche et de consultation du document (Auffret G. 2000).


### 8.5.6.2 Exemple d’indexation de textes avec NooJ

Lors de l’analyse linguistique avec Nooj, nous appliquons, au texte "texte1-chantal.not", uniquement la ressource linguistique "concept_info. nod" (dictionnaire de mots-clés d’informatique que nous avons conçu et présenté à la section 7.5.1). La concordance est :  

![Concordance du texte en appliquant le dictionnaire "concepts_info".](image)

---

45 La norme NF Z 47-102 de l’AFNOR (octobre 1993)
Dans la figure 8.23, nous avons un extrait de la concordance obtenue. Nous voyons qu’il y a 881 concordances pour les 111 mots clés du dictionnaire de concepts.

Nous exportons la concordance avec "Export Index" dans la figure 8.24:

![Image de la concordance](image)

**Figure 8-24 : Exportation de la concordance du texte.**

Nous listons un extrait de l’index (mot et localisation) obtenu:

- accès, 40957, 57162, 57549
- adressage, 56611, 62923, 63175
- adresse, 60112, 4789, 34301
- Adresse, 42056
- algorithme, 75673, 76452, 77873, 78161
- algorithmes, 76905, 76161, 75052
- assembleur, 18322, 19046
- base, 1550, 54329, 64360, 64714, 65555, 41189
- Base, 6275, 6015, 6143
- binaire, 67192, 72577, 18527, 18963, 8338, 19277, 18239, 6533, 66879
bloc, 14109, 66297, 14191, 14293, 67657, 67732, 67920, 14381, 73985, 14442, 20374, 75698, 75720, 20576, 20785, 21365, 14543, 13877, 28639, 14884

branchement, 22478

calcul, 76184, 71635, 16520, 16271, 16493

CARACTERE, 73463
caractère, 9013, 67205, 11656, 12788, 64620, 8164, 8562, 72964, 8882, 8932

Caractère, 8058, 5744, 5782, 12836, 12939
caractères, 1529, 8860, 11415, 27621, 71930, 72054, 72146, 67126, 15693, 72979, 1465

clé, 73779, 73816, 22988, 74440, 74565, 74576, 74990, 75023, 9141, 75359, 75655, 22685, 31067, 31712, 75822, 75892, 31732, 32082, 76197, 76245, 76358, 32171, 32228, 77788, 22920, 77989, 78062, 73737, 78288, 78477, 78624
codage, 70387, 66796, 73030, 73181, 72805, 72816, 77795

8.6 Conclusion

Ce chapitre a montré la façon dont nous avons utilisé NooJ avec des textes d’informatique et effectué l’analyse linguistique, l’annotation automatique de textes, des recherches linguistiques et des concordances. Une analyse linguistique de corpus d’informatique est faite à l’aide des ressources linguistiques que nous avons mises en place pour l’informatique (dictionnaires électroniques, règles de flexions, grammaires locales de termes). Nous pouvons aussi indexer les textes d’informatique à l’aide du dictionnaire de concepts construit.

Un corpus de trois textes a été retenu pour tester l’extraction automatique de terminologie informatique. La comparaison entre les deux méthodes d’extraction n’est pas possible, les résultats obtenus avec ANA ayant fourni en grande majorité, des termes de longueur deux. Grâce au patron syntaxique, développé, des termes de longueur deux et trois ont été extraits.
Conclusion et perspectives
Conclusion et perspectives

L’objectif principal visé par ce travail a été de construire un dictionnaire électronique de terminologie informatique, en vue de l’analyse syntaxique et sémantique de textes d’informatique. Pour pouvoir effectuer l’analyse automatique de textes d’informatique avec NooJ, nous avons élaboré toutes les ressources linguistiques, nécessaires à la langue technique retenue.

Nous remplissons le dictionnaire des termes composés "info_comp" avec les termes recensés. Aussi, dans un premier temps, nous avons effectué une collecte d’une trentaine d’ouvrages divers d’informatique. Ensuite, nous avons lu ces ouvrages et effectué une extraction manuelle de termes composés, à partir de ces textes écrits. L’extraction automatique avec NooJ, quant à elle n’a été faite qu’après une familiarisation avec NooJ.

Nous avons fait une étude des termes extraits (composants, longueur, flexion) et classé les termes composés dans la catégorie syntaxique qui leur correspond. La conception des règles de flexions des termes composés et la formalisation du dictionnaire ont été réalisées en parallèle. Chaque terme retenu est codifié avec le formalisme NooJ d’entrée de dictionnaire, et ajouté au dictionnaire "info_comp" réalisé. Cela nous a permis de remplir le dictionnaire avec les termes composés de longueur deux et plus.

Lors de l’extraction nous avons constaté que beaucoup de termes composés sont des sigles et des abréviations. Nous avons aussi, mis en place des dictionnaires séparés des sigles et des abréviations, car les informations contenues dans les entrées sont différentes.

Nous avons aussi construit les grammaires locales des termes, en regroupant par famille les termes, autour d’un concept.

Dans un deuxième temps, nous avons pu extraire automatiquement environ 1500 termes à l’aide des patrons syntaxiques que nous avons construits, ils sont en cours de codification. Les patrons syntaxiques élaborés permettent l’extraction automatique de terminologie informatique avec NooJ. On remarque que l’analyse linguistique et l’application des patrons de recherche permettent d’extraire tous les termes existants, mais avec un bruit élevé. Des contraintes peuvent être introduites dans les patrons afin de réduire le bruit, par exemple en obligeant à ce que le nom de tête soit un "mot-clé" d’informatique.

On notera, lors de l’analyse linguistique, la reconnaissance automatique des termes composés de longueur deux, et plus est faite, uniquement s’ils se trouvent déjà dans les dictionnaires, ou les grammaires locales établies pour les termes composés.

L’enrichissement des dictionnaires peut se faire, soit en récupérant les termes non reconnus, après analyse linguistique, en examinant s’ils sont des entrées possibles de dictionnaires, soit après l’extraction automatique de terminologie avec NooJ.

Notre contribution est la construction des ressources linguistiques indispensables pour l’analyse automatique de textes d’informatique avec NooJ. Nous avons élaboré quatre dictionnaires d’informatique : pour les termes composés, les sigles, les abréviations, les concepts. Ainsi, le dictionnaire des termes composés contient actuellement 10250 entrées codifiées, celui des sigles 521 entrées pour les lettres A et B, dans celui des abréviations, on a 466 entrées pour
les lettres A, B, C, D, E. Un dictionnaire contenant les concepts d’informatique a été élaboré et contient 111 entrées codifiées.

Nous avons aussi, construit les grammaires flexionnelles, contenant les principales règles de flexions, des termes codifiés dans le dictionnaire, ainsi que les grammaires locales syntaxiques (au nombre de 40) de termes composés, les grammaires locales pour l’extraction automatique de terminologie avec NooJ, ainsi que des grammaires locales de coordination.

Notre contribution comporte enfin un corpus de textes d’informatique, créé à partir de fichiers divers : "+.pdf", "+.ps.", "+.doc.", "+.txt", "+.html" que nous avons transformés en fichiers textes "+.txt" avant de les importer dans NooJ. (On notera qu’actuellement NooJ permet l’importation de fichiers de types divers). Le corpus "textes-info1.noc", de taille 34Mo contient 1066 fichiers textes.

La perspective principale de ce travail est de développer la traduction automatique terme à terme des mots composés d’informatique du français vers l’anglais. En effet, le dictionnaire des termes composés que nous avons construit contient plus de 10250 termes composés, et sachant que 30000 termes sont déjà recensés, nous pouvons envisager en perspective, la traduction automatique, avec la construction des grammaires locales de traduction. Dans (Aoughlis F. 2010), nous trouvons une approche de ce projet.
Annexes
Annexe1 : Liste des ouvrages utilisés pour l'extraction manuelle

*Architectures et programmation*, Gilles Deghilage, édition Addison-Wesley 1992

*8080-8085, Programmation en langage assembleur*, Lance A. Leventhal, édition Radio 1982

*Systèmes d’exploitation, Concepts & algorithmes*, Joffroy Beauquier & Beatrice Bérard, Ediscience 1993

*Système d'exploitation (conception & fonctionnement)*, D.Barron, édition Masson 1986.

*Systèmes d'information, COO* (Rapport ANSI/SPARC norme ISO 82) N°695,

*Dictionnaire CEGOS; "définition du vocabulaire Micro-informatique & Micro-électronique avec lexique Anglais Français"*
Christiane Morvan, CEDIC/ Fernand Nathan 1980.

*Crocus, systèmes d'exploitation des ordinateurs*, édition DUNOD 1975.


*Glossaire informatique des termes de la commission interministérielle de terminologie informatique*, Philippe Deschamps, INRIA, 1995-1998


*Principes des systèmes d'exploitation des ordinateurs*, Sacha Krakowiak édition MASSON 1887


*Le monde informatique du 04 octobre 1996 N°692*


*Structure et fonctionnement des ordinateurs*, J.P.Menadier, édition LAROUSSE 1971


*Paralations, un modèle universel de programmation parallèle*, Gary W.Sabot Bordas ,1990

*Dictionnaires électroniques & analyse automatique (le système INTEX)*
Max Siberstein, édition MASSON 1993

Annexe2 : Outils formels de base, théorie

De multiples définitions des machines à états finis existent. Les différences entre celles-ci consistent essentiellement dans le choix des symboles décrivant les composants dont les transitions et les états. Sur le plan théorique, ces définitions sont équivalentes ; elles contribuent à la simplification de la description et l’explication des applications qui y sont liées.

**Expressions rationnelles**

Définition 1 (Alphabet, mot, langage) :

Soit $\Sigma$ un ensemble fini non vide de symboles appelé alphabet. Les symboles de $\Sigma$ sont dénommés lettres ou caractères. Un mot $\omega$ sur $\Sigma$ est une suite finie $x_1, \ldots, x_n$ d’éléments de $\Sigma$. $\omega$ est aussi appelé séquence et sa longueur est notée $|\omega|$. On note $\Sigma^*$ l’ensemble des mots définis sur $\Sigma$. Un langage est un sous-ensemble de $\Sigma^*$.

L’ensemble des langages réguliers sur $\Sigma$ peut-être défini récursivement par les règles suivantes :

- le langage vide $\emptyset$ est un langage régulier ;
- le langage $\{\varepsilon\}$ est un langage régulier ;
- pour tout $x$ de $\Sigma$, $\{x\}$ est un langage régulier ;
- si $L_1$ et $L_2$ sont des langages réguliers, alors nous pouvons construire les langages réguliers $L_1 \cup L_2$ : l’union ensembliste des deux langages ;
  $L_1.L_2$ : la concaténation des deux langages ;
  $L_1^*$ : la fermeture de Kleene$^{46}$

Notons que certains langages ne sont pas réguliers, comme par exemple le langage $\{a^n.b^n \mid n > 0\}$ défini sur le vocabulaire $\{a, b\}$. En particulier, la plupart des langages de programmation ne sont pas des langages réguliers.

Définition 2 (Expression rationnelle) :

Tout langage régulier sur $L$ peut être décrit par une expression rationnelle, c’est-à-dire un terme construit sur l’alphabet $\Sigma \cup \{+, \cdot, *, \varepsilon, \emptyset\}$ de la manière suivante :

- $\emptyset$ décrit le langage vide $\emptyset$ ;
- $\varepsilon$ décrit le langage $\{\varepsilon\}$ ;

$^{46}$ La fermeture de Kleene du langage régulier L1, notée L1*, désigne l'ensemble de toutes les chaînes de caractères qui peuvent être formées en concaténant zéro, une ou plusieurs chaînes de L1.
• pour tout x appartenant à Σ, x décrit le langage {x} ;  
• si e1 et e2 sont des expressions rationnelles sur Σ décrivant respectivement les langages L1 et L2 alors :  
  ✓ e1 + e2 décrit le langage L1 U L2  
  ✓ e1.e2 décrit le langage L1.L2  
  ✓ e1* décrit le langage L1*  

Par convention, nous considérons que l’opérateur uneaire « * » est plus prioritaire que l’opérateur « . », lui-même plus prioritaire que l’opérateur « + ».

Nous donnons, ci-dessous, quelques exemples d’expressions rationnelles définies sur le vocabulaire {a, b, c}, ainsi que les langages (réguliers) correspondants :

<table>
<thead>
<tr>
<th>Expression rationnelle</th>
<th>Eléments du langage</th>
</tr>
</thead>
<tbody>
<tr>
<td>a + b.c</td>
<td>{a, bc}</td>
</tr>
<tr>
<td>(ab)*</td>
<td>{ε, ab, abab, ababab, …}</td>
</tr>
<tr>
<td>ab + c*</td>
<td>{ε, ab, c, cc, ccc, …}</td>
</tr>
</tbody>
</table>

Notons que, les langages réguliers sont les langages reconnus formellement par des automates finis.

**Automates à états finis**

Un automate est un modèle abstrait de machine qui peut être vu comme une représentation graphique des données à modéliser. Il possède un alphabet, un ensemble d’états et une relation de transition. Une transition, δ(q, a)= q, indique que l’état q est atteint à partir de l’état p en lisant le symbole a de l’alphabet.

On formalise les notions liées aux automates par les définitions suivantes :

**Définition 3 (Automate fini non déterministe) :**

Un automate fini non déterministe est un quintuplet A = (Q, Σ, δ, I, F), où :
• Q est un ensemble fini non vide d’états ;
• Σ est l’alphabet ;
• δ : Q x Σ → 2^Q est une fonction de transition. Quand la transition δ(q,a) est non définie, nous notons que δ(q,a) = ⊥. Nous étendons la fonction δ à la fonction de transition de mots δ^* : 2^Q x Σ* → 2^Q définie comme suit : (où a ∈ Σ, x ∈ Σ*) :
δ(q, ε) = q
δ^*(q, ax) =  
\[ \begin{align*} 
δ^*(δ(q,a),x) & \text{ si } δ(q,a) \neq ⊥ \\
⊥ & \text{ sinon} 
\end{align*} \]
• I ⊆ Q est l’ensemble non vide des états initiaux ;
• F ⊆ Q est l’ensemble non vide des états finaux.

**Définition 4 (Automate fini déterministe) :**

Contrairement au cas de non déterminisme, un automate fini déterministe ne comporte qu’un seul état initial q₀, c’est-à-dire I = {q₀} et |I|=1. De plus, pour chaque couple (q, a), où q∈ Q et a∈Σ, la
fonction $\delta(q,a)$ détermine une valeur unique ; sa fonction de transition est définie comme étant : $\delta : Q \times \Sigma \rightarrow Q$.

Le déterminisme d’un automate à états finis nous garantit une séquentialité de l’analyse d’un mot.

**Transducteurs à états finis**

Les transducteurs sont des automates avec deux alphabets pour étiqueter les transitions. Le deuxième jeu de symboles est appelé l'alphabet d'émission. Tandis qu’un automate à états finis se contente de reconnaître les éléments d’un langage, un transducteur est un automate “étendu” permettant, pour chaque étape de transition, d’égalemnt produire un symbole en sortie. Il existe plusieurs formalismes pour décrire un transducteur à états finis, prenons celui de la machine de Mealy :

**Définition 5 (Transducteur à états finis) :**

Un transducteur à états finis est un quintuplet $T = (Q, \Sigma, \delta, q_0, F)$, où :

• $Q$ est un ensemble fini non vide d’états ;

• $\Sigma$ est un alphabet de symboles complexes. Chaque symbole est constitué d’une paire (e,s), où $e \in$ à un alphabet d’entrée $E$, et $s \in$ à un alphabet de sortie $S$. $\Sigma \subseteq E \times S$ ;

• $\delta : Q \times \Sigma \rightarrow Q$ est la fonction de transition ;

• $q_0 \in Q$ est l’état initial ;

• $F \subseteq Q$ est l’ensemble non vide des états finaux.

Les transducteurs ainsi définis sont dits sous-séquentiels ; une information est émise par leurs états terminaux. Un mot est reconnu par un transducteur de la même façon que par un automate, mais de plus une sortie est produite.

**RTNs / ATNs**

Un réseau de transitions récursif (Recursive Transition Network : RTN) est défini par ensemble de graphes semblables à ceux d’un automate fini dans lequel chaque transition permet d’atteindre un état terminal ou non-terminal. La différence par rapport à un automate à états finis se situe au niveau du traitement des états non-terminaux : le RTN traite chaque état non-terminal comme un éventuel appel à d’autres réseaux (y compris le RTN lui-même)

**Définition 6 (Réseau de Transition Récursif - RTN) :**

Un RTN peut être défini formellement par un 6-uplet $M = (Q, I, \Sigma, \delta, q_0, F)$, où :

• $Q$ est un ensemble fini non vide d’états ;

• $I$ est l’ensemble des états sous-initiaux (états qui étiquettent au moins une transition du transducteur RTN, et représentent donc un appel récursif au sous-RTN) ;
• \( \Sigma \) est un alphabet de symboles complexes. Chaque symbole est constitué d’une paire \((e, s)\) avec \(e \in \text{un alphabet d'entrée } E\), et \(s \in \text{un alphabet de sortie } S\). \(\Sigma \subseteq E \times S\);

• \(\delta: Q \times (\Sigma \cup E \cup \{ \varepsilon \}) \rightarrow Q\) est la fonction de transition ;

• \(q_0 \in Q\) est l’état initial ;

• \(F \subseteq Q\) est l’ensemble non vide des états finaux.

Un réseau de transitions récursif (RTN) peut être étendu pour donner un réseau de transitions augmenté (ATN : Augmented Transition Network). Un ATN est un RTN auquel s’ajoutent certaines extensions qui lui donnent un pouvoir descriptif supérieur à celui d’une grammaire non-contextuelle. Trois extensions sont apportées, il s’agit notamment :

• d’ajouter des registres aux réseaux de transitions ;

• d’imposer des conditions sur les transitions ;

• d’associer des actions aux transitions effectuées.

**Les machines à états finis dans NooJ**

La plateforme linguistique NooJ fait usage des différentes machines à états finis citées ci-dessus afin de représenter des données, formaliser des phénomènes linguistiques et analyser des textes :

• Les expressions rationnelles représentent un moyen rapide pour les requêtes simples. Par exemple, lorsque la séquence recherchée consiste en quelques mots, il est possible d'énumérer ces mots directement dans une expression rationnelle ;

• Les transducteurs à états finis peuvent servir à décrire divers phénomènes linguistiques; notamment pour associer chaque patron retenu à un résultat d'analyse. Ils sont aussi utilisés pour stocker des données lexicalisées ainsi que toutes les informations morpho-syntaxiques qui s’y rattachent ;

Les automates à états finis ne sont qu’un cas particulier des transducteurs : ils produisent le mot vide. Ils servent à localiser des phénomènes morpho-syntaxiques dans un corpus, extraire les séquences reconnues, construire des tables de concordances, etc. ;

• Les RTNs se présentent sous la forme d’ensembles organisés de graphes. Ces graphes peuvent eux-mêmes être des automates, des transducteurs à états finis ou des RTNs. Dans la pratique, les RTNs sont utilisés pour construire des bibliothèques de graphes facilement réutilisables ;
Bibliographie
Bibliographie


Assadi, H. & Bourigault, D. 1996. Acquisition et modélisation de connaissances à partir de textes : outils informatiques et éléments méthodologiques, In Actes du 10ème congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA’96): AFCET.


Bourigault D. 1994. « LEXTER, un logiciel d’extraction de terminologie. Application à l’extraction de connaissances à partir de textes », Thèse de doctorat en mathématiques,


Courtois, B. 1990a. Dictionnaire électronique du LADL pour les mots simples du français, DELAS V06/2, LADL, Université de Paris 7, CNRS.


Drouin P. 2003a. « Term extraction using non technical corpora as a point of leverage ». in Terminology, vol 9, No1, pp. 99-117.


L’Homme M.C. 2000. Évaluation de logiciels d'extraction de terminologie : examen de quelques critères. Communication donnée à la Réunion Inter institutions sur la terminologie et la traduction assistées par ordinateur (JIAMCATT), Office des Nations Unies, Vienne (Autriche)


Morin E. 1999. Extraction de liens sémantiques entre termes à partir de corpus de textes techniques, thèse de Doctorat, IRIN, Nantes.


Roche M., Heitz Thomas, Matte-Tailliez Oriane, Kodratoff Yves. 2004a. EXIT : Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés, *Exposé dans le cadre de la conférence JADT'04 (7th International Conference on the Statistical Analysis of Textual Data), Louvain-La-Neuve, Belgique.*


Roche M., Kodratoff Y. 2006. Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition, *dans proceedings of onToContent’06 workshop (Ontology content and evaluation in Enterprise) - OTM’06, Springer-Verlag, LNCS, p.1107-1116, octobre 2006, Montpellier, France.*


Sabatier P. 2009. Les Verbes français de J. Dubois et F. Dubois-Charlier : Prises de vues syntaxiques, *LIF.*


Vergnes J. 2004. Un exemple de traitement "alingue" endogène : extraction de candidats termes dans des corpus bruts de langues non identifiées par étiquetage mot vide - mot plein. *Conférences invitées à l’Université Stendhal Grenoble 3 - novembre 2004*