Modèles et outils pour des bases lexicales "métier" multilingues et contributives de grande taille, utilisables tant en traduction automatique et automatisée que pour des services dictionnairiques variés

Ying Zhang

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **INFORMATIQUE**

Arrêté ministériel : 7 août 2006

Présentée par

Ying ZHANG

Thèse dirigée par **Christian BOITET** et
codirigée par **Valérie BELLYNCK** et **Mathieu MANGEOT-NAGATA**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale « Mathématiques, Sciences et Technologies de l'Information, informatique »**

Modèles et outils pour des bases lexicales "métier" multilingues et contributives de grande taille, utilisables tant en traduction automatique et automatisée que pour des services dictionnairiques variés

Thèse soutenue publiquement le **28 juin 2016**,
devant le jury composé de :

Prof. Ahmed LBATH
Professeur, UGA, Président

Prof. Denis MAUREL
Professeur, Tours, Rapporteur

Prof. Alain POLGUERE
Professeur, Nancy, Rapporteur

MdC. Mathieu LAFOURCADE
Maître de conférence, Montpellier II, Examinateur

Prof. Antoine Chalvin
Professeur, INaLCO, Examinateur

Prof. Christian BOITET
Professeur, UGA, Directeur

MdC. Valérie BELLYNCK
Maître de conférence, Grenoble-INP, Co-Directrice

MdC. Mathieu MANGEOT-NAGATA
Maître de conférence, UdSavoie, Co-Directeur

UNIVERSITÉ DE GRENOBLE

N° attribué par la bibliothèque
/__/__/__/__/__/__/__/__/__/__/

THÈSE

pour obtenir le grade de

DOCTEUR ÈS SCIENCES

délivré par l'UNIVERSITÉ GRENOBLE ALPES

Spécialité : "INFORMATIQUE"

Thèse préparée au laboratoire GETALP-LIG (CNRS-INPG-UJF) dans le cadre de

l'École Doctorale "Mathématiques, Sciences et Technologies de l'Information, Informatique"

présentée et soutenue publiquement

par

Ying ZHANG

Le 28/6/2016

Modèles et outils pour des bases lexicales "métier" multilingues et contributives de grande taille, utilisables tant en traduction automatique et automatisée que pour des services dictionnairiques variés

JURY

M. Ahmed LBATH, prof. UGA	Président
M. Denis MAUREL, prof. Tours	Rapporteur
M. Alain POLGUERE, prof. Nancy	Rapporteur
M. Mathieu LAFOURCADE, MdC Montpellier II	Examinateur
M. Antoine Chalvin, prof. INaLCO	Examinateur
M. François BROWN DE COLSTOUN, PDG de L&M	Invité
M. Christian BOITET, prof. UGA	Directeur de thèse
Mme Valérie BELLYNCK, MdC G-INP	Codirecteur de thèse
M. Mathieu MANGEOT-NAGATA, MdC UdSavoie	Codirecteur de thèse

Remerciements

Les travaux présentés dans cette thèse ont fait l'objet d'une convention CIFRE entre la société Lingua et Machina et le GETALP (Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole) du LIG (Laboratoire d'Informatique de Grenoble).

Je voudrais tout d'abord remercier le professeur Christian Boitet, mon directeur de thèse, qui est à l'origine de ce travail. C'est un honneur pour moi de travailler avec lui et je ne peux qu'admirer son talent. Je lui suis infiniment reconnaissante, non seulement parce qu'il a accepté de me prendre en thèse, mais aussi parce qu'il a partagé ses idées avec moi. Il a dirigé ma thèse avec beaucoup de patience et il a dédié beaucoup de temps à mon travail en étant toujours très disponible et en venant me chercher très souvent pour que l'on discute, ce qui m'a énormément encouragée. Je le remercie aussi d'avoir lu très sérieusement beaucoup de versions préliminaires de ces travaux.

J'adresse de chaleureux remerciements à mon co-directeur M. Mathieu Mangeot. Il m'a accueilli gentiment chaque fois que je venais l'embêter pour lui poser des questions et ses réponses m'ont toujours éclairci les idées. J'admirerai toujours son savoir ainsi que sa capacité à l'exposer et à le partager. Il a aussi dédié beaucoup de son temps à discuter avec moi à propos de mon avenir et je lui en suis très reconnaissante.

Un grand merci à mon autre co-directeur Mme Valérie Bellynck. Tout d'abord pour son accueil, mais surtout pour les nombreuses discussions que nous avons eues sur son concept de "langage narratif" et pour son aide constante en ce qui concerne les supports techniques pour mes développements. Elle m'a beaucoup appris, non seulement sur la vie scientifique, mais aussi sur la vie culturelle, et sur la vie tout court. J'ai énormément apprécié son enthousiasme et sa sympathie.

Je tiens à remercier particulièrement le président de Lingua et Machina, M. François Brown de Colstoun, sans qui cette thèse CIFRE n'aurait sûrement jamais vu le jour.

Mes remerciements s'adressent également aux professeurs Denis Maurel et Alain Polguère pour avoir accepté de rapporter sur ma thèse et pour l'intérêt qu'ils ont porté à ces travaux. J'espère qu'à l'avenir nous serons amenés à échanger de nouveau sur ce sujet passionnant.

Un grand merci aussi au professeur Ahmed Lbath pour m'avoir fait l'honneur de présider le jury de ma thèse.

Antoine Chalvin, professeur d'estonien à l'INaLCO, et contributeur majeur du projet GDEF (Grand Dictionnaire Estonien-Français) développé en JIBIKI avec le support de M. Mangeot depuis 2003, a accepté de participer à ce jury malgré ses lourdes tâches à Paris. Je l'en remercie d'autant plus.

Mathieu Lafourcade, Maître de Conférences HDR à Montpellier, et spécialiste de la création contributive de grandes bases lexicales vers des "jeux sérieux", et de désambiguïsation lexicale, a accepté de faire partie de mon jury. C'est un grand honneur pour moi de les remercier.

Mes derniers remerciements iront évidemment à ma famille. Je pense tout d'abord à mes parents qui m'ont toujours épaulée dans mes études, mes décisions et mes choix, même si je n'ai pas pu être à leurs côtés pendant une longue période. Ensuite, je voudrais remercier mon mari Wang Ling Xiao, qui partage ma vie, et a fait aussi sa thèse en CIFRE avec L&M, sur un sujet complémentaire (les bases de données "corporales") et mon fils Wang Xin Di qui m'a apporté beaucoup de joie durant ma vie de doctorante.

Résumé en français

Notre recherche se situe en lexicographie computationnelle, et concerne non seulement le support informatique aux ressources lexicales utiles pour la TA (traduction automatique) et la THAM (traduction humaine aidée par la machine), mais aussi l'architecture linguistique des bases lexicales supportant ces ressources, dans un contexte opérationnel (thèse CIFRE avec L&M).
Nous commençons par une étude de l'évolution des idées, depuis l'informatisation des dictionnaires classiques jusqu'aux plates-formes de construction de vraies "bases lexicales" comme JIBIKI-1 [Mangeot, M. et al., 2003 ; Sérasset, G., 2004] et JIBIKI-2 [Zhang, Y. et al., 2014]. Le point de départ a été le système PIVAX-1 [Nguyen, H.-T. et al., 2007 ; Nguyen, H. T. & Boitet, C., 2009] de bases lexicales pour systèmes de TA hétérogènes à pivot lexical supportant plusieurs volumes par "espace lexical" naturel ou artificiel (UNL). En prenant en compte le contexte industriel, nous avons centré notre recherche sur certains problèmes, informatiques et lexicographiques.
Pour passer à l'échelle, et pour profiter des nouvelles fonctionnalités permises par JIBIKI-2, dont les "liens riches", nous avons transformé PIVAX-1 en PIVAX-2, et réactivé le projet GBDLEX-UW++ commencé lors du projet ANR TRAOUIERO, en réimportant toutes les données (multilingues) supportées par PIVAX-1, et en les rendant disponibles sur un serveur ouvert.
Partant d'un besoin de L&M concernant les acronymes, nous avons étendu la "macrostructure" de PIVAX en y intégrant des volumes de "prolexèmes", comme dans PROLEXBASE [Tran, M. & Maurel, D., 2006]. Nous montrons aussi comment l'étendre pour répondre à de nouveaux besoins, comme ceux du projet INNOVALANGUES. Enfin, nous avons créé un "intergiciel de lemmatisation", LEXTOH, qui permet d'appeler plusieurs analyseurs morphologiques ou lemmatiseurs, puis de fusionner et filtrer leurs résultats. Combiné à un nouvel outil de création de dictionnaires, CREATDICO, LEXTOH permet de construire à la volée un "mini-dictionnaire" correspondant à une phrase ou à un paragraphe d'un texte en cours de "post-édition" en ligne sous IMAG/SECTRA, ce qui réalise la fonctionnalité d'aide lexicale proactive prévue dans [Huynh, C.-P., 2010]. On pourra aussi l'utiliser pour créer des corpus parallèles "factorisés" pour construire des systèmes de TA en MOSES.

Abstract in English

Our research is in computational lexicography, and concerns not only the computer support to lexical resources useful for MT (machine translation) and MAHT (Machine Aided Human Translation), but also the linguistic architecture of lexical databases supporting these resources in an operational context (CIFRE thesis with L&M).
We begin with a study of the evolution of ideas in this area, since the computerization of classical dictionaries to platforms for building up true "lexical databases" such as JIBIKI-1 [Mangeot, M. et al., 2003 ; Sérasset, G., 2004] and JIBIKI-2 [Zhang, Y. et al., 2014]. The starting point was the PIVAX-1 system [Nguyen, H.-T. et al., 2007 ; Nguyen, H. T. & Boitet, C., 2009] designed for lexical bases for heterogeneous MT systems with a lexical pivot, able to support multiple volumes in each "lexical space", be it natural or artificial (as UNL). Considering the industrial context, we focused our research on some issues, in informatics and lexicography.
To scale up, and to add some new features enabled by JIBIKI-2, such as the "rich links", we have transformed PIVAX-1 into PIVAX-2, and reactivated the GBDLEX-UW++ project that started during the ANR TRAOUIERO project, by re-importing all (multilingual) data supported by PIVAX-1, and making them available on an open server.
Hence a need for L&M for acronyms, we expanded the "macrostructure" of PIVAX incorporating volumes of "prolexemes" as in PROLEXBASE [Tran, M. & Maurel, D., 2006]. We also show how to extend it to meet new needs such as those of the INNOVALANGUES project. Finally, we have created a "lemmatisation middleware", LEXTOH, which allows calling several morphological analyzers or lemmatizers and then to merge and filter their results. Combined with a new dictionary creation tool, CREATDICO, LEXTOH allows to build on the fly a "mini-dictionary" corresponding to a sentence or a paragraph of a text being "post-edited" online under IMAG/SECTRA, which performs the lexical proactive support functionality foreseen in [Huynh, C.-P., 2010]. It could also be used to create parallel corpora with the aim to build MOSES-based "factored MT systems".

中文摘要

我们的研究领域是计算词典编纂，不仅仅是只关注对 MT（机器翻译）和 MAHT（机助人译）等 IT 相关的词汇资源支持，也是在工业背景下（与 L&M 合作的 CIFRE 论文），支持其资源的词汇数据库的语言学体系的架构建模。
我们从该领域的想法的演变史开始研究，从传统字典的信息化到真正的“词汇数据库” 平台的建立，例如 JIBIKI-1 [Mangeot, M. et al., 2003 ; Sérasset, G., 2004] 和 JIBIKI-2 [Zhang, Y. et al., 2014]。我们的出发点是 PIVAX-1 系统 [Nguyen, H.-T. et al., 2007 ; Nguyen, H. T. & Boitet, C., 2009]，一个词汇数据库，服务于异质的自动翻译系统，通过中间（pivot）词汇来支持多个卷（volume）的自然的或人工的（如 UNL）“词汇空间”。考虑到工业背景，我们把我们的研究集中在 IT 和词典编纂的某些问题上。
为了扩大规模，也为了享受到 JIBIKI-2 支持的新功能，即“复杂链接”，我们把 PIVAX-1 改造为 PIVAX-2，并且在 ANR TRAOUIERO 项目中重新激活 GBDLEX-UW++项目，并重新导入 PIVAX-1 中的所有（多语言）数据，并使其在开放的服务器上可用。
由于 L&M 的一个需求涉及到首字母缩略词，我们扩展了 PIVAX 的宏观结构，用于整合“代词素（prolexèmes） ”类型的卷，如 PROLEXBASE [Tran, M. & Maurel, D., 2006] 中所做的那样。我们也展示了如何扩展新的 PIVAX 的宏观结构以满足新的需求，如在 INNOVALANGUES 项目中。最后，我们创建了一个“词形还原中间件”，LEXTOH，它允许调用多个形态分析器或词形还原器，然后整合和过滤它们的结果。与另一个新的创建字典的工具，CREATDICO，相结合，LEXTOH 允许为 IMAG/SECTRA 系统中，一个正在进行在线“后编辑”的句子或段落文字，在其侧创建一个相对应的“小字典”，这实现了在[Huynh, C.-P., 2010]中所预期的积极词汇辅助功能。我们也能使用该工具，用于创建适用于摩西的要素自动翻译系统所需的“要素”平行语料。

Conventions diverses

1. Les citations sont signalées en général par une référence, et toujours par l'emploi d'un style de caractères particulier (par exemple : ceci est une citation).
2. Les noms de logiciels et de systèmes ou les extensions de fichiers sont dans un autre style particulier, qui utilise de petites majuscules (par exemple : JIBIKI ou XHTML).
3. Les exemples linguistiques (comme *Organisation des Nations Unies*), sont également distingués par un style de caractères particulier.
4. Ce qui concerne des programmes ou des messages d’un logiciel (par exemple : `TIMESLEEP = 30s`) est également dans un style particulier.
5. Les acronymes ou abréviations, comme TA, THAM, GETALP, L&M ou BDLex, sont développés lors de leur première occurrence, et regroupés dans le glossaire qui suit la table des figures.
6. Des "définitions" et des "idées-guide" sont introduites au fur et à mesure de la progression de l'exposé. Elles sont reprises et regroupées après la bibliographie et la netographie.

Table des matières

Remerciements **2**
Résumé en français **3**
Abstract in English **3**
中文摘要 4
Table des matières **5**
Table des figures **9**
Table des tableaux **12**
Glossaire 13
Introduction 15
Chapitre I Contexte de la recherche et problèmes abordés **17**
INTRODUCTION 17
I.1 ÉVOLUTION DES IDEES DE 1980 A 2012 17
I.1.1 1980— : approches visant à faire le lien entre dictionnaires pour les systèmes de TA et pour les humains *17*
I.1.1.1 "Ouverture" des dictionnaires de TA aux traducteurs 17
I.1.1.2 Intégration des dictionnaires des traducteurs dans des aides à la traduction 18
I.1.1.3 Prototypage de dictionnaires intégrant les deux aspects 19
I.1.1.4 Exemples de "bonnes pratiques" et de "cercles vertueux" 19
Conclusion 20
I.1.2 1985— : tentatives pour unifier les informations générales et terminologiques *21*
I.1.3 1992— : dictionnairique avec évolution vers les réseaux et le contributif *22*
I.1.3.1 1992— : travaux sur la construction de dictionnaires informatisés 22
I.1.3.2 1995 ou 1998 : vers la construction contributive de dictionnaires en ligne 27
I.1.4 1991—: évolution vers des bases lexicales *30*
I.1.4.1 Bases lexicales permettant la symétrie 30
I.1.4.2 Structure générique (microstructures et macrostructures) 32
I.1.4.3 Ingénierie des BDLex contributives 33
Synthèse *35*
I.2 SITUATION ET PROBLEMES EN 2012 35
I.2.1 Au GETALP *36*
I.2.1.1 Projets dérivés des thèses de G. Sérasset et de M. Mangeot 36
I.2.1.2 La génération de mini-dictionnaires, une nouvelle application générique 37
I.2.1.3 Résultats de thèses antérieures non liées à JIBIKI 37
I.2.2 Chez L&M *38*
I.2.2.1 Une BDLex simple destinée aux glossaires multilingues 38
I.2.2.2 Problèmes perçus 39
I.2.3 Dans la "communauté scientifique" du TAL *40*
I.2.3.1 Extraction de termes techniques 40
I.2.3.2 Extraction d'expressions polylexicales (EPL) 41
I.2.3.3 Projets de BDLex "sémantiques" 41
I.2.3.4 Création contributive de dictionnaires pour la TA 43
I.2.3.5 Désambiguïsation lexicale 44
I.3 THEMES DE RECHERCHE ABORDES 44
I.3.1 Thèmes encore ouverts *44*
I.3.1.1 Trois thèmes "lexicographiques" 44
I.3.1.2 Trois thèmes concernant la conception de services lexicaux 46
I.3.1.3 Deux thèmes plus liés au GL 47
I.3.2 Thèmes retenus *47*
I.3.2.1 Conception d'une BDLex unifiant tous les types d'unités lexicales (générales, terminologiques, situées et très situées) 48
I.3.2.2 Réalisation d'un service générique de création de mini-dictionnaires pour SECTRA à partir d'une BDLex en JIBIKI 48
I.3.2.3 Conception et implémentation d'un intergiciel de lemmatisation 48
I.3.2.4 Recherche et intégration d'un système de gestion de tâches entre serveurs ou agents à gros grain 48
Remarques finales 49
Chapitre II Extensions fonctionnelles et opérationnelles : de PIVAX-1 à PIVAX-2 ..... **50**
INTRODUCTION 50
II.1 JIBIKI-1 50
II.1.1 Présentation de JIBIKI-1 ..... *50*

II.1.2 Architecture de JIBIKI-1 ..... *51*
II.1.2.1 Spécification 51
II.1.2.2 Implémentation 52
II.1.3 Types de BDLex déjà développés en JIBIKI-1 en 2011 ..... *53*
II.1.3.1 PAPILLON 53
II.1.3.2 LEXALP 54
II.1.3.3 GDEF 54
II.1.3.4 MOTÀMOT 55
II.1.3.5 PIVAX-1 56
II.2 PIVAX-1 56
II.2.1 Motivations *56*
II.2.2 Structure de PIVAX-1 ..... *56*
II.2.2.1 Macrostructure 56
II.2.2.2 Microstructure 57
II.2.2.3 Algorithme de calcul des liens 58
II.2.2.4 Interface 59
II.2.2.5 Début de programmabilité 59
II.2.3 Utilisations de PIVAX-1 ..... *61*
II.2.3.1 Dans OMNIA 61
II.2.3.2 Dans TRAOUIERO et GBDLEX-UW++ 62
II.2.4 Qualités et limitations *64*
II.3 EXTENSIONS FONCTIONNELLES APPORTEES PAR LE PASSAGE A JIBIKI-2 65
II.3.1 Liens riches *65*
II.3.1.1 Motivations 65
II.3.1.2 Implémentation 66
II.3.2 Listes non bornées *68*
II.3.3 Possibilité générique de recherche avec lemmatisation *69*
II.4 PIVAX-2 : OPERATIONNALISATION ET EXTENSION DE PIVAX-1 70
II.4.1 Algorithme générique de calcul des liens *70*
II.4.2 Passage à l'échelle et accélération *71*
II.4.2.1 Données lexicales supportées par PIVAX-2 71
II.4.2.2 Évaluations comparatives des temps de réponse 72
II.4.3 Support au projet GBDLEX-UW++ et mise à disposition de ressources *73*
II.4.3.1 Support au projet GBDLEX-UW++ 73
II.4.3.2 Mise à disposition de l'outil et de données sur le Web 74
Chapitre III Une nouvelle architecture intégrant les données lexicales générales, terminologiques et "situées" : PIVAX-3 75
INTRODUCTION 75
III.1 ANALYSE D'UN PROBLEME POSE PAR L&M 75
III.1.1 Présentation du problème rencontré par L&M *75*
III.1.1.1 Contexte 75
III.1.1.2 Extraits de la "ressource" des clients 76
III.1.1.3 Demande précise de L&M 77
III.1.2 Analyse des problèmes posés *78*
III.1.2.1 Défauts de la ressource lexicale du client 78
III.1.2.2 Problèmes conceptuels 78
III.1.2.3 Problèmes venant de la structure de la BDLex de LIBELLEX 78
III.1.3 Étude lexicologique et lexicographique *78*
III.1.3.1 Nécessité d'un niveau conceptuel (lexies et axies) 78
III.1.3.2 Nécessité de la notion de "prolexème" pour les entités nommées 79
III.1.3.3 Différence entre le niveau terminologique et le niveau des prolexèmes 79
III.1.3.4 Nécessité de distinguer les lexèmes spécifiques d'un "sous-langage" 79
III.1.3.5 Possibilité de se référer à la théorie de la cognition située 79
III.2 ÉLEMENTS D'UNE SOLUTION 79
III.2.1 Systèmes dont on pouvait s'inspirer *79*
III.2.1.1 CJK.ORG 79
III.2.1.2 IATE 81
III.2.1.3 EDR 82
III.2.2 PROLEXBASE et les prolexèmes *83*
III.2.2.1 Le projet PROLEX 83
III.2.2.2 Concepts essentiels venant de Coseriu 84
III.2.2.3 Aspects logiciels : PROLEXBASE 84
III.2.3 Esquisse d'une solution *87*
III.3 CONCEPTION ET IMPLEMENTATION D'UNE SOLUTION BASEE SUR LES "LIENS RICHES" 87

III.3.1 1° prototypage chez L&M *87*
III.3.1.1 Contraintes techniques 88
III.3.1.2 Contraintes industrielles 90
III.3.1.3 Spécification et implémentation d'une solution ad hoc 90
III.3.1.4 Démonstration 91
III.3.2 Une solution plus générale basée sur JIBIKI-2 : PIVAX-3 ..... *92*
III.3.2.1 Extension de l'architecture de PIVAX-2 92
III.3.2.2 Implémentation de PIVAX-3 97
III.3.3 Un exemple complet de gestion des acronymes *101*
III.3.3.1 Exemple en quatre langues pour le sens "Organisation des Nations Unies" 101
III.3.3.2 Modélisation de cet exemple dans PIVAX-3 101
III.3.3.3 Démonstrations 106
III.4 AUTRES EXTENSIONS ENVISAGEABLES 109
III.4.1 Vers l'intégration général-terminologique-situé *109*
III.4.1.1 Intégration pour l'utilisation dans un domaine linguistique spécifique 109
III.4.1.2 Intégration pour l'utilisation des quatre dimensions du diasystème 109
III.4.2 Autres structures (ex: INNOVALANGUES-LEXINNOVA) *110*
III.4.2.1 Contexte du projet INNOVALANGUES-LEXINNOVA 110
III.4.2.2 Proto-structure du dictionnaire 111
III.4.2.3 Modélisation de la macrostructure avec des exemples 112
Chapitre IV Outils génériques pour BDLex "actives" 113
INTRODUCTION 113
IV.1 GESTION DES TRAVAUX PAR ACTIVEMQ 113
IV.1.1 Motivations : besoins attestés et fonctionnalités désirées *113*
IV.1.1.1 Besoins attestés 113
IV.1.1.2 Fonctionnalités désirées 115
IV.1.2 Approches envisageables *115*
IV.1.2.1 Extension de BLEXISMA 115
IV.1.2.2 Reprise du "réseau CASH/LIDIA" 115
IV.1.2.3 Utilisation de JOBCENTER 116
IV.1.2.4 Utilisation de ACTIVEMQ 116
IV.1.3 Intégration d'ACTIVEMQ *117*
IV.1.3.1 Étude et analyse d'ACTIVEMQ 117
IV.1.3.2 Implémentation 117
IV.1.3.3 Expériences et validations 118
IV.2 LEXTOH 119
IV.2.1 Motivations *119*
IV.2.1.1 Support de systèmes de TA de type MOSES 119
IV.2.1.2 Consultation dictionnairique avancée 120
IV.2.1.3 Production de mini-dictionnaires de formats variés 120
IV.2.2 Conception de LEXTOH *121*
IV.2.2.1 Fonctionnalités désirées 121
IV.2.2.2 Architecture globale de LEXTOH 122
IV.2.2.3 Utilisateurs et scénarios 124
IV.2.3 Expérimentation et validation *126*
IV.2.3.1 Interface principale et disponibilité 126
IV.2.3.2 Test du système 127
IV.3 CREATDICO 129
IV.3.1 Motivations : des besoins variés *129*
IV.3.1.1 Besoins des systèmes d'aide à la traduction 129
IV.3.1.1 Besoin humain de lecture active 129
IV.3.1.2 Besoins pour des systèmes de TA 130
IV.3.1.3 Besoins pour des outils spécialisés 131
IV.3.2 Conception de CREATDICO *131*
IV.3.2.1 Fonctionnalités désirées 131
IV.3.2.2 Architecture globale de CREATDICO 133
IV.3.2.3 Utilisateurs et scénarios 134
IV.3.3 Expérimentation et validation *137*
IV.3.3.1 Interface principale et disponibilité 137
IV.3.3.2 Tests fonctionnels 138
Conclusions et perspectives 142
CONCLUSIONS 142
PERSPECTIVES 143
Bibliographie 144

Netographie 152
Table des définitions 153
Table des "idées-guides" 154
Annexes 155
ANNEXE 1 LISTE D'UNE PARTIE DES RESSOURCES LEXICALES 155
ANNEXE 2 EXEMPLE DE DIVERSITE DES UW DANS LE PROJET UNL 157
ANNEXE 3 HISTORIQUE DU PROJET PAPILLON 158
ANNEXE 4 QUESTIONS ET REPONSES SUR HOWNET 159
ANNEXE 5 EXEMPLES DES MICROSTRUCTURES DE LEXALP 176
ANNEXE 6 IMPLÉMENTATION DE L'AFFICHAGE EN COLONNES DANS PIVAX-2 177
ANNEXE 7 SCHEMA DE LA BASE DE DONNEES LEXICALES DE LIBELLEX 178
ANNEXE 8 ALGORITHMES DE CALCUL DANS PIVAX-3 EN PSEUDO-CODE 179
ANNEXE 9 EXPERIENCE SUR LES APPELS DE TRADOH PAR SECTRA VIA ACTIVEMQ POUR LES PRETRADUCTIONS 181
ANNEXE 10 SPECIFICATIONS DE LEXTOH 193
ANNEXE 11 SPECIFICATIONS DE CREATDICO 197

Table des figures

Figure 1 : Exemple de structure "furcoïde" d'un article du dictionnaire FeM 22
Figure 2 : Vue affichée d'un article du FeM 23
Figure 3 : Architecture de Yakushite.net 28
Figure 4 : Structure et exemples de PARAX 31
Figure 5 : Interface de consultation en colonnes de PARAX 32
Figure 6 : Exemples de métadonnées et de macrostructure décrites dans SUBLIM en Lexard 32
Figure 7 : Exemple de microstructure décrite dans SUBLIM en Lingard 33
Figure 8 : Architecture à 3 couches de la plate-forme Jibiki 34
Figure 9 : Architecture de MediaWiki 35
Figure 10 : Interface de DicsToolsSuite 43
Figure 11 : Exemple d'utilisation de CDM 52
Figure 12 : Macrostructure du dictionnaire Papillon-Nadia 53
Figure 13 : Exemple de microstructure du projet Papillon-Nadia 53
Figure 14 : Structure de terminologie de LexAlp 54
Figure 15 : Un article du GDEF et sa structure Xml 55
Figure 16 : Article abandonner dans MotÀMot 56
Figure 17 : Macrostructure de Pivax et exemples de volumes 57
Figure 18 : Exemple d'un volume UNL de Pivax-1 et de ses pointeurs CDM 58
Figure 19 : Algorithme de calcul des liens dans Pivax-1 58
Figure 20 : Interface de consultation en colonnes de Pivax 59
Figure 21 : Processus de traitement du projet OMNIA 61
Figure 22 : Exemple d'annotation de texte dans OMNIA 62
Figure 23 : Microstructure de volume monolingue de CommonUNLDict 63
Figure 24 : Exemples d'articles de UWpedia 64
Figure 25 : Schéma de la base de données de Jibiki-1 66
Figure 26 : Exemple de données dans la table d'indexation 66
Figure 27 : Exemple d'utilisation de CDM-Links 67
Figure 28 : Schéma de la base de données de Jibiki-2 68
Figure 29 : Interface de liste non bornée 69
Figure 30 : Interface de recherche avec lemmatisation 69
Figure 31 : Problèmes de la ressource de V. Dikonov 73
Figure 32 : Interface d'affichage en colonnes des Pivax-2 74
Figure 33 : Interface d'affichage classique (en lignes) de Pivax-2 74
Figure 34 : Interface de consultation graphique pour la terminologie monolingue dans Libellex 76
Figure 35 : Ressource bilingue importée (avec phrases) dans la BDLex de Libellex 76
Figure 36 : Exemple de ressource prétraitée dédiée à des acronymes 77
Figure 37 : Exemple d'une ressource de Louis Vuitton prétraitée 77
Figure 38 : Exemple d'une ressource d'ExaleadSuggest prétraitée 77
Figure 39 : Structure de EDR Electronic Dictionary 82
Figure 40 : Modèle à quatre niveaux de Prolexbase 87
Figure 41 : Structure TBX standard 88
Figure 42 : Définition d'une entrée TEI 89
Figure 43 : L'interface d'import de Libellex 91
Figure 44 : L'interface d'export de Libellex 91
Figure 45 : Consultation de bleu jean sur l'interface de Libellex 92
Figure 46 : Macrostructure de Pivax-3 93
Figure 47 : Exemple des liens dans Pivax-3 94
Figure 48 : Exemple de la première microstructure des lexies 95
Figure 49 : Exemple de la deuxième microstructure (vocable > lexie) 96
Figure 50 : Exemple de la microstructure d'un volume d'axèmes 96

Figure 51 : Exemple de la microstructure des axies 96
Figure 52 : Exemple de la microstructure des prolexèmes 97
Figure 53 : Exemple de la microstructure des proaxies 97
Figure 54 : Exemple de l'utilisation d'étiquettes libres dans le volume des prolexèmes français ... 97
Figure 55 : CDM correspondant aux deux exemples de microstructure de PIVAX-3 98
Figure 56 : Exemple de calcul des liens dans PIVAX-3 100
Figure 57 : Exemple des liens de lexie Nations Unies dans la ressource lexicale 102
Figure 58 : Exemple d'axème et ses CDM 102
Figure 59 : Liens entre les axèmes et les axies 103
Figure 60 : Exemple d'axie et ses CDM 103
Figure 61 : Liens entre les prolexèmes et les proaxies 103
Figure 62 : Exemple de prolexème et ses CDM 104
Figure 63 : Liens entre les proaxies et les prolexèmes 104
Figure 64 : CDM correspondants des entrées proaxies 104
Figure 65 : Modélisation complète de l'exemple organisation des nations unies dans PIVAX-3 ... 105
Figure 66 : Terme UN de l'anglais vers toutes les langues 106
Figure 67 : Affichage agrandi de l'exemple "UN" 107
Figure 68 : Terme 国連 du japonais vers toutes les langues 108
Figure 69 : Terme onusien du français vers toutes les langues 109
Figure 70 : Modélisation de l'exemple "le pape" dans PIVAX-3 110
Figure 71 : Modélisation de BDLex pour LEXINNOVA 112
Figure 72 : Décalage de prétraductions dans SECTRA après plusieurs appels consécutifs à TRADOH 114
Figure 73 : Architecture "client- serveur" de gestion de travaux en utilisant ACTIVEMQ 118
Figure 74 : Page d'accueil de l'interface de contrôle d'ACTIVEMQ 118
Figure 75 : Architecture de LEXTOH 123
Figure 76 : Interface principale de LEXTOH 127
Figure 77 : Résultat de LEXTOH pour l'exemple utilisant ARIANE-HELOISE 128
Figure 78 : Résultat en mode avancé de LEXTOH pour l'exemple utilisant ARIANE-HELOISE 128
Figure 79 : Fichier de configuration des outils appelables 128
Figure 80 : Affichage d'un message d'erreur de LEXTOH (lemmatiseur indisponible) 129
Figure 81 : Lecture active dans Kindle 129
Figure 82 : Lecture active dans JIBIKI-JAPONAIS□FRANÇAIS 130
Figure 83 : Structure d'ARIANE-G5 et exemples d'entrées de dictionnaires 130
Figure 84 : Exemples de dictionnaires SYSTRAN 131
Figure 85 : Architecture de CREATDICO 133
Figure 86 : Procédure de demande de mini-dictionnaires par SECTRA 135
Figure 87 : Composant "Créer un script" de l'interface CREATDICO 137
Figure 88 : Mini-dictionnaire (multicible) intégré dans SECTRA 138
Figure 89 : Consultation d'un segment en "multicible", produisant une sortie simple 139
Figure 90 : Consultation d'un segment avec une seule langue cible, et sortie détaillée 140
Figure 91 : Fichier de configuration des dictionnaires appelables 140
Figure 92 : Fichier de configuration de plugin de mini-dictionnaires de SECTRA 141
Figure 93 : HowNet Interface 1 165
Figure 94 : HowNet Interface 2 165
Figure 95 : HowNet Interface 3 166
Figure 96 : HowNet Interface 4 166
Figure 97 : HowNet Interface 5 167
Figure 98 : HowNet Interface 6 167
Figure 99 : HowNet Interface 7 168
Figure 100 : HowNet Interface 8 168
Figure 101 : HowNet Interface 9 168

Figure 102 : Microstructure XML du terme "espèce protégée" 176
Figure 103 : Microstructure XML d'une axie 176
Figure 104 : Modèle de l'affichage en colonnes 177
Figure 105 : Sortie native de XIP 194
Figure 106 : Sortie native du DELAF 194
Figure 107 : Exemple d'une sortie brute 195
Figure 108 : LEMMATIX.DTD 195
Figure 109 : Deux exemples de `lemmatix intermédiaire` 196

Table des tableaux

Table 1 : Exemples pour les différents degrés de situement 45
Table 2 : Valeurs de pointeurs CDM dans différents dictionnaires 52
Table 3 : Ressources importées dans PIVAX-1 par H.-T. Nguyen 62
Table 4 : Nombre des entrées COMMONUNLDICT 71
Table 5 : Nombre des entrées UWPEDIA 71
Table 6 : Environnement des tests de PIVAX-1 et PIVAX-2 72
Table 7 : Résultat des évaluations comparatives des temps de réponse 72
Table 8 : Étiquettes utilisées pour l'exemple "Organisation des Nations Unies" 101
Table 9 : Trois niveaux de traduction : terme UN de l'anglais vers toutes les langues 106
Table 10 : Trois niveaux de traduction : terme 国連 (kokuren) du japonais vers toutes les langues 107
Table 11 : Trois niveaux de traduction : terme onusien du français vers toutes les langues 108
Table 12 : Exemple de descripteur morphologique 112
Table 13 : Comparaison des mécanismes de traitement de messages d'ActiveMQ 117
Table 14 : Expériences sur les appels de TRADOH à partir de SECTRA via ACTIVEMQ 119
Table 15 : Interfaces d'ACTIVEMQ 181
Table 16 : Structure de `message` 182

Glossaire

ACTIVEMQ	Intergiciel de gestion de tâches par messages, développé par Apache
AM	Analyse Morphologique
ANR	Agence-Nationale-Recherche
ARIANE-G5	Environnement de développement et d'exécution de systèmes (linguiciels) de traduction automatique
BDLex	Base de Données LEXicales (abréviation utilisée dans le texte)
BYU	Brigham Young University
CDM	Common Dictionary Markup
CMU	Carnegie Mellon University
CREATDICO	Intergiciel de création de mini-dictionnaires
DEC	Dictionnaire Explicatif et Combinatoire
EFPG	École Française de Papeterie et industries Graphiques
FEM	French-English-Malay
GETA	Groupe d'Étude pour la Traduction Automatique
GETALP	Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole
IMAG	Passerelle interactive d'accès multilingue (interactive Multilingual Access Gateway)
INNOVALANGUES	Innovation et transformation des pratiques de l'enseignement-apprentissage des langues dans l'enseignement supérieur
JIBIKI	Plate-forme générale pour la construction et la gestion de BDLex
KBMT	Knowledge-based MT
L&M	SAS Lingua et Machina
LIG	Laboratoire d'Informatique de Grenoble
LEXTOH	Intergiciel de lemmatisation
LN	Langue Naturelle
MCL	Macintosh Common Lisp
MSR	MicroSoft Research
NLPG	Natural Language Processing Group
OMNIA	Outils et Méthodes Numériques pour l'Interrogation et l'Analyse de textes accompagnant des images
OSLT	Observatoire de la Linguistique Sens-Texte
PAHO	Pan American Health Organization
PEG	PLNLP English Grammar
PIVAX	Base lexicale à pivot par acceptions (monolingues et interlingues) pour la mise en commun de ressources lexicales ouvertes et propriétaires pour la TA
PLNLP	Programming Language for Natural Language Processing
POS	Partie du discours (Part Of Speech)
PROLEX	Traitement automatique des noms propres et création d'un dictionnaire électronique relationnel multilingue de noms propres
PROLEXBASE	Base de la modélisation du domaine des noms propres utilisée dans le projet Prolex. Elle repose sur deux concepts centraux, le *pivot* et le *prolexème*.
SAAS	Software as a Service
SECTRA	Système d'Exploitation de Corpus de Traductions
SECTRA_W	Système d'Exploitation de Corpus de Traductions sur le Web
SGBD	Système de Gestion de Base de Données

TA	Traduction Automatique
TAL/TALN	Traitement Automatique des Langues Naturelles
TAO	Traduction Assistée par Ordinateur
THAM	Traduction Humaine Assistée par la Machine
TM/2	Translation Manager/2
TRADOH	Un outil créé en 2003 par Hung Vo-TRUNG, pour sa thèse, qui permet d'obtenir une traduction dans une langue donnée en paramètre, par mise en œuvre automatique d'un ou plusieurs systèmes de TA disponibles en local ou à distance, avec composition éventuelle.
TRAOUIERO	TRAduction : Outils Unifiés, Intégrables, Embarquables, et Ressources Opérationnelles (Projet ANR-emergence)
UDM	Université de Montréal
UGA	Université Grenoble Alpes
UJF	Université Joseph Fourier
UL	Unité Lexicale
UNL	Universal Networking Language
UNU	Université des Nations Unies (Tokyo)
USM	Universiti Sains Malaysia (Penang)
UTMK	Unit Terjemahan Melalui Komputer / Computer-Aided Translation Unit (USM)
UW	"Universal Word", terme dénotant les unités lexicales d'UNL, qui sont des "lexèmes interlingues". Exemple : `land(agt>person, obj>aircraft, goal>sea)` pour (faire) amerrir, `land(agt>person, obj>aircraft, goal>ground)` pour (faire) atterrir, et `land(agt>person, obj>aircraft)` pour poser (un engin volant).
VISULEX	Visualisation synthétique des informations Lexicales d'un système de TA développé sous ARIANE-G5

Introduction

Cette thèse a été effectuée dans l'équipe GETALP du LIG, et dans le cadre d'une bourse CIFRE avec Lingua et Machina, "une jeune société qui vise à prendre en charge la communication multilingue de l'entreprise". Il y a trois axes parallèles.

D'une part, le sujet initialement défini était centré sur l'amélioration de plusieurs aspects de génie logiciel de PIVAX, une base lexicale à pivot par acceptions monolingues et interlingues pour la mise en commun de ressources lexicales ouvertes et propriétaires pour la TA, réalisée par Hong Thai Nguyen dans le cadre de sa thèse [Nguyen, H.-T., 2009]. Il s'agissait principalement d'augmenter la vitesse et la sécurité, de passer à l'échelle, et surtout de transformer PIVAX en un vrai serveur lexical, et pour cela de proposer et d'utiliser des méthodes de génie logiciel adaptées, permettant de produire un logiciel réellement opérationnel et maintenable.

Pour améliorer PIVAX, Mathieu Mangeot a proposé d'améliorer d'abord la plate-forme sous-jacente JIBIKI. La nouvelle version JIBIKI-2 a apporté plusieurs améliorations importantes, dont PIVAX, adapté en PIVAX-2, a tout de suite hérité. Avec PIVAX-2, nous avons atteint l'objectif du passage à l'échelle. D'autre part, nous avons pu utiliser PIVAX-2 pour mettre à disposition sur le Web toutes les ressources lexicales mises dans PIVAX-2 par le projet ANR TRAOUIERO.

D'autre part, au début de cette thèse, L&M rencontrait des problèmes de gestion des acronymes. Notre première recherche a été motivée par ce besoin réel. En effet, il concerne non seulement les acronymes dans la gestion de terminologies multilingues, mais aussi l'association de plusieurs termes d'une même langue à un même référent : *Jean-Paul II* et *Karol Jozef Wojtyla* en français, ou en anglais *John Paul II* et *Karol Jozef Wojtyla*. De même, certains liens évoluent avec le temps : *le pape* désignait *Jean-Paul II* en 2004 et *Benoît XVI* en 2012. Des pays parlant la même langue (par exemple : France et Suisse romande) peuvent également utiliser des mots différents pour le même concept. Par exemple, *chien renifleur* et *chien drogue*. Inversement, le même terme peut désigner des concepts différents : dans la province de langue allemande de Bolzano en Italie, le *Landeshauptmann* est le président du conseil provincial, avec des compétences beaucoup plus limitées que le *Landeshauptmann* autrichien, qui est à la tête de l'un des États (Länder) de la fédération autrichienne. Pour la gestion des acronymes, un terme et son acronyme peuvent par exemple désigner le même référent.

Dans un contexte multilingue, la difficulté est d'établir une correspondance entre ces termes. La notion de PROLEXEME [Tran, M., 2006] présente le problème des termes ayant des acronymes dans certaines langues, mais pas dans d'autres. Dans le projet PROLEXBASE, Mickaël Tran considère le prolexème comme le regroupement de lemmes associés aux différentes formes d'un nom propre qui apparaissent dans les différents textes d'une langue donnée. Par exemple, en français, PROLEXBASE regroupe dans le même prolexème *organisation des nations unies*, *Nations unies*, *ONU* et *onusien*. En anglais, PROLEXBASE regroupe *United Nations* et son acronyme *UN*. Quelles solutions mettre en place de façon à choisir, pour un terme donné dans une langue donnée, le meilleur équivalent dans une langue cible ?

Pour trouver une solution générique, qui permette à une base lexicale "métier" de contenir tous les types d'unités lexicales, nous avons enrichi la structure de PIVAX en utilisant la notion de PROLEXEME et en créant de nouvelles notions, comme celle de PROAXIE. Nous avons pu installer le premier prototype sur JIBIKI-2 et faire des démonstrations.

Un troisième axe de recherche a été motivé par le manque de services lexicaux généraux. Il s'agit non seulement d'utilisations par des humains, mais aussi par des programmes.

L'absence d'une bonne gestion des travaux provoquait aussi beaucoup de difficultés. On ne pouvait jamais assurer le délai de temps de réponse pour un grand nombre de requêtes. C'est un besoin pratique pour presque tous nos outils. D'autre part, la conception et la réalisation de services génériques pour la lemmatisation (et autres opérations lexicales) et pour la création de mini-dictionnaires évite beaucoup de développements séparés. SansLEXTOH, par exemple, on devrait programmer un service de lemmatisation pour chaque système et chaque langue. Nous avons regroupé les aspects génie logiciel et logiciel en tant que service à la fin de ce mémoire.

Cette thèse contient quatre chapitres.

Le premier chapitre présente l'évolution des idées en lexicographie computationnelle de 1980 à 2012. Nous présentons l'état du domaine au début de cette thèse et les besoins industriels. Nous analysons les thèmes de recherche encore ouverts, et justifions le choix des thèmes sur lesquels a porté cette recherche.

Le deuxième chapitre concerne l'amélioration de PIVAX. Elle a été faite à deux niveaux : l'amélioration de la plate-forme sous-jacente JIBIKI, et l'extension du modèle de base lexicale PIVAX. Nous présentons d'abord la plate-forme JIBIKI-1. Puis, nous présentons le type de base lexicale PIVAX, avec sa macrostructure et ses microstructures, ses utilisations, et ses limitations. Ensuite, nous décrivons la nouvelle plate-forme JIBIKI-2, améliorée principalement par M. Mangeot. Nous terminons en présentant PIVAX-2, la nouvelle version de PIVAX, obtenue par adaptation de PIVAX-1 à JIBIKI-2.

Le troisième chapitre est centré sur la base lexicale "métier". Initialement motivée par un besoin industriel, cette partie de notre recherche a consisté à étudier comment construire une base lexicale couvrant tous les types d'unités lexicales, par exemple, générales, situées et terminologiques. Notre recherche s'est beaucoup inspirée de l'idée de PROLEXEME de M. Tran et de la théorie de la cognition située de E. Coseriu. Nous avons construit PIVAX-3, un prototype permettant un bon traitement des acronymes, grâce à l'utilisation des "liens riches" introduits dans JIBIKI-2.

Dans le quatrième chapitre, nous présentons nos contributions en termes de mise à disposition de supports pour des services lexicaux généraux. D'abord, nous avons utilisé ACTIVEMQ pour la gestion des travaux. Ensuite, nous avons construit l'intergiciel LEXTOH pour les services généraux de lemmatisation, et l'intergiciel CREATDICO pour les services généraux de création de mini-dictionnaires.

Chapitre I Contexte de la recherche et problèmes abordés

Introduction

Après avoir retracé l'évolution des idées en lexicographie computationnelle de 1980 à 2012, en nous limitant à ses aspects liés aux recherches en TA et TAO (Traduction Assistée par Ordinateur), nous présentons l'état du domaine au début de notre thèse, et les problèmes importants et actuels à l'époque. À partir de là, et du contexte mixte (université-industrie) dans lequel se plaçait notre thèse, nous dégageons les thèmes de recherche qui nous ont paru les plus intéressants, et sur lesquels nous avons principalement travaillé.

I.1 Évolution des idées de 1980 à 2012

I.1.1 1980— : approches visant à faire le lien entre dictionnaires pour les systèmes de TA et pour les humains

I.1.1.1 "Ouverture" des dictionnaires de TA aux traducteurs

Par rapport aux dictionnaires de TA internes et invisibles, comment "ouvrir" ou "rendre accessibles" les dictionnaires de TA aux vrais utilisateurs, et pas seulement aux développeurs de systèmes de TA, est la première étape d'interaction pour les humains.

VISULEX pour ARIANE-G5 (1982). Le système ARIANE-G5 [Guillaume, P., 1989 ; Boitet, C., 1990] est un environnement de développement pour la TAO. Un linguiste peut créer un système de TA en ARIANE-G5 sans compétences en informatique. Un système de TA sous ARIANE-G5 peut avoir jusqu'à 56 fichiers[1] de dictionnaires pour un couple de langues. Chaque fichier de dictionnaire contient un type d'information, et utilise des codes ("formats" ou "conditions") définis dans d'autres fichiers. Pour rendre accessibles ces dictionnaires de TA aux traducteurs, on a construit le système VISULEX [Bachut, D. & Verastegui, N., 1984] en 1981-1982. C'est un outil qui permet de visualiser synthétiquement les informations lexicales d'un système de TA développé en ARIANE-G5. Pour chaque unité lexicale (UL) sélectionnée, VISULEX cherche toutes les informations relatives. Il produit une "vue double" de la BDLex de ce système de TA, avec en regard les codes (des variables, des formats et des dictionnaires), et leurs commentaires.

CONVERSERFORHEALTHCARE (Spoken Translation Inc.). La société Spoken Translation Inc. a été créée par Mark Seligman en 1998 [Boitet, C. et al., 2009]. CONVERSERFORHEALTHCARE est un logiciel de TA anglais↔espagnol pour le domaine de la santé, pour les communications orales entre le personnel médical et les patients et leurs familles. Comme on a vraiment besoin de désambiguïsation lexicale, il faut rendre compréhensibles intuitivement les sens distingués dans les dictionnaires du système de TA utilisé de façon interne et codés (par des conditions sur des attributs syntaxiques et sémantiques). On le fait via des "gloses", des définitions, des icônes, et dans la dernière version en donnant un ensemble de quasi-synonymes (synset).

Idée guide 1 Établir et maintenir des correspondances étroites entre informations lexicales formelles (codées) et "naturelles" (exprimées en termes usuels).

[1] 21 dictionnaires d'analyse morphologique (phase AM : 7 pour les morphes usuels et 14 pour les tournures figées connexes), 14 pour l'expansion lexicale dans l'analyse (phases AX et AY, allant de lemmes à simples unités lexicales, unités lexicales composées et acceptions), 21 en transfert (7 pour le transfert lexical dans TL, 14 pour l'expansion lexicale en langue cible dans TX et TY), et 14 pour la génération morphologique.

Définition 1. Les **informations lexicales formelles** sont définies de manière "exacte" (ex. respecter certaines règles syntaxiques), et sont compréhensibles/lisibles par les machines.

Définition 2. Au contraire des informations lexicales formelles, les **informations lexicales naturelles** sont définies de manière usuelle, en langue naturelle, et sont orientées vers les humains.

I.1.1.2 Intégration des dictionnaires des traducteurs dans des aides à la traduction

Dictionnaires "main gauche" ("shoe box"). En 1982, Alan K. Melby (BYU, Brigham Young University) a implémenté une intégration de dictionnaires personnels "main gauche" sur les premiers micro-ordinateurs [Melby, A., 1989]. Cette aide dictionnairique était couplée très simplement à l'interface de traduction[1]. On pouvait consulter les dictionnaires en faisant un "copier/coller" d'un mot. Melby considérait cette fonction comme un support "pur" de premier niveau.

TANSACTIVE d'ALPS (Utah). Ce système a été créé par le groupe linguistique de BYU en 1980. C'est un des premiers systèmes de TA interactifs sur PC (sous CP/M). Il a une fonction de consultation des mots dans un dictionnaire bilingue pour un texte en cours de traduction. Cette fonction est accessible dans un panneau de dictionnaire en bas de l'interface de traduction. Si le mot n'est pas dans le dictionnaire, on peut aussi l'ajouter.

Système CAT de Weidner. Le système de TA de Weidner (Weidner Communications Corporation) a été créé en 1977 [Perscheid, M. M., 1985]. Il y a deux versions : une version mono-utilisateur, appelée MICROCAT, et une version pour plusieurs utilisateurs collaboratifs, appelée MACROCAT. Ce système permet la gestion de dictionnaires à distance. Une fois les textes sources entrés dans le système, celui-ci compare le résultat d'analyse morphologique de ce texte avec ses dictionnaires. Si certains mots ne sont pas dans les dictionnaires, le système les envoie aux terminologues ou aux lexicographes pour compléter leurs dictionnaires au fur à mesure du processus de traduction. Ce système permet aussi aux traducteurs, de contribuer aux dictionnaires.

TM/2. TRANSLATIONMANAGER/2 ou TM/2 [TM/2, 1992] est un système d'aide à la traduction utilisé par IBM. Il a été conçu un peu avant 1990 pour la plate-forme OS/2. Ce système utilise une mémoire de traductions en réseau pour faciliter la réutilisation des traductions existantes. Il supporte des dictionnaires préparés, ainsi que la création de dictionnaires à la volée ou en mode batch. Les dictionnaires externes en format SGML peuvent aussi être intégrés dans TM/2. Comme la plupart des utilisateurs ont besoin de leurs propres dictionnaires, et comme ces dictionnaires sont beaucoup plus adaptés à leurs exigences, IBM avait prévu de fournir ce service comme un service local. En 2002[2], IBM l'a retiré de la commercialisation, mais il a toujours été et est encore utilisé pour faire les traductions internes d'IBM, non seulement par les employés d'IBM, mais aussi par les agences de traduction externes et les traducteurs à temps partiel travaillant pour IBM. En 2010, TM/2 a été mis en source ouvert[3] et a été renommé OPENTM/2[4].

PAHO. PAHO (Pan American Health Organization) a commencé à créer ses systèmes de TA en 1980 pour le couple de langues anglais-espagnol [Vasconcellos, M. & León, M., 1985].

---

[1] Il suffisait d'appuyer à la fois sur les deux touches de majuscule, à gauche et à droite du clavier.

[2] https://transitnxt.wordpress.com/2013/11/13/what-happens-in-boblingen/

[3] "en source ouvert" et pas "en source ouverte", car c'est l'abréviation de "en source et ouvert" ou "en listing/programme source ouvert". De même, quand on achète "un série noire", on achète un roman de la série noire, et pas une série noire…

[4] http://www.opentm2.org

Les premiers systèmes (1980-1985) furent SPANAM (espagnol→anglais), puis ENGSPAN (anglais →espagnol). Beaucoup plus tard, on intégra le portugais (anglais↔portugais en juillet 2003, et espagnol↔portugais en mars 2004). Ces systèmes permettent d'importer de nouveaux termes non traduits dans les dictionnaires des systèmes de TA grâce à un outil de fusion de dictionnaires après les retours des traducteurs. C'est un des premiers systèmes reliant effectivement les dictionnaires de TA et les dictionnaires pour traducteurs.

I.1.1.3 Prototypage de dictionnaires intégrant les deux aspects

Dictionnaires furcoïdes. Les dictionnaires furcoïdes ou multicible [1] [Boitet, C. & Nedobejkine, N., 1986] sont des dictionnaires avec une seule langue source et plusieurs langues cibles. L'indexage (voir la *Définition 3*) des entrées source vers les entrées cible est monodirectionnel. En 1986-91, le GETA a construit trois dictionnaires bicible dans le domaine des télécommunications, pour le français, l'anglais et le japonais, durant le contrat DGT-KDD-GETA-Champollion. Ces dictionnaires ont été implémentés en Prolog[2] avec le SGBD (système de gestion de base de données) du commerce CLIO.

Définition 3. L'**indexage** d'une entrée source vers des entrées cible consiste à associer aux traductions possibles des conditions sur l'occurrence de l'entrée dans le contexte syntaxique ou sémantique permettant de les choisir.

Par exemple, "*manquer: X ~ de Y → X lacks Y; X ~ à Y: X misses Y*", ou "*prix [achat] → price; ~ [récompense] → prize*", ou "*layer [gen.] → couche; ~ [PAO] calque*".

EUROLANG OPTIMIZER. EUROLANG OPTIMIZER [Brace, C., 1994] est un outil d'aide à la traduction pour six langues sources et vers onze langues européennes cibles, réalisé dans le cadre du projet européen Eureka Eurolang, et présenté en 1994 par SITE/Eurolang (une filiale de iTEP/Sonovision). Cet outil est intégrable dans Microsoft Word, Framemaker et Interleaf comme un plugin et peut prétraduire un nouveau document en utilisant une mémoire de traductions et une base de données terminologiques sur serveur. Si le système ne trouve pas de résultat, il appelle un système de TA (on avait choisi LOGOS™). Le résultat fusionné est un document coloré, dans lequel les différentes couleurs représentent les correspondances parfaites (exact matches), les correspondances floues (fuzzy matches), les termes techniques connus, et les traductions obtenues par TA. Mais la société Site/Eurolang a arrêté ce système en 1995 après avoir perdu un appel d'offres de la CEE face à TRADOS.

I.1.1.4 Exemples de "bonnes pratiques" et de "cercles vertueux"

PAHOMTS. Nous avons déjà mentionné au I.1.1.2 PAHOMTS, le système de TA de la PAHO (Pan American Health Organization), entre l'anglais, l'espagnol et le portugais, basé sur l'approche transfert. Ce système est développé par des linguistes informaticiens et des traducteurs de PAHO TR (Translations Services unit), et est utilisé pour traiter plus de 90% des traductions quotidiennes [Aymerich, J. & Camelo, H., 2009] (4,5 millions de mots par an en moyenne). Grâce à sa fonction d'import de nouveaux mots par un outil de fusion de dictionnaires après les retours de traducteurs, chaque dictionnaire contient aujourd'hui plus de 150.000 mots, locutions, et règles contextuelles, dans sa dernière version 4.11, sortie en décembre 2014.

---

[1] On a bien au pluriel "multicible" et pas "multicibles", comme pour "multicanal", On parle de "langues cibles" ("cibles" est opposé à "langues"), mais de textes cible, comme de tables marron ! On a ici une élision : textes [en langue] cible.

[2] Il s'agissait de Prolog-CRISS, une version de Prolog capable de traiter les "grands caractères", codés sur 4 octets, et préfigurant Unicode et UTF-32 (mais on ne traitait que 256 systèmes d'écriture au maximum). C'est Philippe Vauquois, fils du regretté Pr. Bernard Vauquois, qui réalisa cette extension.

BehaviorTran. C'est un système de TA (anglais↔chinois) basé sur des règles de transfert, écrit en C, et développé par la NTHU (National Tsing Hua University, Taiwan) et Behavior Design Corporation à partir de 1985 [Hsu, Y.-L. U. & Su, K.-Y., 1997]. Il a commencé à être utilisé pour un service commercial en juillet 1989. Pour ses dictionnaires, il y a six niveaux différents : (1) dictionnaire général, (2) dictionnaire général de tournures connexes (ex. in order to), (3) dictionnaire de tournures non connexes (ex. turn…on), (4) dictionnaire spécialisé, (5) dictionnaire du client, (6) dictionnaire du projet. S'il y a plusieurs entrées trouvées dans des dictionnaires différents, le système utilise un ordre de priorité : le dictionnaire du projet a la plus haute priorité, puis le dictionnaire du client, ensuite le dictionnaire spécialisé, et enfin le dictionnaire général.

Pensee, Yakushite.net (Oki Electric)**.** Pensee [Shimohata, S. et al., 1999], un système de TA japonais↔anglais, a été commercialisé en 1986 pour la partie japonais→anglais et en 1988 pour la partie anglais→japonais. Sa version Web, Yakushite.net [Murata, T. et al., 2003] permet de contribuer avec des termes ou des schémas liés à des termes. C'est un des premiers systèmes de TAO intégrant directement une contribution lexicale humaine aux ressources lexicales utilisées par un système de TA. Ces données lexicales sont enregistrées dans une base lexicale, et ces dictionnaires peuvent être intégrés directement dans le système Pensee après une validation manuelle.

Systèmes de TA et THAM et leurs liaisons. Dans certains cas[1], les traductions par les mémoires de traductions sont bien adaptées. Dans d'autres cas[2], les traductions automatiques sont bien adaptées. Enfin, dans certaines situations[3], la traduction humaine est la seule bonne méthode. Il y a beaucoup de sociétés qui ont cherché à utiliser à la fois les systèmes de TA et de THAM, mais sans les intégrer au niveau des ressources lexicales. Par exemple, chez IBM, TM/2™ puis OpenTM/2™ est couplé avec LMT[4], chez EuroLang, EuroLang Optimizer est couplé avec Logos™. Les deux systèmes sont indépendants et ont très peu de liaison.

Le manque d'intégration entre ces deux systèmes est mauvais du point de vue économique [Boitet, C., 1996], et aussi pour les traducteurs, du point de vue du dictionnaire. En effet, les traducteurs peuvent faire évoluer les dictionnaires de THAM, mais pas les dictionnaires de TA. Donc, les résultats de la TA perdent leur cohérence terminologique avec les dictionnaires de THAM.

Idée guide 2 Pour les dictionnaires, la plus grande difficulté est l'incohérence entre les dictionnaires pour traducteurs, les dictionnaires pour humains, et les dictionnaires pour la TA.

Conclusion

Idée guide 3. Sans bonne liaison, ou mieux intégration, entre informations lexicales formelles[5] et naturelles[6], elles ne peuvent pas rester cohérentes, et les unes ou les autres deviennent de moins en moins utiles dans les applications impliquant une synergie homme-machine.

---

[1] Par exemple, des versions successives de fichiers d'aide en ligne, des avertissements successifs en cas de crise.

[2] Par exemple, pour les bulletins météorologiques, qui ne présentent pas une bonne répétition au niveau des phrases complètes, mais dont le vocabulaire est très limité.

[3] Par exemple, pour les nouvelles phrases d'un nouveau sous-langage, ou pour des slogans publicitaires.

[4] LMT : Logic-based Machine Translation Système, disponible depuis 1984 chez IBM [McCord, M. C., 1989].

[5] Voir la *Définition 1*.

[6] Voir la *Définition 2*.

I.1.2 1985— : tentatives pour unifier les informations générales et terminologiques

BDTAO. Les chercheurs du GETA ont étudié comment centraliser tous les types d'information lexicale dans une base lexicale dès 1986 [Boitet, C. & Nedobejkine, N., 1986]. La société B'VITAL a construit son système industriel BV/AERO/F-E et sa BDLex BDTAO en 1987. Concrètement, elle a fusionné les dictionnaires généraux monolingues et les dictionnaires terminologiques de systèmes de TA écrits en ARIANE-G5 dans une BDLex. Elle a ensuite construit (par programme) les dictionnaires du système de TA (ici écrit en Ariane) à partir de cette BDLex.

BDTAO a été conçue spécifiquement pour les systèmes de TA, et indépendamment d'un système particulier. Cette méthode a eu un succès partiel : on arrivait à produire tous les dictionnaires des systèmes de TA de type ARIANE-G5, sauf les dictionnaires de transfert lexical des unités lexicales générales. C'est le premier système commun de gestion de BDLex et il a été utilisé dans plusieurs systèmes de TA.

Ontoterminologie. Le néologisme "ontoterminologie" a été introduit en 2007 par Christophe Roche [Roche, C., 2007]. En fait, ce concept a été défini sous d'autres noms et utilisé en TA bien avant. C'est en effet une des bases de l'approche "KBMT" (Knowledge-based MT) introduite dans KBMT-89 à CMU (Carnegie Mellon University) [Nirenburg, S. & Defrise, C., 1990], puis déployée dans KANT et CATALYST pour Caterpillar [Mitamura, T. & Nyberg, E., 1992].

Définition 4. La **terminologie** [ISO 1087-1] est définie comme "l'étude scientifique des notions et des termes en usage dans les langues de spécialité".

Définition 5. Une **ontologie** (informatique)[1] est un ensemble structuré de termes et de concepts représentant les sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou par les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts.

Définition 6. Une **ontoterminologie** [Roche, C., 2007] est une terminologie dont le système notionnel est une ontologie formelle.

Ch. Roche insiste sur l'importance des principes épistémologiques qui président à la conceptualisation du modèle — c'est l'ontologie dans sa définition première. Il insiste également sur la nécessité d'une approche scientifique de la terminologie où l'expert joue un rôle fondamental — c'est l'ontologie dans ses définitions plus récentes où la logique et les langages de représentation des connaissances tiennent une place prépondérante. Enfin, une ontoterminologie met en relation le modèle conceptuel et les termes (d'usage ou normés) qui en parlent, tout en distinguant les définitions formelles des concepts (spécifications logiques) des définitions en langue naturelle des termes (explications linguistiques).

Conclusion

Idée guide 4. L'unification des informations lexicales générales et des terminologies est un besoin réel pour les systèmes de TA et THAM, et a été réalisée dès 1992 par le système opérationnel KANT/CATALYST basé sur KBMT-89.

[1] https://fr.wikipedia.org/wiki/Ontologie_(informatique)#cite_ref-3

I.1.3 1992— : dictionnairique avec évolution vers les réseaux et le contributif

I.1.3.1 1992— : travaux sur la construction de dictionnaires informatisés

I.1.3.1.1 *Le FeM*

Cette section est largement reprise d'une présentation de Ch. Boitet, directeur du projet.

a. *Lancement du projet*

Le FeM [Gaschler, J. & Lafourcade, M., 1994] est un dictionnaire monodirectionnel trilingue (français vers anglais et malais). Le but de ce projet était la création d'un dictionnaire français →malais informatisé et papier, mais on y a ajouté l'anglais, qui avait servi de "pont" initial. Il a été lancé en coopération entre le service Culturel de l'Ambassade de France à Kuala Lumpur, le Dewan Bahasa dan Pustaka (l'Institut de Langue et de Littérature, Malaisie), l'Universiti Sains Malaysia et l'équipe GETA avec l'association Champollion.

b. *Structure du dictionnaire*

Comme les dictionnaires usuels, le FeM est constituée d'une liste d'*articles*, appelés aussi *entrées*, chacun correspondant à un vocable, comme *tour n.f.* ou *tour n.m.*.

Définition 7. Un **vocable** est la forme de citation d'un lexème, accompagnée de sa classe morphosyntaxique. La **nomenclature** d'un dictionnaire est la liste de ses vocables. On appelle (micro-)structure du dictionnaire la structure générique de ses articles.

On développa d'abord une version 0 avec une structure de chaque article ("microstructure") en séquence : français-anglais et anglais-malais. Des problèmes apparurent rapidement. Chaque sens en français était souvent traduit en 2 ou 3 sens en anglais, chaque sens en anglais était éventuellement traduit en 1 ou 2 sens en malais. Quand on créa le dictionnaire français-malais en supprimant l'anglais, des doublons apparurent dans le dictionnaire.

On modifia ensuite la structure des entrées pour qu'elles mènent parallèlement et non plus en séquence du français à l'anglais et du français au malais. Le dictionnaire devint donc "furcoïde" (voir I.1.1.3). Puis on ajouta le thaï en 1997 [Lafourcade, M., 1997] et le vietnamien fin 1998. On réalisa également une version "glossaire informatique" du FeM. Voici un exemple d'un article du dictionnaire FeM.

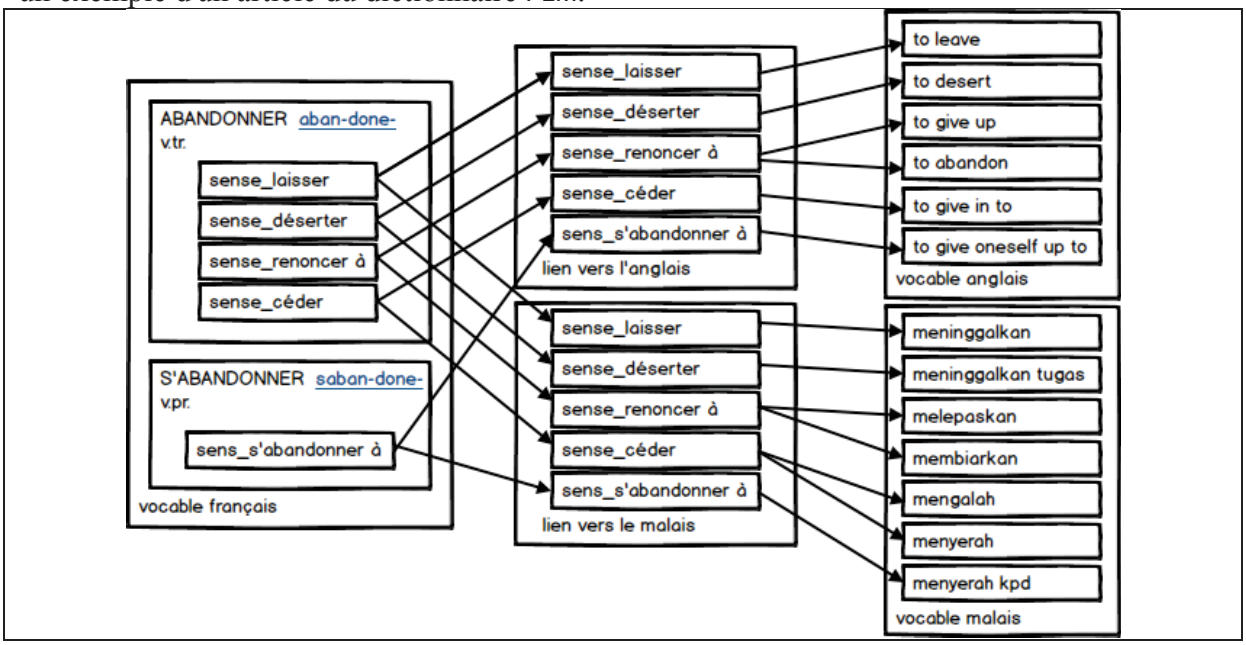
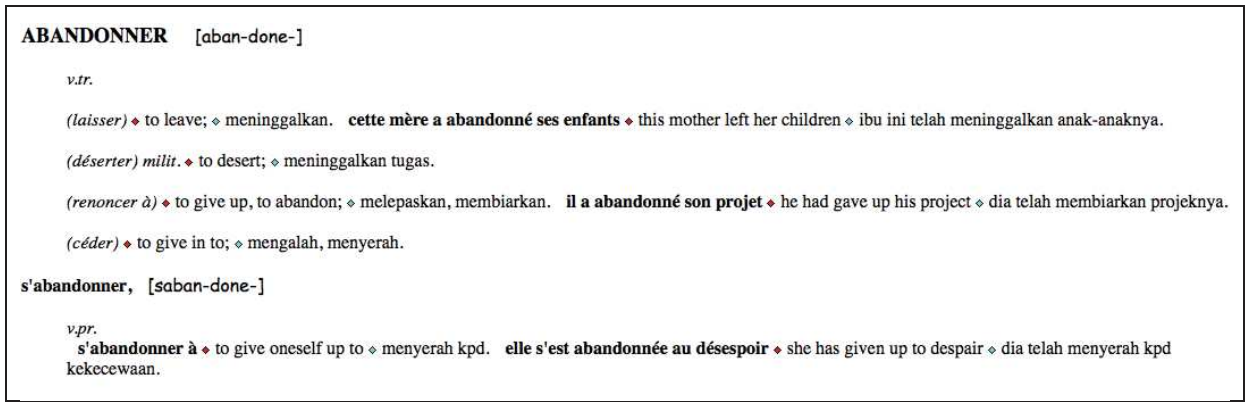


Figure 1 : Exemple de structure "furcoïde" d'un article du dictionnaire FeM



ABANDONNER [aban-done-]

v.tr.

(laisser) ◆ to leave; ◇ meninggalkan. **cette mère a abandonné ses enfants** ◆ this mother left her children ◇ ibu ini telah meninggalkan anak-anaknya.

(déserter) milit. ◆ to desert; ◇ meninggalkan tugas.

(renoncer à) ◆ to give up, to abandon; ◇ melepaskan, membiarkan. **il a abandonné son projet** ◆ he had gave up his project ◇ dia telah membiarkan projeknya.

(céder) ◆ to give in to; ◇ mengalah, menyerah.

s'abandonner, [saban-done-]

v.pr.
s'abandonner à ◆ to give oneself up to ◇ menyerah kpd. **elle s'est abandonnée au désespoir** ◆ she has given up to despair ◇ dia telah menyerah kpd kekecewaan.

Figure 2 : Vue affichée d'un article du FEM[1]

c. *Méthode de construction du dictionnaire*

La première méthode de construction a été proposée et implémentée par Ch. Boitet à Penang en 1991 en une semaine environ. C'était une BD en 4D[2] avec les interfaces en français, anglais et malais. Mais 4D ne fonctionnait que sur Mac, et pas sur Windows, ce qui ne convenait pas aux lexicographes de l'UTMK[3] (Universiti Sains Malaysia, Penang). D'autre part, l'interface "BD" avec une fiche par mot ne convenait pas pour les articles "généraux" à structure complexe.

Voyant cela, il proposa et mit en place deux semaines plus tard l'utilisation de Microsoft Word™ comme "éditeur pseudo-syntaxique", pour plusieurs raisons : (1) il fonctionnait sur Mac et sur PC, (2) les lexicographes le connaissaient déjà, (3) il était déjà installé sur toutes les machines, (4) pour les traitements informatiques, on pouvait travailler sur des fichiers RTF (Rich Text Format) équivalents. La méthode consiste à prédéfinir des styles (de paragraphe) particuliers pour les éléments logiques (mot-vedette, prononciation etc.). Il est ensuite assez facile d'analyser les fichiers RTF et d'importer le contenu dans une base de données (Postgres).

En pratique, la construction du FEM a consisté à travailler en pipeline sur une trentaine de fichiers (1 ou 2 par lettre initiale), puis à consolider le tout dans la BDLex. Ensuite, on a produit à partir de la BDLex (1) les fichiers Word, un par lettre, utilisés pour "remplir" les gabarits PageMaker du dictionnaire papier final, et (2) les données des outils informatiques de consultation.

d. *Consultation du FEM : version imprimée et versions électroniques*

En 1996. La première version de l'outil ALEX 1.0, développée par M. Lafourcade [Lafourcade, M., 1996] a été livrée en mai 1996. Cet outil permettait non seulement la consultation par les entrées, mais aussi par le contenu des articles. Il permettait également de filtrer les informations, de les trier et de les compter, etc. Mais cet outil, programmé en MCL (Macintosh Common Lisp), ne fonctionnait que sur Mac.

Une version électronique en MICROSOFT WINHELP a été développée pour Windows (par Ee Churn, une informaticienne de l'UTMK). C'est une version basique de consultation.

Une version imprimée (le dictionnaire français-malais "Kamus Perancis-Melayu Dewan") a été publiée à Kuala Lumpur [Gut, Y. et al., 1996] en juillet 1996. Dans ce dictionnaire, on a supprimé l'anglais et bien révisé et complété le malais avec l'aide de spécialistes.

---

[1] Dans la vue imprimée, la transcription phonétique est en API (Alphabet Phonétique International).

[2] 4D : un système commercial de gestion de base de données. http://www.4d.com/fr/

[3] UTMK : Unit Terjemahan Melalui Komputer / Computer-Aided Translation Unit

En 1998. L'anglais a été complété par P. Lafourcade. Pour cette version, G. Sérasset a développé une application en Java. Cette application fonctionne sur tous les systèmes, mais ne permet que la consultation.

Vers 1999-2000. M. Lafourcade a développé une version sur Internet, WALEX.

En 2003. La partie malaise des entrées a été révisée encore une fois, cette fois-ci non pas par le DBP, mais par L. Metzger et la Maison du Monde Malais (MMM, La Rochelle), et on a intégré cette version dans la BDLex du projet PAPILLON.

I.1.3.1.2 Informatisation du DEC

Le dictionnaire DEC (Dictionnaire Explicatif et Combinatoire) [Mel'čuk, I., 1992] est un dictionnaire très complexe. En novembre 1994, dans le cadre d'une collaboration post-doctorale avec le RALI (UdM)[1] et l'OLST[2], Gilles Sérasset a défini un projet visant à informatiser le DEC et à mettre à disposition des lexicologues des outils simplifiant la création d'articles du DEC [Sérasset, G., 1996]. Ce projet a été mené à bien durant plusieurs séjours d'été à Montréal.

Il y avait une version disponible sous forme de fichiers Word correspondant à la version imprimée du DEC. Hai Doan-Nguyen a participé avec son outil RECUPDIC[3] à la récupération de ces ressources lexicales [Doan-Nguyen, H., 1998b]. En fait, il a fait cette opération deux fois. La première fois, en 1995, il a travaillé sur la forme RTF, comme M. Lafourcade l'avait fait pour le FEM. Il a donc écrit en H-GRAMMAR un premier "récupérateur" travaillant sur cette forme, et produisant une représentation des articles comme des objets Common Lisp (LISPO). Ça a marché moyennement bien, car la forme RTF était trop compliquée, avec des balises parasites[4] qu'il a passé beaucoup de temps à corriger.

La seconde fois, en 1996, il l'a récupéré avec une approche différente, qui s'est révélée nettement plus efficace et plus rapide. Il a d'abord utilisé une macro VISUALBASIC pour transformer et normaliser le fichier, de façon à ce qu'il ne contienne plus que des caractères normaux (ISO-646). Par exemple, il a marqué les définitions avec `§§D`, les chaînes en indice par `#[d` et `#]d`, et de façon similaire les chaîne en gras, en exposant, les symboles spéciaux, etc. À ce point, il a fini la normalisation à la main, bien plus rapidement que sur du RTF. Il a écrit en H-GRAMMAR un nouveau récupérateur, beaucoup plus simple, travaillant sur cette forme normalisée en texte brut, et obtenu un résultat un peu meilleur que la première fois, et beaucoup plus vite. Il a enfin utilisé PRODUCDIC[5] pour convertir le dictionnaire récupéré vers le format de DECID.

DECID est un véritable éditeur spécialisé, créé par G. Sérasset, et écrit lui aussi en Common Lisp[6] [Sérasset, G., 1996]. L'interface de DECID est directement inspirée de la version papier

---

[1] Groupe de recherche en automatisation de la linguistique, Université de Montréal. Guy Lapalme a activement participé à ce projet, pendant ces séjours, et ensuite lors d'une année sabbatique à Grenoble.

[2] Observatoire de la Linguistique Sens-Texte, créé par Alain Polguère.

[3] Système de récupération de ressources dictionnairiques, développé par Hai Doan-Nguyen dans le cadre de sa thèse. Il contient l'outil H-GRAMMAR, qui st un générateur d'analyseur d'articles de dictionnaire à partir de leur structure, définie par une grammaire hors-contexte.

[4] Par exemple, un blanc accidentellement mis en gras dans un exemple en style normal est invisible sous Word mais provoque un ajout de balises inattendues quand on sauve le fichier en RTF.

[5] PRODUCDIC, basé sur un langage de type ensembliste, est l'outil qu'il a créé pour transformer des dictionnaires en LISPO en d'autres dictionnaires également en LISPO. Il a ainsi inversé et composé des dictionnaires.

[6] Il s'agit de MCL (Macintosh Common Lisp).

du DEC. Cet éditeur, facile à utiliser, et par construction très bien adapté au DEC, a servi plusieurs années à l'OSLT.

I.1.3.1.3 Problème de l'asymétrie et introduction de quelques termes

Pour tous les types de dictionnaire de traduction (*Définition 8*), il y a un problème qui n'a jamais été résolu. C'est le problème de la non-symétrie : on ne peut pas inverser la source et la cible d'un volume (*Définition 9*) pour obtenir un nouveau volume de même qualité allant de la langue cible vers la langue source. Le problème vient de la différence de niveau d'information entre source et cible(s). Cela crée un "bruit" énorme, que ce soit en inversion (ex. fr→zh ⊢ zh →fr) ou en composition (ex : fr→en + en→my ⊢ fr→my).

Définition 8. Un **dictionnaire de traduction** est composé d'un ou plusieurs *volumes,* chacun relatif à une des langues considérées.

Définition 9. Un **volume** d'un dictionnaire est un ensemble d'articles, accédés par des mots-vedettes (lemmes en général), décrivant des mots d'une même langue. Un article d'un dictionnaire de traduction contient les traductions du mot identifié dans la ou les langues cibles considérées.

Définition 10. Un **article de dictionnaire** comporte au moins le mot-vedette, et le plus souvent d'autres informations (prononciation, classe grammaticale, définition, gloses identifiant des sens, ou *lexies,* exemples, etc.), ainsi que des traductions s'il s'agit d'un dictionnaire de traduction.

Idée guide 5. La non-symétrie des dictionnaires de traduction classiques est une difficulté majeure.

I.1.3.1.4 Travaux de MSR[1] (équipe venant d'IBM)[2]

En 1978, Gorges Heidorn du groupe d'IBM de Yorktown-Heights créa le langage spécialisé PLNLP (Programming Language for Natural Language Processing), à l'aide duquel le groupe construisit une grammaire de l'anglais extraordinairement couvrante pour l'époque, PEG (PLNLP English Grammar) [Jensen, K., 1986][3]. La grammaire PEG a été utilisée dans les systèmes EPISTLE puis CRITIQUE destinés à la correction orthographique, grammaticale, terminologique, phraséologique et stylistique, pour les documents d'IBM. En parallèle, cette équipe a mené un projet de longue haleine avec l'université de Lancaster pour la production d'un grand corpus arboré, le LANCASTER/IBM TREEBANK. Les linguistes de Lancaster utilisaient un éditeur d'arbres pour choisir le meilleur arbre parmi les résultats de l'analyseur PEG.

Au début des années 80, ce groupe a aussi utilisé PEG pour développer avec IBM-Japon un système de TA vers le japonais, SHALT-1. Pour obtenir des structures logico-sémantiques rendant la traduction vers le japonais moins lourde, ils ajoutèrent à PEG un module supplémentaire, dit "relationnel". SHALT-1 a été utilisé (et l'est peut-être encore) pour traduire en interne les documentations techniques d'IBM, en enchaînant TA, post-édition et révision.

Au printemps 1991, tout ce groupe quitta IBM. S. Richardson, G. Heidorn, and K. Jensen (au moins) fondèrent le NLPG (Natural Language Processing Group) à Microsoft Research. Bill

[1] MSR : MicroSoft Research

[2] Le texte de cette sous-section provident d'une présentation interne de Ch. Boitet, à la quelle j'ai ajouté de nombreuses précisions.

[3] PEG pouvait analyser de très nombreux types de texte, du Wall Street Journal aux menus des restaurants chinois en passant par les documents techniques d'IBM ! La grammaire d'ETAP-3 (IPPI, Moscou), construite par l'équipe d'Apresyan, Boguslavskij et Iomdine, depuis 1977 environ, est encore plus couvrante et précise.

Dolan et Lucy Vanderwende y arrivèrent en 1992. Ezra Black rejoignit ATR (Nara, Japon) où il créa un nouvel analyseur qui fut la base du projet ATR/LANCASTER TREEBANK[1]. Pour des raisons contractuelles, le groupe du NLPG dut attendre 4 ans avant de pouvoir travailler de nouveau sur l'analyse, la correction et la TA. Mais G. Heidorn créa dès 1992 G, une version améliorée de PLNLP.

En 1991, le tout nouveau NLPG commença à développer MINDNET [Dolan, W. et al., 1993 ; Richardson, S. D. et al., 1998], une structure de données destinée à l'acquisition et au stockage de connaissances sémantiques, sur laquelle L. Vanderwende fit ensuite une bonne partie de sa recherche. Une des premières idées fut d'extraire les connaissances sémantiques des dictionnaires informatisés, et pour cela de commencer à les transformer en une base de connaissances lexicales. Il apparut que MINDNET serait tout à fait adapté. MSR se procura les droits d'utilisation de la version électronique de l'AHD3 (American Heritage Dictionary, 3nd Edition), et le NLPG le transforma en un *réseau lexical* (de type réseau de Hopfield) implémenté en MINDNET. Les nœuds correspondent aux divers éléments des articles (mot-vedette, glose, sens, définition…) et les arcs portent des poids positifs pour les attractions (par exemple entre une définition et les mots qu'elle contient) et négatifs pour les répulsions (par exemple entre deux sens séparés).

Pour désambiguïser les sens des mots apparaissant dans les définitions, K. Jensen écrivit en G un parseur des définitions du dictionnaire. Cela permit de lever la plupart des ambiguïtés syntaxiques (sur les parties du discours, ou *POS*). Pour la désambiguïsation sémantique, le NLPG utilisa un algorithme de type recuit simulé *(simulated annealing)*, ce qui permit de relier chaque mot d'une définition au nœud représentant son sens dans le réseau.

Quand l'analyseur fut plus avancé et permit d'analyser des phrases complètes et plus seulement des définitions, cette méthode fut adaptée à la désambiguïsation lexicale de phrases, sous le nom de *lexical priming* (amorçage lexical). On analyse la phrase à traiter, on relie les nœuds lexicaux de l'arbre obtenu (une *logical form)* aux nœuds de ces mots dans le MINDNET, ce qui l'étend temporairement, on les "chauffe", puis on laisse "refroidir". Les nœuds des sens (lexies) les plus probables dans le contexte de la phrase sont les plus "chauds" [Dolan, W. B. & Richardson, S. D., 1996].

Ensuite, le NLPG fit de même pour le LDOCE (Longman Dictionary of Contemporary English), et unifia ces deux dictionnaires dans un seul réseau lexical implémenté en MINDNET.

À partir de 1996, le NLPG se mit à travailler sur la TA. Il développa des parseurs (dits plus tard "experts") et des générateurs en G pour 7 ou 8 autres langues. Pour le transfert, n'ayant pas de dictionnaires de traduction disponibles, ni d'outils pour écrire des transferts structuraux complexes, ses membres se tournèrent vers l'apprentissage automatique. En analysant les traductions de leurs documents techniques (faites par des professionnels), ils obtinrent de gros corpus d'arbres (dits *logical forms,* ou formes logiques) alignés, de type <LF_eng, LF_fra>, <LF_eng, LF_esp>, etc., à partir desquels on peut automatiquement apprendre la phase de transfert (lexical aussi bien que structural !).

Au moyen de calculs statistiques (information mutuelle, etc.), on compile, pour chaque paire de langues, une structure MINDNET, dans laquelle sont stockés des morceaux des arbres source (arbrelets, ou *treelets*), auxquels sont associés les morceaux les plus probables de leur correspondant en langue cible.

Pour traduire une phrase, on l'analyse, on cherche (par un algorithme de type Viterbi) une meilleure couverture de l'arbre obtenu, par des arbrelets source, et on effectue une descente

[1] http://kmcs.nii.ac.jp/nlp_annual/nlp1996-B7-02/ye=1996&ys=1996

récursive de l'arbre source en construisant un arbre cible avec les arbrelets cible. Si l'arbre cible obtenu n'est pas une forme logique cible bien formée, on le corrige avec quelques règles *ad hoc* (expertes, donc). On utilise ensuite le générateur de la langue cible et on obtient une traduction. Le système de TA obtenu est donc *hybride.*

Entre 1998-2001, MSR présenta ce système à plusieurs reprises. Ensuite, comme Microsoft traduit vers au moins 45 langues et ne peut pas développer des analyseurs et générateurs couvrants et corrects pour tant de langues, le NLPG s'est tourné vers l'apprentissage direct des deux phases successives de transfert et de génération. On aligne donc chaque LF_eng avec la traduction correspondante, sous forme d'une suite de mots ou de vecteurs (liste de propriétés, dont le lemme, la classe, le nombre, le mode, le temps…). L'étape d'analyse reste experte (avec un choix probabiliste), ainsi que l'étape de génération morphologique (si on aligne avec des suites de vecteurs), et le "transfert descendant" est appris par des méthodes statistiques.

Notons pour finir que, dans tous les cas, on peut si on le souhaite intégrer dans le système un dictionnaire bilingue appris à partir des alignements, ou bien construit manuellement ou semi-automatiquement.

I.1.3.1.5 Versions électroniques de dictionnaires commerciaux

Il y a beaucoup de dictionnaires commerciaux (par exemple, Larousse, Petit Robert, Oxford etc.) qui ont été produits sous forme électronique à partir de bases de données. Ils sont construits, depuis près de 20 ans, en utilisant des outils d'exploration de grands corpus. Par exemple, à l'aide de concordanciers, les éditeurs des dictionnaires comme Collins, Longman, Larousse, etc. cherchent la fréquence des expressions et de leurs usages dans différents sens dans des corpus récents et variés [Monteleone, M., 2003].

Grâce à Internet, il y a beaucoup de dictionnaires consultables en ligne[1], par exemple, Reverso de Softissimo[2], Cambridge Dictionaries[3] etc., et plus récemment, sur les téléphones mobiles. Ces dictionnaires sont uniquement consultables, mais presque jamais téléchargeables[4]. Nous avons ajouté une liste de ressources informatisées et téléchargeables en annexe (Annexe 1).

I.1.3.2 1995 ou 1998 : vers la construction contributive de dictionnaires en ligne

Deux ou trois ans après la naissance d'Internet (début des années 1990), on a commencé à étudier les apports possibles des contributions en ligne à un dictionnaire.

I.1.3.2.1 YAKUSHITE.NET (OKI electric)

YAKUSHITE.NET a été introduit au I.1.1.4. Le but de cet environnement est de faire construire les dictionnaires du système de TA PENSEE de façon contributive par des traducteurs humains, via Internet. Les dictionnaires sont organisés de façon hiérarchique en dictionnaires et sous-dictionnaires.

Le service Web *YAKUSHITE.NET* a été mis en service le 18 septembre 2001. Toutes les fonctions sont directement intégrées dans le système PENSEE grâce à un module spécial. Le service Web permet la traduction, la post-édition, la gestion de dictionnaires et la gestion de communautés (gestion des contributeurs, propagation des bulletins et service de FAQ) [Shimohata, S. et al., 1999].

---

[1] Voir [OneLook, 2016] pour une liste de dictionnaires consultables complémentaires, principalement sur l'anglais.

[2] http://www.reverso.net/text_translation.aspx?lang=FR

[3] http://dictionary.cambridge.org/

[4] Nous avons vérifié les 50 premiers dictionnaires (dans la liste triée alphabétiquement des dictionnaires consultables en ligne), et aucun n'est téléchargeable.

L'architecture de YAKUSHITE.NET est très intéressante. Il y a deux niveaux de dictionnaire : un SGBD pour les utilisateurs, et un serveur spécial de dictionnaires pour le système de TA PENSEE. Voir la *Figure 3*. Après une contribution d'un contributeur, le résultat est stocké dans les deux dictionnaires.

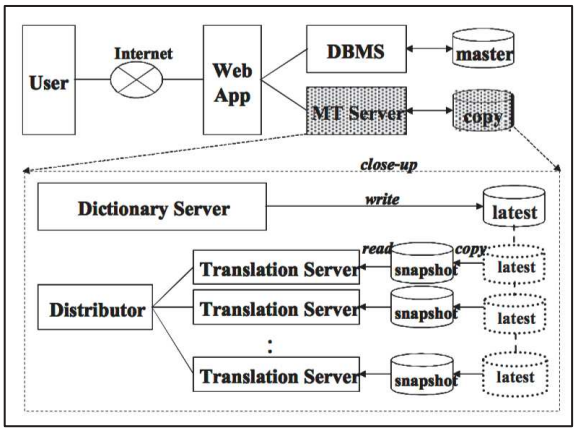


Figure 3 : Architecture de YAKUSHITE.NET

OKI a arrêté le service Web YAKUSHITE.NET en octobre 2014 à cause du manque de ressources financières. En effet, YAKUSHITE.NET était sponsorisé par une société de services de traduction, intéressée uniquement par certains domaines spécifiques. Après sa disparition, les services ont été arrêtés.

I.1.3.2.2 Projet UNL

En décembre 1996, le projet UNL fut lancé par l'IAS (Institute of Advanced Studies) de l'UNU (Université des Nations Unies) à Tokyo. Le directeur et concepteur d'UNL est Hiroshi Uchida. Le but de ce projet était de réaliser les communications multilingues pour que tout le monde puisse surmonter les barrières linguistiques à un horizon de 10 ans.

Il s'agit de développer un langage intermédiaire pivot, puis de traduire rapidement entre ce langage intermédiaire et chaque langue naturelle par un enconvertisseur et un déconvertisseur.

Les trois premières années (1997-1999) ont été utilisées pour finaliser les spécifications du langage intermédiaire UNL, faire des développements informatiques pour les 12 langues sélectionnées (arabe, chinois, espagnol, français, hindi, indonésien, italien, japonais, portugais, russe, swahili[1] et thaï) avec 14 partenaires, et, à cause de la présence à l'UNU d'un doctorant mongol, une étude seulement préliminaire pour le mongol. L'anglais servait de langue de travail. Le vocabulaire d'UNL est d'ailleurs volontairement basé sur l'anglais, mais il n'y avait pas de groupe UNL venant d'un pays anglophone. C'est le centre de Tokyo qui a le premier réalisé un déconvertisseur et un enconvertisseur de l'anglais. Le centre de Moscou (IPPI) en a ensuite réalisé une autre version dans ETAP-3, comme sous-produit de la construction du déconvertisseur et de l'enconvertisseur du russe.

En 2001, la fondation UNDL a été établie à Genève pour promouvoir l'application réelle d'UNL. Les outils ENCO et DECO sont fournis par UNDL, pour créer les enconvertisseurs et déconvertisseurs. Plus précisément, pour chaque enconvertisseur et déconvertisseur, on a besoin d'un dictionnaire d'UW (Universal Words) ↔ LN (langue naturelle) et d'un ensemble de règles (écrites en ENCO et en DECO).

Pour unifier la liste des UW, l'UNL-C (UNL Center, Tokyo) a développé l'outil UWGATE. Cet outil permet de manipuler la BD des UW via Internet. Avec UWGATE, on peut exporter un dictionnaire UW-LN vers un fichier ayant le format attendu par ENCO et DECO. Il offre une application graphique sur Windows à distance, appelé UWGATEPLUS. Malheureusement, UWGATE marchait mal. En fait, il était mal conçu (échanges seulement par courriel) et beaucoup trop lent (et assez souvent, on n'avait aucun retour).

[1] En fait, le professeur invité au séminaire de lancement d'UNL en décembre 1996 est reparti avec un PC sur lequel UNL-jp avait mis des outils et des ressources… puis on n'a plus jamais entendu parler de lui. Rien n'a été fait sur le swahili depuis lors.

Le GETALP a lancé en 2011 le projet GBDLEX-UW++ dans le cadre du projet ANR TRAOUIERO et du consortium UNL++ (lui-même lancé en 2004 lors du colloque LREC à Lisbonne, en coopération avec l'IPPI (le centre de langue du russe) et l'UPM (le centre de langue de l'espagnol). Ce projet a pour but d'unifier les dictionnaires des UW de différents groupes (voir l'Annexe 2) et de construire pour de multiples usages une grande base lexicale multilingue (en au moins 4 langues et UNL). Nous le présenterons en détail au II.4.3.

I.1.3.2.3 Projet PAPILLON

On donne en Annexe 3 un historique assez détaillé du projet PAPILLON. Deux points importants sont qu'il a produit deux bases lexicales en ligne, de taille et d'ambition lexicologique très différentes, et que G. Sérasset et M. Mangeot en ont tiré une plate-forme générique de construction de BDLex multilingues en ligne, JIBIKI.

La première base lexicale, PAPILLON-CDM, est une version opérationnelle depuis plus de 15 ans de la BDLex multilingue créée par M. Mangeot pour sa thèse vers 2000. Grâce à un ensemble de balises XML appelé CDM (Common Dictionary Markup), il était arrivé à "récupérer" des dictionnaires informatisés classiques (monolingues ou multilingues furcoïdes) quelconques, pourvu qu'ils soient en XML, et à les présenter de façon homogène à travers une interface Web unique[1]. Très rapidement, on y intégra de nombreux dictionnaires mis en "source ouvert" par leurs auteurs (le GETA et l'USM pour le FEM et le dictionnaire russe→français tiré du système de TA RUS-FRA, Jim Breen pour JMDICT, avec ses 90000 entrées en japonais→ anglais, Ulrich Apel pour ses 200000 ou 250000 articles en japonais↔allemand, J-Y. Desperrier pour la traduction en japonais→français de 15000 articles du JMDICT, Ho Ngoc Duc pour ses 41000 et 39000 articles en français↔vietnamien, etc.).

En 2016, PAPILLON-CDM contient 2 millions d'articles concernant les langues : allemand, anglais, français, japonais, malais, vietnamien, chinois, russe, peul, wolof et fas. Les articles concernant des langues africaines viennent du projet DILAF, mené depuis 2008 par M. Mangeot.

On peut consulter plusieurs dictionnaires à la fois, avec lemmatisation de la forme recherchée, télécharger tout ou partie de la base, sous forme XML, et y contribuer par une interface générée automatiquement à partir de la structure CDM.

La seconde base lexicale, PAPILLON-NADIA, a été prévue comme une version structurée selon l'architecture générale NADIA proposée dans la thèse de G. Sérasset (un volume monolingue pour chaque langue, où chaque article correspond à une *lexie* (*Définition 11*), et un volume central d'acceptions interlingues ou *axies* (*Définition 12*). Pour chaque volume, on voulait adopter une structure conforme à la lexicologie explicative et combinatoire d'Igor Mel'čuk qui a défini la notion de lexie. On choisit en 2000 non pas la structure du DEC, jugée trop complexe, mais la structure générique DICO de Mel'čuk et Polguère [Mel'čuk, I. & Polguère, A., 2006], qui en est une simplification et était en cours d'implémentation sous FRAMEMAKER.

Les schémas XML des volumes monolingues de type DICO et du pivot (acceptions interlingues ou axies) ont été écrits en 2001 (puis améliorés), et un petit nombre d'articles en français, japonais et anglais a été mis dans ces formats pour validation.

Définition 11. Une **lexie** est un sens de mot dans un dictionnaire.

Définition 12. Une **axie** est une classe d'équivalence de lexies synonymes.

[1] En bref, la technique consiste, étant donné un dictionnaire à importer dans la base, à définir un XPATH pour chaque balise CDM.

I.1.3.2.4 Projet ITOLDU

ITOLDU [Bellynck, V. et al., 2005] (Industrial Technical On Line Dictionary for Universities) est un site pédagogique destiné à faire développer par les étudiants, dans le cadre de leur apprentissage de l'anglais, des dictionnaires techniques anglais-français, dans le cadre de l'enseignement de l'anglais à l'EFPG (École Française de Papeterie et industries Graphiques, renommée PAGORA en 2008). Ce projet a été lancé par V. Bellynck et J. Kenwright et a fonctionné durant 3 ans (2003-2006). Il a permis la construction collaborative d'un dictionnaire terminologique anglais→français de microstructure très simple : terme-en, terme-fr, domaine, exemple d'usage.

Selon V. Bellynck, l'un des objectifs était de "motiver les étudiants à participer à la constitution d'un savoir commun en valorisant l'effort fourni, et d'éviter une surcharge de travail à l'enseignant".

ITOLDU a été un vrai succès du point de vue des contributions, sans doute parce que le travail avec cet outil donnait 1/3 de la note d'anglais. Pendant l'année scolaire 2003-2004, on a obtenu 17020 entrées bilingues (français et anglais) avec environ 250 étudiants répartis en 12 classes, et 5 professeurs d'anglais. Les professeurs pouvaient évaluer les contributions des étudiants via une interface réservée aux professeurs. Les équivalents étaient corrects à 90%. Par contre, les exemples d'usage n'étaient bons qu'à 60% environ, et cela parce que cette notion avait été mal expliquée aux étudiants. Cela fut amélioré en 2004-05 (on obtint environ 6000 nouveaux articles, avec 90% et 85% de qualité).

Le problème rencontré en 2005-06 fut que les étudiants se mirent à "détourner" l'environnement, par exemple en allant juste après un cours monopoliser les mots de la "chasse aux mots" proposée durant le cours, au détriment des autres étudiants. Il aurait fallu avoir des ressources supplémentaires pour développer une version permettant aux professeurs d'anglais de définir eux-mêmes de nouvelles activités pédagogiques et des stratégies pour les contrôler et les évaluer.

La ressource produite par le projet ITOLDU a récemment été intégrée à PIVAX-2, voir II.4.2.1.

Idée guide 6. Une base lexicale contributive ne peut bien fonctionner que si (1) on peut motiver les contributeurs, et (2) un ou des animateurs (non-informaticiens) peuvent définir de nouvelles tâches et organiser puis contrôler eux-mêmes le fonctionnement de l'outil.

I.1.4 1991—: évolution vers des bases lexicales

I.1.4.1 Bases lexicales permettant la symétrie

On a posé le problème de la non-symétrie dans les dictionnaires au I.1.3.1.1. À cause de ce problème, on ne peut pas inverser ou composer des dictionnaires classiques en conservant leurs qualités. À partir des années 1990, on a construit des BDLex symétriques grâce aux notions de lexie (sens de mot dans un dictionnaire) et d'acception (sens de mot en usage).

Définition 13. Une **acception** est un ensemble de lexies synonymes. On appelle **axème** une acception monolingue, et **axie** une acception interlingue (pouvant regrouper des lexies de différentes langues).

Par exemple, le mot français *bleu* correspond à plusieurs lexies : *bleu_nm#couleur*, *bleu_nm#fromage*, *bleu_nm#contusion*, *bleu_adj#couleur*, *bleu_adj#cuisson*, etc. Un

[1] http://www.postgresql.org/

[2] http://enhydra.ow2.org/about.html

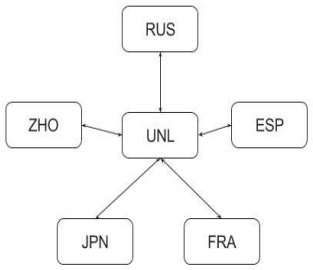
axème pour le sens de résultat d'un choc pourrait être *axème_n#18903*, regroupant la lexie *bleu_nm#contusion* et la lexie *ecchymose_nf#contusion*.

Ces idées ont été introduites par Étienne Blanc, linguiste-informaticien, quand il créa la BDLex PARAX [Blanc, E., 1999]. Le but était de créer une base de données lexicales multilingues à acceptions interlingues basée sur HYPERCARD[1]. Depuis, il l'a convertie en REVOLUTION[2], puis en LIVECODE. Dans cette BDLex, il y a un dictionnaire pour l'espace lexical (*Définition 14*, reprise ici telle quelle de la thèse de H.-T. Nguyen) d'UNL, qui contient des UW.

Définition 14. On appelle "**espace lexical**" d'une langue l'ensemble structuré de ses unités, à des niveaux de plus en plus abstraits ou génériques : formes, lemmes, racines (dans certaines langues) familles dérivationnelles plus ou moins productives, prolexèmes (réunissant par exemple *US*, *USA* et *Etats-Unis*), et "acception" ou "sens de mots"... sans oublier qu'il y a à tous ces niveaux des unités simples et des unités complexes (mots composés, lexies ou acceptions complexes).

Au début du projet UNL, l'équipe de Hiroshi Uchida à Tokyo avait créé environ 100K UW, à partir des ressources de projets précédents (essentiellement, EDR). Chaque groupe en a ensuite créé de nouvelles, en fonction des nouveaux mots à traiter dans sa langue. Dans PARAX, il y a 5 volumes monolingues (français, japonais, chinois, espagnol et russe). Chaque volume de langue contient de 30K à 60K entrées. Chaque mot est relié à un ou plusieurs sens (lexies), et chaque sens est relié à un UW.

Dans le dictionnaire français, les entrées contiennent une description morphosyntaxique assez complète, utilisée pour construire les dictionnaires du français de systèmes de TA du labo. Cela n'a pas été fait pour les autres langues.



Exemple pour FRA
[biens] {CAT(CATN),GNR(MAS),NUM(PLU)} "goods(icl>functional thing)"; [catalogue] {CAT(CATN),GNR(MAS),N(NC)}"catalog(agt>thing,obj>thing)"; [centaines] {CAT(CATN),N(NP)} "hundreds of";
Exemple pour ESP
[necesidad] {VAR} "need(icl<abstract thing)"; [necesidad] {VAR} "need(icl>thing)"; [nivel] {VAR} "level(icl>thing)"; [no ambiguo] {VAR} "unambiguous";

Figure 4 : Structure et exemples de PARAX

[1] HYPERCARD est un programme et un environnement de programmation développé par Apple qui ne fonctionne que sous Mac OS versions 9 et précédentes.

[2] REVOLUTION est un descendant d'HYPERCARD, porté sur les plates-formes usuelles (MacOS, Unix, Windows), et s'appelle désormais LIVECODE.

Figure 5 : Interface de consultation en colonnes de PARAX

I.1.4.2 Structure générique (microstructures et macrostructures)

Chaque système de gestion de BDLex est basé sur des connaissances en linguistique, en lexicographie et en informatique. Les structures d'organisation de dictionnaires sont donc hétérogènes. Dans sa thèse, G. Sérasset [Sérasset, G., 1994b] a construit SUBLIM, un système universel de BDLM (Base de Données Lexicales Multilingue). Dans ce système, on décrit la structure d'une BDLex en utilisant deux langages, LEXARD et LINGARD. Ces deux langages sont réalisés en CLOS (Common Lisp Object System)[1].

Le langage LEXARD permet d'écrire des descriptions à deux niveaux : (1) le niveau des *métadonnées* telles que le type de dictionnaire (monolingue, bilingue unidirectionnel, bilingue bidirectionnel ou interlingue), la langue du dictionnaire, le propriétaire du dictionnaire etc., et (2) le niveau de la *macrostructure* (*Définition 15*) de la BDLex, qui regroupe tous les dictionnaires avec des commentaires etc.

(define-monolingual-dictionary english :language "English" :owner "CRL-NMSU")	**Niveau de métadonnées**
(define-lexical-database ULTRA :owner "CRL-NMSU" :comment "Une base lexicale fondée sur une approche interlingue" :dictionaries (english german spanish japanese chinese IR))	**Niveau de macrostructure**

Figure 6 : Exemples de métadonnées et de macrostructure décrites dans SUBLIM en LEXARD

Définition 15. La **macrostructure** d'une BDLex est la description de son architecture générale, c'est à dire des types de ses volumes et de leurs relations.

Le langage LINGARD permet de décrire la microstructure (*Définition 16*) des dictionnaires.

Définition 16. La **microstructure** d'un dictionnaire est la structure de ses articles, c'est à dire l'organisation de ses entrées.

En LINGARD, on peut décrire la structure informatique d'une unité lexicale en utilisant des constructeurs de base du langage, prédéfinis (ensemble, arbre, graphe, liste, automate, énumération, etc.). Voir la Figure 7.

[1] CLOS est l'ensemble des primitives ajoutées à Common Lisp pour le transformer en un langage à objets.

```
(def-linguistic-class french_entry
   (feature-structure
     (lexical_unit string)
     (Part-of-Speech (one-of "n.m" "n.f" "v.t" "v.i" "v.pr." "a" "adv"
"loc" "prep"))
     (example (set-of string))
     (indexer string)
     (quality (one-of "manual" "auto" "reviewed"))
     (properties (set-of property))
     (uws (set-of string))))
```

Figure 7 : Exemple de microstructure décrite dans SUBLIM en LINGARD

L'idée de définir la structure d'une BDLex multilingue a été adoptée dans le projet PAPILLON [Mangeot, M. & Sérasset, G., 2001 ; Boitet, C. et al., 2002 ; Mangeot, M., 2002] (voir I.1.3.2.3). Pour cela, on a utilisé XML au lieu de MCL. Mais, en pratique, seul le niveau des microstructures a été décrit de cette façon, au moyen de schémas XML avec variantes. Les variantes permettent de spécifier certains attributs et leurs valeurs, comme le genre, le cas, les dialectes, qui diffèrent pour chaque langue. Les métadonnées ont également été définies en XML, sous une forme proche du langage XML PLIST.

Par contre, la macrostructure de chaque base a été définie de façon seulement semi-formelle, en décrivant les relations entre les volumes. La macrostructure à "pivot par axies" de PARAX a été réutilisée telle quelle pour PAPILLON-NADIA [Sérasset, G., 1994a], LEXALP (voir II.1.3.2) [Sérasset, G., 2008], et (avec quelques extensions) PIVAX-1 (voir II.2).

Pour PAPILLON-CDM [Boitet, C. et al., 2002], la macrostructure se réduit à un ensemble de volumes, sans relations particulières entre eux. Le point intéressant est l'utilisation de *pointeurs CDM* (*Définition 17*) pour accéder aux dictionnaires électroniques intégrés dans la base comme s'ils étaient tous de même microstructure. On les présentera en détail au II.1.2.

Définition 17. **CDM** (Common Dictionary Markup) est une DTD (extensible) définissant un ensemble de balises XML correspondant aux types d'information des dictionnaires en source ouvert existants. Un **pointeur** CDM associé à une balise CDM `<bbb>` et à un dictionnaire `Dic` est un chemin XPATH permettant d'accéder à l'information correspondant à `<bbb>` dans `Dic`.

Cette technique permet d'importer très rapidement un dictionnaire dans la BDLex PAPILLON-CDM, de présenter l'ensemble des dictionnaires de façon unifiée, et de faire des recherches multicritère sur tout ou partie des dictionnaires de la base. Pour chaque dictionnaire, les pointeurs CDM sont stockés dans un fichier de métadonnées, de nouveau sous forme XML.

I.1.4.3 Ingénierie des BDLex contributives

JIBIKI. G. Sérasset et M. Mangeot ont adopté une architecture à 3 couches pour implémenter les BDLex contributives : une couche de présentation (responsable de l'interface avec les utilisateurs), une couche "métier" (qui fournit les services) et une couche "données" (responsable du stockage des données). La plate-forme JIBIKI a été implémentée avec ces trois couches, et développée principalement par M. Mangeot [Mangeot, M. & Chalvin, A., 2006] et G. Sérasset [Sérasset, G., 2004 ; Sérasset, G., 2008]. C'est une plate-forme générique en ligne de gestion, de création et de consultation de BDLex contributives. Elle a été principalement réalisée en JAVA sous l'environnement ENHYDRA[1] avec le SGBD POSTGRESQL, les connexions avec la BD étant réalisées par un pilote JDBC de type 4[2].

[1] http://enhydra.ow2.org/about.html

[2] http://www.commentcamarche.net/contents/596-les-types-de-pilotes-jdbc

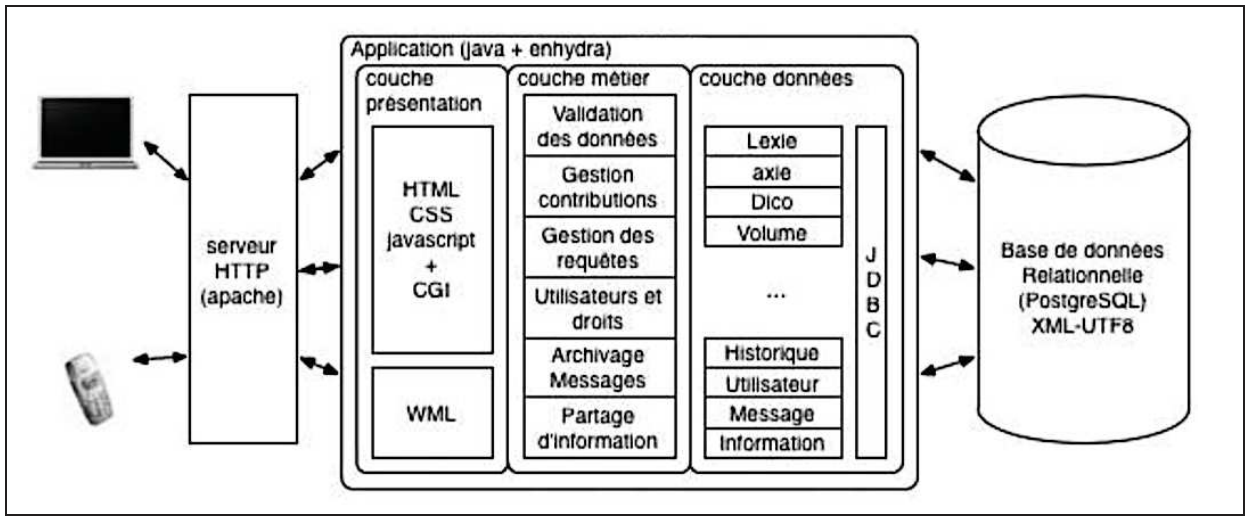


Figure 8 : Architecture à 3 couches de la plate-forme JIBIKI

Cette plate-forme a été conçue en 2003, après plusieurs évolutions du projet PAPILLON. Elle permet de construire et de mettre en œuvre des BDLex contributives en ligne de différentes architectures. C'est une sorte de "framework". JIBIKI a été utilisé dans plusieurs projets, comme PAPILLON (dont le logiciel a été réimplémenté en JIBIKI) [Mangeot, M. et al., 2003], LEXALP [Sérasset, G., 2008], MOTÀMOT [Mangeot, M. & Nguyen, H. T., 2009], GDEF [Mangeot, M. & Chalvin, A., 2006], PIVAX-1 [Nguyen, H.-T. et al., 2007] et DILAF.

Il y a ici deux niveaux de généricité, JIBIKI se plaçant au niveau d'un métamodèle : chaque type de BDLex réalisé en JIBIKI (comme LEXALP) peut être considéré comme un modèle de BDLex, et les BDLex concrètes en sont des instances. Ainsi, on pourrait déployer plusieurs bases terminologiques différentes de type LEXALP.

Après une dizaine d'années d'utilisation assez satisfaisante, des problèmes de rapidité et de passage à l'échelle sont apparus, principalement lors du projet OMNIA, quand on a voulu faire un grand nombre de recherches d'un seul coup dans une base PIVAX-1 (par exemple, 50 quand on cherchait tous les mots d'une légende de photo de BELGA-NEWS), ainsi qu'un traitement sur les résultats. Les causes de ces problèmes semblaient être la complexité de la macrostructure, et la grande taille des données. La taille elle-même ne pouvait pas être la seule cause, car elle était du même ordre de grandeur que celle de PAPILLON-CDM.

Au début de ma thèse en 2012, M. Mangeot avait prévu une grande mise à jour de JIBIKI, nommée JIBIKI-2, et j'y ai participé. Cette mise à jour a consisté à modifier la structure de la couche "données" et les algorithmes d'interaction avec le SGBD. Avec JIBIKI-2, le temps de réponse est beaucoup plus court (voir II.4.2.2).

WIKTIONARY/MEDIAWIKI. Aujourd'hui, WIKTIONARY est le dictionnaire contributif le plus connu au monde. Ce projet a commencé en décembre 2002. Il a été construit avec le logiciel MEDIAWIKI qui est le moteur de wiki de WIKIPEDIA. La première version (appelée USEMODWIKI) a été développée en PERL, puis améliorée et réécrite par Magnus Manske et Lee Daniel Crocker en PHP avec le SGBD MYSQL à partir de 2002. C'est en 2003 que le moteur a été appelé MEDIAWIKI.

La structure générale de MEDIAWIKI[1] a quatre couches : utilisateur, réseau, logique et données. La base de données contient le code wiki des pages et de nombreuses informations auxiliaires sur les pages, les utilisateurs, etc. Elle contient aussi les versions précédentes de toutes les

[1] https://www.mediawiki.org/wiki/MediaWiki

pages. Quand une page est consultée, le code wiki est converti en XHTML, ou bien ce code est pris du cache et envoyé à l'utilisateur, qui utilise son navigateur pour afficher le XHTML.

Couche utilisateur	Client Web		
Couche réseau	Serveur Web Apache		
Couche logique	Scripts PHP de MEDIAWIKI		
	PHP		
Couche données	Système de fichiers	Base de données MYSQL	Système de cache

Figure 9 : Architecture de MEDIAWIKI

Synthèse

Nous venons de voir l'évolution des idées en lexicographie computationnelle depuis 1980, date à laquelle on peut faire remonter la naissance de cette discipline.

Depuis 1980, on a cherché à faire le lien entre dictionnaires pour les systèmes de TA et pour les humains. Il faut reconnaître que les succès ont été très rares, que les difficultés sont réelles, et que les conséquences sont très dommageables. En effet, si les informations des dictionnaires de TA et des dictionnaires pour humains ne peuvent pas rester cohérentes, les unes ou les autres deviennent de moins en moins utiles dans les applications impliquant une synergie homme-machine, comme les outils d'aide à la traduction.

La seconde difficulté majeure, reconnue depuis 1985, est l'unification des dictionnaires généraux et des dictionnaires terminologiques, en les stockant (directement) dans une même base de données relationnelle, et en les transformant automatiquement vers les dictionnaires des systèmes de TA.

À partir de 1992, et de l'avènement d'Internet, la dictionnairique a évolué vers les réseaux et le contributif. La consultation en ligne est devenue très répandue. Par contre, même avec le Web 2.0 et les nouvelles possibilités de contribution, la construction contributive de dictionnaires s'est avérée très difficile à mettre en œuvre. Pour ce qui est des dictionnaires multilingues, on ne peut guère noter que YAKUSHITE.NET et WIKTIONARY.

Enfin, en parallèle, on s'est dirigé vers la conception de bases lexicales multilingues symétriques, pour surmonter les problèmes liés à l'asymétrie de dictionnaires de traduction. En passant par une base lexicale multilingue symétrique (de type PARAX ou NADIA), contenant des dictionnaires monolingues de lexies et des liens entre lexies synonymes (ce qu'on a appelé plus haut axies), on devrait pouvoir créer des dictionnaires inversés, ou composés, sans perte de qualité.

Nous allons maintenant décrire plus précisément la situation au départ de cette thèse, et dans son contexte particulier, académique et industriel.

I.2 Situation et problèmes en 2012

Au début de cette thèse, le contexte de la recherche sur les bases lexicales était complexe et multiple. D'autre part, comme il s'agissait d'une thèse CIFRE (université-industrie), les problèmes jugés les plus intéressants n'étaient pas les mêmes des deux côtés.

Nous présenterons d'abord la situation au laboratoire, puis les intérêts de L&M, et enfin les thèmes émergents au niveau de la communauté du TALN, comme l'extraction de termes bilingues et d'expressions polylexicales.

I.2.1 Au GETALP

I.2.1.1 Projets dérivés des thèses de G. Sérasset et de M. Mangeot

PIVAX. J'ai commencé à m'intéresser aux bases lexicales contributives lors d'un stage de M2P-GI, puis d'un stage "prédoc", les deux dans le cadre du projet ANR TRAOUÏERO (2011-2013), et plus précisément de la sous-tâche PIVAX++. Il s'agissait d'améliorer PIVAX et de le rendre vraiment opérationnel. PIVAX [Nguyen, H.-T. et al., 2007 ; Nguyen, H.-T., 2009 ; Nguyen, H. T. & Boitet, C., 2009] est une BDLex multilingue contributive orientée vers les systèmes de TA hétérogènes et diverses autres utilisations. Elle contenait non seulement d'assez gros volumes monolingues de langues naturelles (français, anglais, espagnol, russe…), mais aussi des dictionnaires d'UW (lexèmes interlingues) venant de plusieurs projets UNL, tout le WordNet anglais (v3), et l'ontologie du projet OMNIA. Le projet TRAOUÏERO visait en particulier à opérationnaliser l'outil PIVAX et à promouvoir la construction d'une base de données contenant au moins 4 langues et 1M UW.

JIBIKI. PAPILLON, LEXALP, et PIVAX ont été construits sur JIBIKI [Mangeot, M., 2006 ; Mangeot, M. & Chalvin, A., 2006 ; Sérasset, G. et al., 2006], une plate-forme générique de développement contributif de BDLex. Cette plate-forme avait été créée par G. Sérasset et M. Mangeot, et ce dernier commençait la création d'une nouvelle version, JIBIKI-2.

PAPILLON. Ce projet a été présenté plus haut. En 2012, l'objectif en ce qui le concernait était de reprogrammer son logiciel de base de JIBIKI-1 en JIBIKI-2, et de continuer à y ajouter des données venant de divers projets dictionnairiques comme MOTÀMOT et DILAF.

LEXALP. Il s'agit d'un projet européen INTERREG (janvier 2005 - février 2008), visant à harmoniser la terminologie en aménagement du territoire et développement durable au sein de la Convention Alpine afin de permettre aux six pays participant à cette convention (Autriche, France, Allemagne, Italie, Suisse et Slovénie) de communiquer et de coopérer efficacement dans quatre langues officielles (français, allemand, italien et slovène). À la fin du projet, la BDLex LEXALP contenait plus de 13K entrées [Sérasset, G. et al., 2006 ; Sérasset, G., 2008]. En 2012, ce projet n'était plus actif au laboratoire, mais la BDLex était toujours disponible.

DBNARY. G. Sérasset a commencé le projet DBNARY en 2012 (juste au début de cette thèse). Ce projet vise à extraire des données lexicales structurées en RDF (conforme au modèle LEMON), à partir des différentes éditions des wiktionnaires. En 2015, il y a des extracteurs pour 21 langues, qui produisent une "édition" par semaine[1].

GDEF. GDEF [Mangeot, M. & Chalvin, A., 2006] est un projet visant à créer un Grand Dictionnaire bilingue Estonien-Français, commencé en 2003. La BDLex sous-jacente est basée sur JIBIKI. Le GDEF contient actuellement plus de 16K entrées.

MOTÀMOT. C'est un autre projet (2009-2012) basé sur JIBIKI [Mangeot, M. & Nguyen, H. T., 2009 ; Mangeot, M. & Touch, S., 2010]. Il vise à élaborer un système lexical multilingue par le biais de la construction de dictionnaires bilingues ciblés sur des langues peu informatisées d'Asie du Sud-Est. Pour l'instant, il est limité au français et au khmer. Le volume français contient 13 249 entrées, le volume khmer 23 766, et le volume des axies 32 402.

UNL/U++C (UNL++). Ce projet [Boitet, C. et al., 2007 ; Nguyen, H.-T., 2009] est une variante du projet UNL. Il s'agit des travaux du consortium U++C, visant à construire des dictionnaires UW++-LN à partir des dictionnaires UW-LN existants (par unification des ensembles d'UW des différents groupes). Ces dictionnaires sont stockés dans PIVAX.

[1] Pour le nombre des entrées de chaque langue en détail, voir http://kaiko.getalp.org/about-dbnary/dataset/

I.2.1.2 La génération de mini-dictionnaires, une nouvelle application générique

Il s'agit de construire à la volée et très rapidement un "mini-dictionnaire" spécifique d'une sélection de texte. Cela serait très utile non seulement pour la "lecture active en langue étrangère", mais dans beaucoup d'autres contextes.

Durant le projet ANR OMNIA (2008-2011), Hong-Thai Nguyen a fabriqué des mini-dictionnaires associés aux légendes des photos dans des archives comme Belga-news, Flicker, etc. En parallèle, dans le cadre du projet SECTRA_W/IMAG, Cong-Phap Huynh et H.-T. Nguyen ont cherché à implémenter l'idée d'aide lexicale proactive associée à chaque segment, en faisant calculer à PIVAX-1 un mini-dictionnaire (bilingue ou multilingue) pour chaque segment de la mémoire de traductions associé à un document ou à un site Web. Ils ont alors rencontré quelques difficultés inattendues, puis n'ont pas eu le temps d'implémenter complètement leur solution.

Ces difficultés concernaient la définition exacte du format d'un mini-dictionnaire : il doit être très simple, et aussi permettre aux utilisateurs de corriger ou de compléter une entrée, avec certaines restrictions bien sûr, puis de le renvoyer à la base lexicale pour qu'elle mette à jour les informations. D'où un problème de synchronisation, puisque plusieurs intervenants peuvent en même temps modifier la base lexicale et les mini-dictionnaires. D'autre part, l'idée de calculer tous les mini-dictionnaires en arrière-plan s'est alors heurtée au manque d'un système de gestion de travaux (files d'attente, priorités, etc.).

I.2.1.3 Résultats de thèses antérieures non liées à JIBIKI

RÉCUPDIC/PRODUCDIC. Ces deux logiciels ont été réalisés par H. Doan-Nguyen durant sa thèse (soutenue en décembre 1998) [Doan-Nguyen, H., 1998a]. RÉCUPDIC est un système de récupération de ressources dictionnairiques à partir de leur format d'origine, par transformation en une structure plus profonde (LISPO) avec des informations explicites. PRODUCDIC est un outil destiné à la production de nouveaux ensembles lexicaux à partir de résultats de RÉCUPDIC. Il offre 7 fonctions (sélection, extraction, regroupement, inversion, enchaînement, combinaison parallèle et combinaison en étoile). Ces deux logiciels sont implémentés en LISP (CLOS). Ils ont été réutilisés dans la thèse de M. Mangeot [Mangeot, M., 2001] et dans la thèse de H.-T. Nguyen [Nguyen, H.-T., 2009].

JEMINIE. Il s'agit d'une plate-forme pour structurer les BDLex, produire et filtrer les acceptions bilingues ou multilingues, et évaluer la qualité de BDLex à acceptions interlingues. Elle a été créée par Aree Teeraparbseree dans le cadre de sa thèse [Teeraparbseree, A., 2005].

MULLING. MULLING (MultiLevel Linguistic Graphs) [Archer, V., 2009] est une bibliothèque C++ spécialisée pour la programmation linguistique d'algorithmes d'extraction de connaissances lexicales, et réalisée par Vincent Archer dans le cadre de sa thèse. Le modèle sous-jacent est un modèle de graphe linguistique multiniveau dans lequel les arcs sont internes à un niveau, ou vont d'un niveau à un niveau supérieur. Les opérateurs génériques permettent des applications multiples.

ROBODICO. ROBODICO [Nguyen, H.-T., 2009] est un programme associé à PIVAX, réalisé par H.-T. Nguyen. Il a pour but d'extraire des ressources lexicales à partir de sites de bases terminologiques ou de dictionnaires en ligne comme IATE[1]. Il simule les paramètres des requêtes HTTP, puis récupère les pages HTML et en extrait les ressources. J'ai récupéré cet outil et l'ai mis à jour pour accéder aux ressources d'IATE durant ma thèse.

[1] IATE (InterActive Terminology for Europe) a été lancé en 1999 à partir de l'intégration à Eurodicautom de plusieurs autres bases terminologies. Voir : http://iate.europa.eu.

SEPT et Préterminologie. Il s'agit d'un Système pour Éliciter une PréTerminologie (SEPT), développé par Mohammad Daoud dans le cadre de sa thèse [Daoud, M., 2010]. Ce système permet de construire et de maintenir des structures de graphe de préterminologie multilingue. Il a défini le terme "préterminologie" pour désigner une terminologie non validée, obtenue à partir de ressources non conventionnelles (comme des traces de consultation de sites Web).

I.2.2 Chez L&M

I.2.2.1 Une BDLex simple destinée aux glossaires multilingues

Lingua et Machina [Lingua&Machina, 2015] propose des outils linguistiques, SIMILIS [Planas, E., 2005] et LIBELLEX [Brown de Colstoun, F. et al., 2011]. C'est sur ce deuxième produit, LIBELLEX, que j'ai principalement travaillé durant cette thèse. LIBELLEX est une application Web de gestion des écrits multilingues en entreprise. Cette plate-forme intègre divers outils d'aide à la traduction (concordances bilingues, outils d'extraction et de gestion de terminologies, mémoires de traductions, systèmes de traduction automatique et outils de gestion de projets de traduction).

La fonction de gestion de terminologies de LIBELLEX (Libellex Termino) a été conçue en juillet 2010 par Estelle Delpech [Delpech, E., 2013]. Il y a une BDLex pour cette fonction liée à LIBELLEX, implémentée en Oracle. Les utilisateurs principaux de cette fonction sont des traducteurs. La conception de cette BDLex est bien adaptée aux glossaires, mais pas aux termes généraux.

Le projet ANR METRICC[1] a duré entre décembre 2008 et décembre 2011, et s'est terminé juste avant le début de ma thèse. Ce projet avait pour but l'extraction de ressources terminologiques à partir de corpus comparables. Il y a une BDLex liée à ce projet pour stocker les termes extraits. Ces analyses et extractions sont réalisées vers des formats UIMA[2].

E. Delpech a proposé de réutiliser la structure de BDLex du projet METRICC dans LIBELLEX. LIBELLEX a été étendu pour pouvoir travailler avec les lexiques extraits de corpus comparables au format d'échange de METRICC.

Les formats TBX[3] (TermBase eXchange) et TEI[4] (Text Encoding Initiative) sont des formats adaptables. Le format TEI_METRICC et le format TBX_METRICC sont deux adaptations aux besoins de METRICC et sont utilisés comme formats d'échange dans METRICC. La BDLex de METRICC a été conçue à partir de ces deux formats.

On trouvera la définition des structures de TBX et de TBX-METRICC et la structure de BDLex de METRICC au III.1.2.3.

---

[1] METRICC : http://www.metricc.com. Les participants étaient le laboratoire LINA (Nantes), L&M, Syllabs, LiCoRN, MRIM (LIG, Grenoble), SINEQUA et VALORIA.

[2] UIMA est un standard du consortium OASIS-open (https://www.oasis-open.org/) conçu par CMU (Carnegie Mellon University) et IBM pour la gestion de flux de ressources et de flots de données linguistiques, éventuellement liés à des bases de connaissances ou à des ontologies.

[3] Le format TBX (TermBase eXchange) est un standard XML, publié par l'ISO en 2008 (ISO 30042:2008), pour permettre l'échange de données terminologiques structurées.

[4] Le format TEI est un standard XML mis au point par la Text Encoding Initiative (TEI) pour permettre la représentation de textes sous une forme digitale.

I.2.2.2 **Problèmes perçus**

I.2.2.2.1 Limitation du schéma conceptuel

Comme LIBELLEX a réutilisé la même structure de BDLex que METRICC, sa BDLex a été également conçue à partir des formats d'échange TEI_METRICC et TBX_METRICC. Dans cette BDLex, il n'y a donc pas de niveau pour représenter les sens, ce qui est la source de nombreux problèmes.

En effet, le format TBX standard, tel que décrit dans l'ISO 30042, a été spécifiquement conçu pour encoder des données terminologiques créées par des humains et sans ambiguïtés. Les données terminologiques sont organisées en "concepts", les termes étant de simples désignations de ces concepts dans différentes langues. Les relations sémantiques (synonymie, antonymie, etc.) relient des concepts, et non des termes. Elles sont donc indépendantes des langues.

Mais, dans TBX-METRICC, le schéma conceptuel est limité à l'équivalence de termes. E. Delpech a expliqué cela de la façon suivante dans la spécification du format TBX-METRICC sur l'intranet de Lingua et Machina.

"Les données terminologiques générées automatiquement sont ambiguës, typiquement, un terme source est associé à plusieurs termes cible (traductions candidates). Il n'existe pas de notion de concept : les outils d'extraction terminologique ne reconnaissent que des termes, charge au terminologue de les associer ensuite à des concepts. En plus, dans un contexte multilingue, avec une perspective d'application à la traduction, l'organisation en "concepts" présente peu d'avantages. Cette organisation suppose l'existence de concepts universaux, indépendants des langues et une structuration des concepts identique de langue à langue. Cela ne correspond pas à la réalité que connaissent les traducteurs : chaque langue impose un découpage différent de la réalité et il est très difficile de maintenir un parallélisme entre les langues."

I.2.2.2.2 Une conséquence: le problème des abréviations

Il y a des besoins réels dans l'industrie. Par exemple, EDF, client de L&M, a demandé un traitement complet des acronymes, et un autre client, Wesco, voulait nourrir un dictionnaire d'abréviations. En 2012, LIBELLEX-TERMINO ne pouvait pas traiter ce type de problème. Comme il n'y a pas de niveau pour traiter les sens, c'était très difficile à faire. Il s'agit d'import, d'export, de manipulation, d'édition des acronymes et d'autres abréviations. La seule chose qu'on pouvait faire en 2012, c'était créer une relation entre deux termes de la même langue, et cette relation ne pouvait être que "variante de terme"[1].

En 2013, un autre client, ExaleadSuggest, nous a demandé de traiter leur terminologie avec les relations "vedette, synonyme, variante non typée, variante de type abréviation, et variante de type acronyme" entre termes monolingues. En même temps, Louis Vuitton a demandé à L&M un traitement de la relation hyperonymie/hyponymie des termes monolingues en utilisant le mot-clé "parent-enfant", par exemple, "couleur" est *parent* de "rouge" et "rouge" est *enfant* de "couleur".

I.2.2.2.3 Dépendance d'un logiciel propriétaire

La BDLex de LIBELLEX a été développée sous ORACLE. Pendant ma thèse, vers 2014-15, L&M a fait une migration d'ORACLE vers POSTGRESQL. À partir de ce moment, je ne pouvais plus

[1] La contrainte est spécifiée dans la base de donnée et le programme, LIBELLEX n'accepte ni ne traite aucun autre type.

modifier la structure de BDLex, mais l'API (en SQL) est restée la même, au moins en ce qui concernait ma partie.

I.2.3 Dans la "communauté scientifique" du TAL

I.2.3.1 Extraction de termes techniques

Au début de ma thèse, il y avait beaucoup d'activités sur l'extraction de termes en contexte monolingue et multilingue, et sur l'extraction d'expressions polylexicales (termes et prédicats composés).

La thèse de Violeta Seretan [Seretan, V., 2008] propose (et évalue sur des données en 4 langues, anglais, français, espagnol et italien) une procédure principale d'extraction de collocations binaires qui se base sur l'application de la contrainte de proximité syntaxique aux éléments d'une collocation candidate, à la place de la contrainte de proximité linéaire qui est la plus répandue dans les travaux existants. Ses résultats sont bien meilleurs que ceux des approches linéaires, mais sa méthode suppose la disponibilité, pour chaque langue considérée, d'un analyseur du type et de la qualité de FIPS [Wehrli, E., 2007], qui produit des arbres syntaxiques. Malgré tout, c'est une voie d'avenir.

La thèse d'Estelle Delpech [Delpech, E., 2013] a reçu le prix de thèse de l'ATALA en 2014. C'était aussi une thèse CIFRE avec L&M. Elle concernait l'extraction de lexiques bilingues à partir de corpus comparables, appliquée à la traduction spécialisée. E. Delpech a travaillé d'une part pour le projet ANR METRICC, et d'autre part pour la fonction de terminologie de LIBELLEX chez L&M. Sa conclusion est que ces techniques ne produisent rien de réellement intéressant et utilisable pour la traduction professionnelle, et qu'il faut d'abord faire un travail terminologique, qui produit à la fois les termes et leurs traductions (c'est d'ailleurs ce que font les gros donneurs d'ordres comme IBM). Il y a cependant toujours des recherches de ce type en cours.

CAMELEON[1] (Collaborative and Automatic Methods for the Multilingualisation of Lexica and Ontologies). Ce projet collaboratif réunissait des équipes françaises (GETALP du LIG et MELODI de l'IRIT) et des équipes brésiliennes (UFRGS, PUCRS et UFSCAR). Ce projet a commencé en 2010 et s'est terminé en 2014.

Le projet ANR **METRICC** (MEmoire de Traductions, Recherche d'Information et Corpus Comparables) venait juste de se terminer début 2012. Nous l'avons déjà mentionné au I.2.2.1. Comme la thèse d'Estelle Delpech [Delpech, E., 2013], il avait pour but l'extraction de ressources terminologiques à partir de corpus comparables. Malgré les résultats négatifs (et de principe) de cette thèse, ce thème continue d'être actif dans la communauté du TALN.

Presque en parallèle, le projet européen **TTC**[2] (Terminology Extraction, Translation Tools and Comparable corpora) s'est déroulé du 1er janvier 2010 au 31 décembre 2012. Les participants étaient le LINA (Nantes), l'INLP (Institute for Natural Language Processing) de Stuttgart, UL (University of Leeds), Sogitec, Syllabs, TILDE et Eurinnov. Il visait à exploiter les possibilités offertes par les corpus comparables pour améliorer les performances des outils informatiques de traduction. Il s'agissait de traiter des domaines techniques dans un contexte massivement multilingue où il est nécessaire de traduire un même document dans beaucoup de langues.

[1] Voir http://cameleon.imag.fr

[2] Voir http://www.ttc-project.eu

Le projet chinois (No. 07JC870006) "Research on text mining applied to plagiarism detection" (文本挖掘技术在论文抄袭判定中的研究) et son sous-projet "Chinese Text Keywords Extraction Based on Fuzzy Processing" (基于模糊处理的中文文本关键词提取方法) a été lancé par Anhui Finance and Economics University (Anhui, Chine) entre 2007 et 2012. Ce projet a proposé des algorithmes sur les extractions de mots-clés pour le chinois.

Les **workshops CHAT** (Creation, Harmonization and Application of Terminology Resources)[1] ont eu lieu en 2011 et en 2012.

I.2.3.2 Extraction d'expressions polylexicales (EPL)

En ce qui concerne le traitement des EPL (expressions polylexicales[2]), il y a une activité plus récente, mais assez forte.

Workshops MWE (MultiWord Expressions) : cette série de workshops[3] a commencé en 2003, sur un rythme annuel. On parle de plus en plus de "**PLE**" (PolyLexical Expressions).

La thèse de Carlos Ramisch [Ramisch, C., 2012] a abordé le problème du traitement des EPL dans les applications de TAL. La plate-forme MWETOOLKIT [Ramisch, C., 2012] a été développée dans le cadre de sa thèse et a été utilisée dans le projet CAMELEON. C'est une plate-forme générique en source ouvert permettant de faire plusieurs traitements sur les EPL, y compris les extractions et les évaluations. Toute une communauté l'utilise maintenant.

Le thème de l'extraction d'EPL à partir de corpus parallèles a été envisagé depuis longtemps, mais ne semble pas encore avoir été réellement abordé. Pour l'étudier avec profit, il faudrait sans doute inclure l'allemand dans les langues considérées, mais cela suppose une certaine connaissance de cette langue.

I.2.3.3 Projets de BDLex "sémantiques"

WORDNET [Miller, G. A. et al., 1990] a pour but de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Il y a de nombreux projets dérivés, comme **EUROWORDNET** (néerlandais, italien, espagnol, allemand, français, tchèque et estonien) [Vossen, P., 1998] et **INDOWORDNET** (18 langues utilisées en Inde) [Chakrabarti, D. et al., 2002].

FRAMENET [Fillmore, C. J. & Atkins, B. T. S., 1998] a été lancé depuis 1997. L'objectif est d'élaborer une ressource linguistique accessible par les humains et les machines. Cette ressource décrit initialement le lexique anglais selon les principes des cadres conceptuels, chaque unité lexicale étant illustrée par des exemples annotés manuellement. À ce jour, il y a plusieurs projets FRAMENET pour diverses langues (coréen, chinois, brésilien, allemand, japonais, suédois, etc.).

---

[1] http://www.tilde.com/tilde-research/workshop-creation-harmonization-and-application-terminology-resources

[2] Ce terme (EPL/PLE) est préférable à "multimot" (MWE = MultiWord Expression), car, dans beaucoup de langues, l'allemand en particulier, "multimot" n'est pas approprié. "Hauptbahnhofgepäckaufbewahrung" (consigne à bagages de la gare principale) est un seul mot, contenant 4 lexèmes.

[3] MWE 2003 (premier workshop d'ACL à Sapporo), MWE 2004 (workshop d'ACL à Barcelona), MWE 2006 (workshop d'ACL à Sydney, workshop d'EACL à Trento, workshop de Colloc à Berlin), MWE 2007 (workshop d'ACL à Prague), MWE 2008 (workshop de LREC à Marrakech), MWE 2009 (workshop d'ACL à Singapore), MWE 2010 (workshop de COLING à Beijing), MWE 2011 (workshop d'ACL à Portland), MWE 2012 (workshop de *SEM à Montréal), MWE 2013 (workshop de NAACL à Atlanta), MWE 2014 (workshop d'EACL à Gothenburg), MWE 2015 (workshop de NAACL à Denver). Voir http://multiword.sourceforge.net.

PROLEX[1]**.** Ce projet de lexique syntaxique et sémantique de noms propres, basé sur le concept de *prolexème*, dû au célèbre linguiste et sémiologue Eugène Coseriu, a été lancé par le laboratoire d'informatique de l'université François-Rabelais de Tours. Dans sa thèse, Mickaël Tran [Tran, M., 2006] a développé **PROLEXBASE**, un outil supportant un lexique multilingue de type PROLEX, et proposé la stratégie "prolexème" [Grass, T. et al., 2004 ; Tran, M. & Maurel, D., 2006]. Voici la définition de prolexème dans la thèse de M. Tran :

Définition 18. Dans notre modèle (PROLEXBASE), le **prolexème** correspond à une projection du nom propre conceptuel dans une langue donnée.

Chaque prolexème d'une langue donnée sera donc relié à un seul et unique nom propre conceptuel. C'est en se basant sur cette relation que l'on va pouvoir traduire les prolexèmes d'une langue vers une autre. Le concept de prolexème peut aussi se définir comme une classe d'équivalence de synonymes. Pour simplifier, nous considérons aussi le prolexème comme le lemme associé aux différentes formes d'un nom propre qui apparaissent dans les différents textes d'une langue donnée. Il peut ainsi être considéré comme la forme vedette d'un ensemble de dérivés et d'alias [Tran, M., 2006].

BABELNET [Navigli, R. & Ponzetto, S. P., 2010] est un grand réseau lexical multilingue qui fournit des entrées lexicales (plus de 9 millions) dans plus de 50 langues, reliées entre elles par une grande quantité de relations sémantiques. Il a été créé automatiquement en utilisant les ressources de WIKIPEDIA et de WORDNET, et la traduction automatique.

OMNIA [Rouquet, D. et al., 2013] (2008-2011) fut un projet financé par l'ANR, avec comme participants XRCE (Xerox), LIG (GETALP), et ECL (LIRIS). Ce projet visait à filtrer des documents contenant du texte et des images, dans un contexte de grandes masses de données.

CJK.ORG [Halpern, J., 2002 ; Halpern, J., 2006 ; CJK, 2015], construit une très grosse base qui contient environ 24 millions d'entrées en japonais, chinois (simplifié et traditionnel), coréen, anglais et arabe, et comprenant un riche ensemble d'attributs grammaticaux et sémantiques, surtout pour des noms propres (donc des entités nommées).

HOWNET [Dong, Z. et al., 2010] a été construit à partir de 1988. C'est une base de connaissances du sens commun (online common-sense knowledge base) pour l'anglais et le chinois qui repose sur des annotations exprimant des relations entre concepts et des relations entre attributs de concepts.

Il y a 5 points principaux pour les différences entre WORDNET (WN) et HOWNET (HN) : (1) WN est orienté vers les humains et HN vers les machines ; (2) les annotations de WN sont faites par des humains et celles de HN par des programmes ; (3) WN est basé sur les mots et HN sur les concepts ; (4) WN a des "synsets" qui sont des ensembles de quasi-synonymes, et HN a des sémèmes qui sont des composants de sens ; (5) WN a des définitions en langue naturelle alors que HN utilise des définitions exprimées dans un langage formel, avec des balises.

L'Annexe 4 contient des réponses détaillées du Prof. Dong Zhen Dong à un certain nombre de nos questions.

Ressources ontologiques. Il y a beaucoup d'ontologies disponibles, par exemple, UMLS[2] (Unified Medical Language System) pour les concepts biomédicaux, ou les ressources de la

[1] http://www.cnrtl.fr/lexiques/prolex/

[2] https://www.nlm.nih.gov/research/umls/

FAO (AGROVOC[1] et FAOTERM[2]) pour l'ontologie de l'agriculture et pour l'ontologie de la géopolitique.

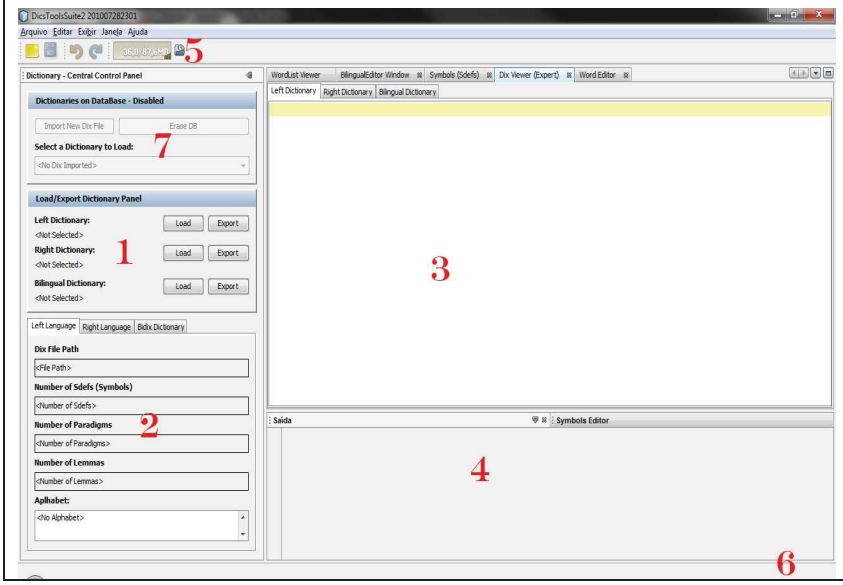
TOTh. La conférence Terminologie et Ontologies : Théories et applications (TOTh) a lieu chaque année en juin à Annecy depuis 2007 ; elle est organisée par l'institut Porphyre, la société française de terminologie, et l'Université de Savoie.

TIA. Le groupe TIA (Terminologie et Intelligence Artificielle) organise des conférences depuis 1995 : TIA'95 (Villetaneuse), TIA'97 (Toulouse), TIA'99 (Nantes), TIA'01 (Nancy), TIA'03 (Strasbourg), TIA'05 (Rouen), TIA'07 (Nice), TIA'09 (Toulouse), TIA 2011 (Paris), TIA 2013 (Villetaneuse), TIA 2015 (Grenade).

I.2.3.4 Création contributive de dictionnaires pour la TA

YAKUSHITE.NET (voir I.1.3.2.1) est une plate-forme Web grâce à laquelle des traducteurs peuvent consulter et augmenter (aspect contributif) des dictionnaires en ligne, organisés en un hiérarchie. Ces dictionnaires sont automatiquement intégrés dans le système TA PENSEE de OKI Electric (Japon).

APERTIUM [Forcada, M. L. et al., 2011 ; Apertium, 2015]. Dans le cadre du projet APERTIUM, il y a 218 dictionnaires monolingues en 93 langues différentes, avec plus de 3,8M entrées, et 236 dictionnaires bilingues en 235 couples de langues différents, avec plus de 2,5M entrées[3]. L'outil associé DICSTOOLSSUITE aide les utilisateurs (non-programmeurs) à faire de la création contributive de dictionnaires XML grâce à des interfaces conviviales.



1. Zone pour dictionnaires sélectionnés (on peut importer ou exporter un dictionnaire)
2. Statut de dictionnaires (reperatoire de fichier, nombre de lemmes, nombre de symboles…)
3. Zone principale pour manipulation
4. Composants pour la communication (pour affichage des erreurs, des informations d'utilisateur etc.)
5. Barre d'outils et status de mémoire (affichage de la mémoire utilisée par l'application)
6. Zone pour afficher les messages de traitement
7. Zone pour la base de données (désactivée temporairement)

Figure 10 : Interface de DICSTOOLSSUITE

UNL. On a déjà mentionné le projet UNL au I.1.3.2.2. En France, les dictionnaires initiaux ont été contribués par très peu de personnes, contribuant chacune des volumes assez grands. Le dernier projet, appelé GBDLEX-UW++, a été supporté par PIVAX-1 à Grenoble. C'est H.-T. Nguyen qui a préparé l'environnement pour qu'on puisse faire de la création contributive, et qui a mis les dictionnaires initiaux en ligne. Malheureusement, le projet n'a plus été "animé" depuis son départ. Il faut cependant noter les très importantes contributions de V. Dikonov de l'IPPI (plus de 800000 UW avec leurs traductions dans 4 à 5 langues).

[1] http://aims.fao.org/fr/agrovoc

[2] http://www.fao.org/faoterm/fr/

[3] La liste complète de dictionnaire : http://wiki.apertium.org/wiki/List_of_dictionaries

I.2.3.5 **Désambiguïsation lexicale**

VIDEOSENSE[1]. Dans le cadre de ce projet (2009-2013), on a proposé d'utiliser un répertoire de sens "pivot" artificiels pour la désambiguïsation de données textuelles multilingues accompagnées de vidéo. Comme candidats à être ce répertoire pivot, il y avait UNL[2] et WORDNET[3], mais leurs représentations des sens ont des problèmes comme la polysémie et l'incomplétude. Finalement, on a utilisé les vecteurs conceptuels (avec "émergence") pour représenter les sens. Mais… ces vecteurs ne donnent qu'une représentation "thématique" des sens, et de plus les résultats dans les campagnes d'évaluation ont été très décevants.

SEMEVAL (Semantic Evaluation). Depuis 1998, il y a eu plusieurs campagnes[4] (semeval-senseval) destinées à évaluer la désambiguïsation lexicale. Les évaluations ont concerné 18 langues : arabe, basque, tchèque, chinois, catalan, hollandais, danois, anglais, estonien, français, japonais, italien, coréen, allemand, espagnol, suédois, roumain, turc.

Thèses et mémoires. Il y a eu plusieurs thèses et mémoires autour de ce thème, avant 2012, comme la thèse de D. Schwab [Schwab, D., 2005], celle de L. Audibert [Audibert, L., 2003], ou le mémoire de master de B. Brosseau-Villeneuve [Brosseau-Villeneuve, B., 2010].

Il y a aussi des recherches sur l'estimation des coefficients de confiance dans les systèmes de TA, par exemple la thèse de N.Q. Luong [Luong, N. Q., 2014].

I.3 Thèmes de recherche abordés[5]

I.3.1 Thèmes encore ouverts

Au début de cette thèse, 8 thèmes étaient encore ouverts dans notre contexte : trois thèmes "lexicographiques", trois thèmes concernant la conception de services, et deux thèmes plus liés au génie logiciel (GL).

I.3.1.1 **Trois thèmes "lexicographiques"**

I.3.1.1.1 Concevoir des BDLex "métier", contenant des éléments plus ou moins "situés"

La difficulté de l'unification dans une même base lexicale de lexèmes (simples ou composés) généraux, de termes techniques, d'abréviations, de noms propres et d'entités nommées semble provenir du fait qu'ils sont plus ou moins "situés".

Un lexème général comme *chemin*, ou *boire*, ou *vite* a une acception (sens en usage) que ne dépend pas d'un contexte d'interprétation particulier. Nous dirons qu'il est *général,* ou *non situé.*

Un terme technique comme *ampoule* a des sens dépendant de domaines particuliers (électricité, médecine, religion), mais pas du lieu ni du temps. Nous dirons qu'il est *terminologique*.

---

[1] http://www.videosense.org/

[2] Un UW décrit un sens par une liste de restrictions appliquée aux sens possibles d'un lemme en général anglais, le *headword* de l'UW.

Ex: `book(icl>do, agt>human, obj> thing); book(icl>document); tatami(icl>furniture)`

[3] Un sens correspond à un couple `<mot, synset>`.

[4] Senseval-1 en 1998 à Sussex, Senseval-2 en 2001 à Toulouse, Senseval-3 en 2004 à Barcelone, SemEval-2007 à Prague, SemEval-2010 à Uppsala, SemEval-2012 à Montréal, SemEval-2013 à Atlanta, SemEval-2014 à Dublin, SemEval-2015 à Denver, SemEval-2016 à San Diego.

[5] La rédaction de cette section a été faite en collaboration avec Ch. Boitet.

Une abréviation comme *CNAM* peut signifier *Caisse Nationale d'Assurance Maladie* ou *Conservatoire National des Arts et Métiers*. Un toponyme comme *Rome* renvoie à une ville, certes, mais s'agit-il de la capitale de l'Italie, de Rome (New-York), ou d'autres villes de ce nom aux USA ? Pour l'interpréter correctement, il faut disposer de la situation, ou au moins de paramètres pertinents de la situation. Nous dirons que *CNAM* et *Rome* sont *situés*.

Enfin, le sens des noms de sociétés, ou des noms de personnes, comme *George W. Bush*, dépend souvent d'une description encore plus précise du contexte. Même si on ajoute *président des USA*, il y en a eu deux de ce nom… et cette qualité leur reste attachée. En fait, le sens de beaucoup d'*entités nommées* dépend non seulement du domaine et du lieu, mais aussi du temps. On peut dire que les noms de personnes et ce qu'on appelle depuis quelques années les entités nommées sont *très situés*. Par exemple, *la capitale de la RFA* a été Bonn, puis Berlin à partir de 1991.

Un *prolexème* est par définition une classe d'équivalence d'expressions synonymes par rapport à une certaine situation. Ainsi, *P1 = {président des USA, Georges W. Bush#2}* est valide par rapport à 2005, et *P2 = {président des USA, Barack Obama#1}* est valide par rapport à 2015.

Définition 19. Nous appellerons "**situement**" la qualité d'être situé, et parlerons du "degré de situement" d'un lexème.

Ce terme "situement" semble naturel, car il est solution de plusieurs équations analogiques [Lepage, Y., 2000] comme : *situer:situement::figer:figement::dénué:dénuement*. On parle de même du "degré de figement" d'une collocation.

Définition 20. Chaque **type de vocable** correspond à un degré de situement : général, terminologique, situé, ou très situé.

Pour bien expliquer les degrés de situement, nous donnons ici quelques exemples.

Table 1 : Exemples pour les différents degrés de situement

Mot-vedette	**Degré de situement**	**Descripteur de situement**	**Signification**
sémaphore	général	sémaphore#général	signalisation maritime
	situé	sémaphore#situé?lieu=Béar	sémaphore du cap Béar (Chemin du Cap Béar, 66660 Port-Vendres)
	termino-logique	sémaphore#terminologique?domaine=informatique	en informatique, dispositif de verrouillage de ressources
diabolo	général	diabolo#général	instrument de jonglage
	termino-logique	diabolo#terminologique?domaine=médecine	aérateur en ORL
président	général	président#général	chef d'État, président d'une compagnie, chef d'un tribunal, chef d'un acte etc.
	très situé	président#situé?lieu=Etats-Unis&année=2015	Barack Obama
	très situé	président#situé?lieu=Etats-Unis&année=2005	George W. Bush

Définition 21. Une "**BDLex métier**" est une BDLex qui couvre tous les types de vocable dans une même base pour un certain domaine.

Une BDLex métier unifie donc les différents types d'unités lexicales, y compris les plus récemment traités comme les prolexèmes (on peut dire que "prolexème" en lexicographie

correspond à "entité nommée" en sémantique). On présentera les prolexèmes en détail dans la section III.2.2.

I.3.1.1.2 Faire évoluer les BDLex vers l'approche "ontoterminologique"

Pour l'instant, on peut intégrer une ontologie ou une classification sémantique en tant que "volume" dans une base PIVAX. On préférerait introduire des liens étiquetés comme tels allant des axies vers des ontologies ou des classifications sémantiques (comme le Goi Taikei[1]). Le but de ce thème est de réaliser les identifications/raffinements des synonymes, des parasynonymes, et leurs dénotations et connotations terminologiques.

Dans les exemples de la Table 1, ces liens étiquetés permettent d'une part de préciser le degré de situement et d'identifier les paramètres pertinents du contexte (temps, lieu, domaine, etc.), et d'autre part de relier entre eux les synonymes et les parasynonymes, par exemple, "*président [des Etats-Unis en 2015]*" et "*Barack Obama*".

I.3.1.1.3 Établir un lien direct entre les BDLex "métier" et les réseaux lexicaux exprimables en RDF

Les données lexicales sont stockées en XML et traitées avec l'API DOM dans la BDLex de la plate-forme JIBIKI. Est-il possible d'implémenter le niveau bas d'une plate-forme comme JIBIKI en RDF, et par là d'obtenir un lien direct avec DBNARY [Sérasset, G., 2012 ; Sérasset, G. & Tchechmedjiev, A., 2014] ? Cela permettrait aussi sans doute de pouvoir considérer une BDLex quelconque comme un "réseau de neurones de Hopfield", et de l'utiliser directement dans des services de désambiguïsation lexicale (voir I.3.1.2.3 ci-dessous).

I.3.1.2 **Trois thèmes concernant la conception de services lexicaux**

I.3.1.2.1 Offrir des services dictionnairiques en ligne non seulement pour les humains, mais pour des logiciels (aspect SaaS : Software as a Service)

Les services dictionnairiques en ligne pourraient être utilisés par des applications dans plusieurs domaines. Par exemple, une BDLex en PIVAX pourrait être utilisée comme *serveur lexical* pour des systèmes de TA, pour des systèmes de recherche translingue d'informations (CLIR) comme celui du projet OMNIA, et pour des système de PE (post-édition) contributive en ligne (ex. SECTRA/IMAG) et de THAM en général, par production proactive de "mini-dictionnaires" [Huynh, C.-P., 2010].

I.3.1.2.2 Accéder à une ou plusieurs BDLex en utilisant un ou plusieurs services de lemmatisation

Il n'y a pas de service de lemmatisation couvrant tous les mots et termes d'une langue, et d'autre part il y a des lemmatiseurs particulièrement adaptés à certains types de termes (par exemple, les termes techniques en aviation, en agronomie, ou en médecine). On aimerait les mettre en synergie. Comme ils ne fournissent pas leurs résultats dans les mêmes formats, ni même ne sont fondés sur des notions identiques, il faudrait trouver comment "normaliser", puis fusionner ces résultats.

I.3.1.2.3 Offrir un service de support à la désambiguïsation lexicale (WSD)

On a déjà mentionné les travaux de MSR au I.1.3.1.4. MSR a montré comment faire cela vers 1993-96 en implémentant un dictionnaire (en fait, la fusion du Longman Dictionary of Contemporary English et de l'American Heritage Dictionary) comme un grand réseau "neuronal" à la Hopfield. Lors de MIDDIM-96, Bill Dolan a présenté une application de

[1] http://www.kecl.ntt.co.jp/icl/lirg/resources/GoiTaikei/index-en.html

désambiguïsation lexicale basée sur cette ressource, avec la méthode dite de l'amorçage lexical (lexical priming) [Dolan, W. B. & Richardson, S. D., 1996].

Le système HOWNET traite la WSD pour le chinois en utilisant un grand réseau de connaissances générales (plus de 50K entrées). La méthode est appelée "élagage de sens basée sur la connaissance". Cette méthode a été présentée par Chi Yung Wang en 2002 [Gan, K.-W. et al., 2002].

I.3.1.3 Deux thèmes plus liés au GL

Si une BDLex fonctionne en serveur lexical pour des programmes, un grand nombre de requêtes pourrait lui être soumis en même temps. Pour mettre en place l'aspect serveur générique, deux problèmes étaient encore ouverts : le passage à l'échelle sans perte d'efficacité, et l'organisation de la gestion des travaux (requêtes et tâches associées).

I.3.1.3.1 Efficacité et passage à l'échelle (les implémentations étaient bien trop lentes)

Concrètement, le problème était d'améliorer les performances de PIVAX-1. Dans le cadre du projet OMNIA, H.-T. Nguyen avait importé plus de 848K d'entrées d'UNL dans sa base, et avait essayé de l'utiliser comme serveur lexical, pour produire à partir de la liste des lemmes d'un paragraphe (une légende de 50 mots en moyenne) une petite grammaire en Systèmes-Q réalisant l'ajout des UW à la sortie de NOOJ sur ce paragraphe[1]. Mais les temps de réponse étaient beaucoup trop longs, et on avait dû remplacer PIVAX par une base de données ad hoc pour réaliser un démonstrateur.

I.3.1.3.2 Gestion des travaux (files d'attente)

Le thème de la gestion des flux numériques en TALN est devenu un domaine reconnu[2]. On a ce besoin pour plusieurs services. Par exemple, SECTRA_W [Huynh, C.-P., 2010] appelle TRADOH [Vo-Trung, H., 2004] pour appeler le ou les systèmes de TA fournissant les prétraductions. Si on fait un grand nombre de requêtes en même temps, il y a souvent un problème de décalage, et les prétraductions ne sont pas bien alignées avec les segments source.

On a aussi besoin d'une gestion des travaux en arrière-plan, par exemple, pour créer des mini-dictionnaires. Cong-Phap Huynh a proposé dans sa thèse de construire les mini-dictionnaires (un pour chaque segment contenu dans SECTRA) par appel de PIVAX, mais il n'a finalement pas pu le réaliser, à cause de plusieurs problèmes techniques, dont la gestion de travaux. Enfin, l'outil SEGDOC [Kalitvianski, R., 2013] a été créé principalement par Ruslan Kalitvianski pour segmenter de grands documents, et lui aussi a besoin d'une fonction de gestion de files d'attente.

I.3.2 Thèmes retenus

Il n'était pas possible de traiter tous ces thèmes dans le cadre d'une thèse. D'autre part, il fallait considérer les besoins de L&M, les intérêts du laboratoire, ainsi que mes intérêts et compétences propres, sachant que je venais d'un master en génie informatique et pas en linguistique ou en sciences cognitives.

Finalement, 4 thèmes ont été retenus : (1) la conception de BDLex unifiant tous les types d'unités lexicales, (2) la réalisation d'un service générique de création de mini-dictionnaires à

---

[1] NOOJ produit un graphe de chaînes donnant les différentes solutions morphologiques, avec pour chacune le lemme et les autres propriétés comme le genre, le nombre, le mode, le temps, la personne…

[2] Nous avons déjà mentionné UIMA. Adobe est aussi très actif dans ce domaine, voir https://www.adobe.com/content/dam/Adobe/en/devnet/linguisticlibrary/lilo-programming-guide-cs6.pdf

partir de BDLex, (3) la conception et l'implémentation d'un intergiciel de lemmatisation, et (4) la recherche et l'intégration d'un système de gestion de tâches "linguicielles" entre serveurs ou agents à gros grain.

I.3.2.1 Conception d'une BDLex unifiant tous les types d'unités lexicales (générales, terminologiques, situées et très situées)

Au début de ma thèse, L&M m'a demandé d'étudier l'intégration de la gestion des acronymes dans la base terminologique de LIBELLEX. Donc j'ai travaillé sur ce thème à cause de ce besoin industriel. Avec la conception d'une BDLex "métier", on peut organiser et gérer les différents types d'unités lexicales, y compris les plus récemment traités comme les prolexèmes dans une même BDLex de PIVAX.

La première version de PIVAX ne pouvait pas implémenter ce type de BDLex, ce qui m'a amenée à produire avec M. Mangeot une nouvelle version de PIVAX, PIVAX-2, avec l'idée de représenter des annotations sémantiques ou d'autres annotations par des liens étiquetés libres. Le problème a été complètement résolu au niveau du laboratoire, et seulement incomplètement pour L&M, car il était impossible de modifier suffisamment le module LIBELLEX-TERMINO.

I.3.2.2 Réalisation d'un service générique de création de mini-dictionnaires pour SECTRA à partir d'une BDLex en JIBIKI

Nous ne nous sommes pas limitée à achever le travail commencé par C. P. Huynh et H. T. Nguyen, qui concernait exclusivement SECTRA et PIVAX. Nous avons voulu concevoir et construire un outil générique de type SAAS destiné à la création de mini-dictionnaires à partir d'une base lexicale quelconque en JIBIKI-2. Concrètement, CREATDICO est un serveur paramétrable qui est un intergiciel de service dictionnairiques.

I.3.2.3 Conception et implémentation d'un intergiciel de lemmatisation

Pour la consultation d'une forme d'un mot, et *a fortiori* pour la création d'un mini-dictionnaire à partir d'un fragment de texte, il est nécessaire, pour la plupart des langues, de commencer par lemmatiser la ou les formes en argument, ce qui peut exiger une étape préalable de segmentation (par exemple en chinois). Or, jusqu'ici, la lemmatisation était implémentée de façon ad hoc. Par exemple, PAPILLON-CDM offre la recherche avec lemmatisation en français, mais cette fonction n'est pas disponible dans les autres bases lexicales construites sur JIBIKI-1.

C'est pourquoi nous avons conçu et réalisé un intergiciel de lemmatisation, LEXTOH, qui intègre les services des lemmatiseurs, des segmenteurs et des racineurs disponibles, pour un nombre de langues inconnu *a priori*. D'autre part, LEXTOH permet d'accéder à une ou plusieurs BDLex en utilisant un ou plusieurs services de lemmatisation.

Cet outil pourrait être utilisé pour développer des systèmes MOSES (ex. MOSES-LIG) pour créer les corpus d'apprentissage nécessaire à la mise en œuvre de l'approche vectorielle[1] pour la segmentation et l'annotation de corpus, pour la consultation dictionnairique avancée (ex. dans les instances de JIBIKI), et pour la création de mini-dictionnaires (envoyant des appels à CREATDICO).

I.3.2.4 Recherche et intégration d'un système de gestion de tâches entre serveurs ou agents à gros grain

Comme dit plus haut, c'est une partie du thème des "flux numériques en TALN", qui est devenu un domaine reconnu. Nous nous en sommes rendu compte à cause de notre étude sur

[1] Un mot est remplacé par un vecteur contenant son lemme et quelques attributs morphosyntaxiques.

METRICC, qui nous a fait étudier UIMA[1], qui permet de gérer les flux de ressources et les flots de données linguistiques.

On a besoin d'une fonction de gestion des travaux pour tous les outils de notre architecture d'accès multilingue et de traduction avec post-édition contributive. Par exemple, SECTRA en a besoin pour les appels de TRADOH pour les prétraductions. PIVAX en a besoin pour toutes les demandes de services dictionnairiques. LEXTOH en a besoin pour les appels à des services de lemmatisation, etc.

Pendant le projet TRAOUIERO (2011-2012), notre équipe avait prévu de développer un tel outil, par extension et adaptation de BLEXISMA, mais cela n'avait pas pu être fait, l'unique personne compétente étant prise sur un autre projet.

Remarques finales

Nous avons cherché à considérer ces quatre thèmes selon trois axes : la demande industrielle, un nouvel apport pour la recherche, et le développement d'offres de services.

[1] La plate-forme APACHE UIMA (Unstructured Information Management Architecture) est utilisée dans le projet METRICC pour l'analyse de corpus et l'extraction de terminologie. Voir https://uima.apache.org

Chapitre II Extensions fonctionnelles et opérationnelles : de PIVAX-1 à PIVAX-2

Introduction

Pour améliorer la première version de PIVAX, il fallait d'abord passer à l'échelle, et ajouter certaines fonctionnalités. Par exemple, au niveau de JIBIKI, on n'avait pas de fonction générique de recherche avec lemmatisation préalable, et on ne pouvait pas choisir le(s) volume(s) cible(s). D'autre part, on avait essayé d'utiliser PIVAX-1 dans le projet OMNIA, en lui faisant produire une "grammaire-Q" réalisant l'annotation des mots d'un paragraphe par des UW UNL. Mais PIVAX-1 était trop faible pour des applications "programmatoires" de ce type. Il fallait aussi augmenter la vitesse de recherche en consultation humaine.

Pour améliorer PIVAX-1, nous avons travaillé (1) sur le niveau sous-jacent, c'est-à-dire sur la plate-forme JIBIKI, et (2) sur l'extension du type "PIVAX" de BDLex.

Le premier travail (avec M. Mangeot) au niveau de la plate-forme JIBIKI de construction de BDLex "métier" contributives, a été un travail préliminaire et plutôt technique, mais le présenter permet de comprendre les méthodes et techniques sur lesquelles JIBIKI s'appuie (par exemple, le "tout XML"), les fonctionnalités introduites à ce niveau (par exemple, liste non bornée, appel des lemmatiseurs de façon générique, choix des volumes cibles), sans devoir écrire du code spécifique pour chaque type de BDLex construite avec JIBIKI-2.

Au niveau de la BDlex, on a réimplémenté PIVAX-1 en utilisant JIBIKI-2. Cette partie a permis de trouver comment mettre réellement à disposition "de la communauté" les ressources lexicales rassemblées dans PIVAX-1 par le projet ANR TRAOUIERO et dans le cadre d'une coopération CNRS-ANR avec l'IPPI (Moscou).

Ce chapitre est divisé en quatre sections : (1) présentation de JIBIKI-1, (2) présentation de PIVAX-1, (3) extensions fonctionnelles apportées par le passage à JIBIKI-2, et (4) PIVAX-2 : opérationnalisation et extension de PIVAX-1.

II.1 JIBIKI-1

II.1.1 Présentation de JIBIKI-1

On a mentionné la plate-forme JIBIKI aux I.1.3.2.3 et I.1.4.3. Elle a été produite suite aux évolutions du projet PAPILLON (deux types différents de BDLex, celui de PAPILLON-CDM, et celui de PAPILLON-NADIA), dans le but de faciliter la création de BDLex de types variés. Le but initial de JIBIKI était donc de séparer un système de gestion de BDLex en deux parties : une partie logicielle commune et une partie spécifique à chaque type différent de BDLex.

Cette plate-forme générique permet la construction de sites Web contributifs dédiés à la construction de bases lexicales multilingues. Elle offre deux interfaces : des formulaires Web pour les humains, et une API REST pour les programmes. Les manipulations des ressources par ces interfaces sont enregistrées directement dans la BDLex.

Le code est disponible en source ouvert et téléchargeable gratuitement par SVN sur https://ligforge.imag.fr/projects/jibiki/.

II.1.2 Architecture de JIBIKI-1

II.1.2.1 Spécification

La plate-forme JIBIKI permet de traiter presque toutes les ressources lexicales de type XML contenant différentes microstructures et macrostructures.

JIBIKI offre de nombreuses fonctionnalités prêtes à l'emploi : import d'un dictionnaire ou d'un volume, création et édition d'entrées, gestion des contributions par l'administrateur, et recherche dans les bases lexicales.

Il y a trois types de recherche : (1) la recherche simple, par mot-vedette et langue source, (2) la recherche par parcours d'un volume, similaire à la consultation d'un dictionnaire papier, et (3) la recherche avancée, qui permet de consulter un mot en précisant une ou plusieurs conditions. Chaque condition est une combinaison de trois éléments : un attribut (ex. le mot-vedette, la variante, la date de création, la date de modification, le statut, l'identifiant etc.), un opérateur (ex. égale, commence par, plus grand que, plus petit que, etc.) et une variable. Par exemple,

```
  mot-vedette commence par "dicti"
& dictionnaire est "FeM"
& langue source est "français"
& variante est "dico"
& date de création est après le "11/10/2013"
& date de création est avant le "11/10/2015"
```

Pour importer un nouveau dictionnaire dans une BDLex de JIBIKI, on n'a pas besoin de modifier les programmes, mais la ressource lexicale doit être en XML, et on doit préparer un certain nombre de fichiers.

1. Le fichier de métadonnées du dictionnaire est le descripteur de sa macrostructure (description des volumes du dictionnaire et de leurs relations).
2. Il y a un fichier de métadonnées pour chaque volume, spécifiant les chemins des informations relatives dans XML en utilisant XPATH. Ces informations relatives sont reliées aux pointeurs CDM (voir la *Définition 17*). Ce fichier permet d'aligner la microstructure de chaque volume avec la microstructure CDM.
3. Un fichier XSD[1] définit la structure XML attendue en entrée ; ce fichier permet de créer automatiquement l'interface d'édition sur le Web.
4. Optionnellement, un formulaire XHTML permet de définir une interface d'édition spécifique. La plate-forme JIBIKI peut créer une interface spécifique en utilisant le fichier XSD, mais créer une interface spécifique pour un certain type d'utilisateur permet d'améliorer l'interaction.
5. On peut associer une feuille de style en XSL[2] à un volume ou à un dictionnaire ; elle permet de personnaliser l'affichage de consultation.
6. Ces fichiers peuvent être préparés automatique avec IPOLEX[3].

[1] XSD (XML Schema Definition) est un langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. XSD est plus puissant que DTD et est lui-même un langage XML.

[2] XSL (eXtensible Stylesheet Language) permet de transformer un document XML en une autre forme.

[3] IPOLEX est un entrepôt de données lexicales, développé par M. Mangeot. Cet outil offre des services d'analyse automatique de ressource lexicale pour une ressource importée, puis crée les fichiers servant à JIBIKI. Voir https://papillon.imag.fr/ipolex/Pages/index.php

II.1.2.2 **Implémentation**

On a déjà mentionné l'architecture de JIBIKI et son environnement de développement au I.1.4.3. Son architecture à trois couches facilite l'ajout et la modification des interfaces pour une nouvelle instance. Elle permet aussi de s'abstraire de la manière dont les données sont stockées.

Pour gérer les différentes microstructures, M. Mangeot a créé une microstructure virtuelle en CDM. Pour chaque pointeur CDM, on indique le chemin XPATH vers l'élément correspondant dans la microstructure XML. La Figure 11 donne un exemple d'une entrée en XML et le contenu des éléments XML correspondants.

Source de l'entrée "vedette"

```
<g:volume>
  <g:article g:id="fra.instruction.1847645">
    <g:vedette>
      <g:h-aspire>false</g:h-aspire>
      <g:mot>instruction</g:mot>
      <g:prononciation>\ɛ̃s.tʁyk.sjɔ̃\</g:prononciation>
      <g:grammaire>
        <g:cat-gram>s.</g:cat-gram>
        <g:genre-nbr>f</g:genre-nbr>
      </g:grammaire>
    </g:vedette>
  </g:article>
</g:volume>
```

Correspondances avec CDM

```
<cdm-elements>
  <cdm-volume xpath="/g:volume"/>
  <cdm-entry xpath="/g:volume/g:article"/>
  <cdm-entry-id xpath="/g:volume/g:article/@g:id"/>
  <cdm-headword xpath="/g:volume/g:article/g:vedette/g:mot/text()" d:lang="fra"/>
  <cdm-pronunciation xpath="/g:volume/g:article/g:vedette/g:prononciation/text()"
d:lang="fra"/>
  <cdm-pos xpath="/g:volume/g:article/g:vedette/g:cat-gram/text()" d:lang="fra"/>
</cdm-elements>
```

Figure 11 : Exemple d'utilisation de CDM

Les CDM permettent d'importer directement un nouveau dictionnaire sans le modifier, pourvu qu'il soit en XML. Voir la Table 2.

Table 2 : Valeurs de pointeurs CDM dans différents dictionnaires

Dictionnaire / Pointeur	FEM	OHD[1]	JMDICT[2] [Breen, J., 2004]
Volume	/volume	/volume	/JMdict
Entry	/volume/entry	/volume/se	/JMdict/entry
Entry ID	/volume/entry/@id		/JMdict/entry/ent_seq/text()
Headword	/volume/entry/headword/text()	/volume/se/hw/text()	/JMdict/entry/k_ele/keb/text()
Pron	/volume/entry/prnc/text()	/volume/se/pr/ph/text()	
PoS	//sense-list/sense/pos-list/text()	/volume/se/hg/ps/text()	/JMdict/entry/sense/pos/text()
Domain		//u/text()	
Example	//sense1/expl-list/expl/fra	//le/text()	/JMdict/entry/sense/gloss/text()

[1] OHD est abréviation de "Dictionnaire Oxford-Hachette", qui est un dictionnaire français-anglais.

[2] JMdict est un dictionnaire japonais-anglais, partiellement adapté à d'autres langues cibles.

II.1.3 Types de BDLex déjà développés en JIBIKI-1 en 2011

Nous avons déjà mentionné ces projets au I.2.1.1. Nous présentons ici plus en détail leur macrostructure et leur microstructure.

II.1.3.1 PAPILLON

On a déjà mentionné les structures de deux sous-projets (PAPILLON-NADIA et PAPILLON-CDM) au I.1.3.2.3.

Le projet NADIA [Sérasset, G., 1994a], lancé au GETA à la suite du projet MULTILEX ESPRIT, a pour but de construire une BDLex pour les dictionnaires de TALN et de TA. Sa macrostructure est Pivot.

La microstructure d'un volume est définie par un schéma XML spécifique avec variantes. Ces schémas XML redéfinissent l'élément "article" du schéma DML (Dictionary Markup Language)[1]. Par exemple, on doit redéfinir la liste des catégories pour chaque volume : le thaï n'a pas d'adjectifs, le japonais en distingue plusieurs, etc. Voici un morceau du schéma spécifique au japonais pour la redéfinition des catégories et un exemple d'une lexie.

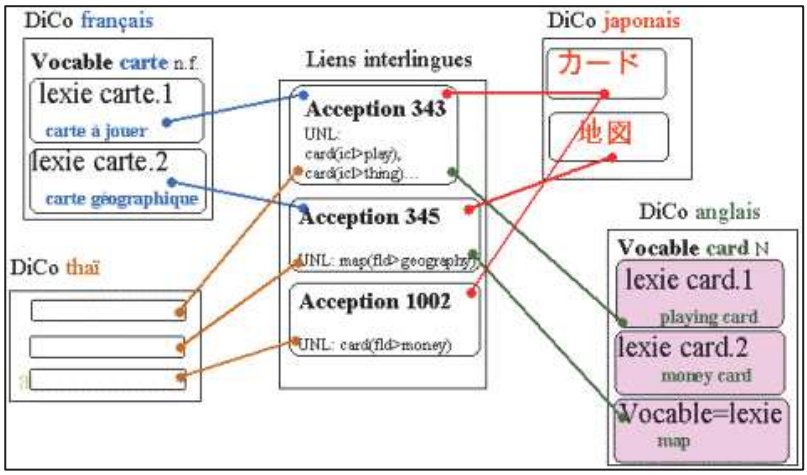


Figure 12 : Macrostructure du dictionnaire PAPILLON-NADIA

Schémas XML spécifique redéfini	Entrée

`<redefine schemaLocation="papillon.xsd">`
`  <simpleType name="posType">`
`  <restriction base="dml:posType">`
`  <!-- settôgo, prefix -->`
`  <enumeration value="接頭語"/>`
`  <!-- setsubigo, suffix -->`
`  <enumeration value="接尾語"/>`
`  <!-- zôgoseibun, productive element -->`
`  <enumeration value="造语成分"/>`
`  <!-- meishi, noun -->`
`  <enumeration value="名词"/>`
`  <!-- keishikimeishi, formal noun -->`
`  <enumeration value="形式名词"/>`
`  ...`
`  </restriction>`
`  </simpleType>`
`</redefine>`
	`<lexie id="洗う$1" basic="true">`
	`  <headword hn="1">洗う</headword>`
	`  <kun-yomi>あらう</kun-yomi>`
	`  <pos>他動詞</pos>`
	`  <language-levels>`
	`  <politeness grade="neutral"/>`
	`  <usage grade="NA"/>`
	`  <reference grade="NA"/>`
	`  </language-levels>`
	`  ...`
	`</lexie>`

Figure 13 : Exemple de microstructure du projet PAPILLON-NADIA

[1] DML est un espace de noms (namespace) de XML, publié sur http://www-clips.imag.fr/geta/services/dml/, et présenté dans la thèse de M. Mangeot. DML permet de décrire la structure générale des ressources lexicales.

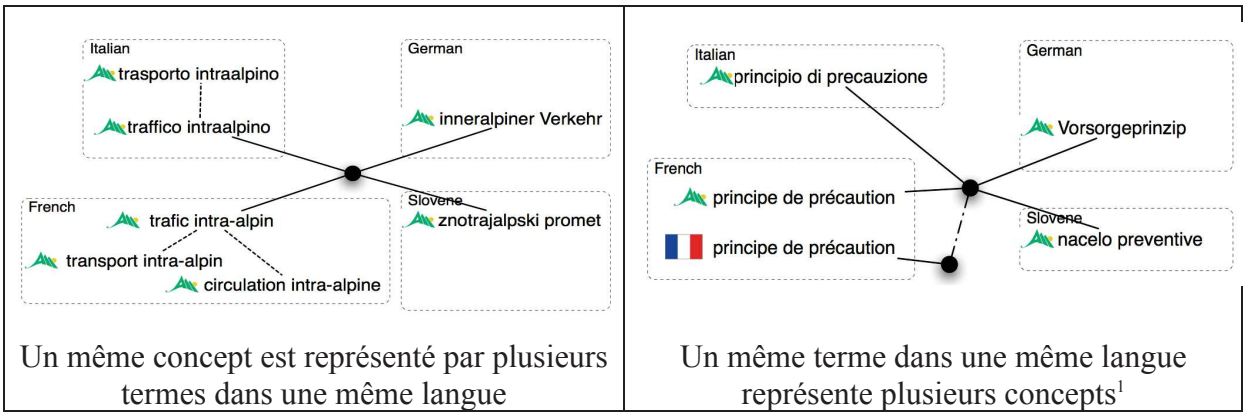
Quant au projet PAPILLON-CDM [Boitet, C. et al., 2002], on a mentionné sa macrostructure au I.1.3.2.3 et I.1.4.2, et sa microstructure au I.1.4.2 et II.1.2.2. Pour les exemples, voir la Figure 11 et la Table 2.

II.1.3.2 LEXALP

LEXALP [Sérasset, G. et al., 2006 ; Sérasset, G., 2008] a été motivé par trois raisons principales :

1) un même mot représente différents concepts juridiques,
2) on utilise des mots différents pour représenter le même concept,
3) une traduction générale directe peut représenter un concept juridique différent de celui associé au mot source.

Au niveau de l'implémentation, G. Sérasset a choisi une macrostructure PIVOT, bien adaptée au fait qu'un concept peut être exprimé par plusieurs termes dans une même langue, et qu'un terme peut représenter plusieurs concepts dans une même langue. Voir la Figure 14. Tous les volumes monolingues partagent une même microstructure en XML. Dans le volume des axies, chaque axie peut être reliée à plusieurs entrées monolingues grâce à un élément (*termref*) et à d'autres acceptions interlingues grâce à un autre élément (*axieref*). On trouvera des exemples de microstructure dans l'Annexe 5.



Un même concept est représenté par plusieurs termes dans une même langue

Un même terme dans une même langue représente plusieurs concepts[1]

Figure 14 : Structure de terminologie de LEXALP

II.1.3.3 GDEF

Contrairement aux deux premiers projets, GDEF utilise une macrostructure directe. Il s'agit d'un dictionnaire bilingue. Dans ce type de BDLex, on a souvent deux volumes bilingues : un volume langue A (LgA) → langue B (LgB) et un volume miroir LgB → LgA. Cette macrostructure ressemble à un dictionnaire bilingue imprimé.

Dans le cadre du projet GDEF, on a un seul sens (est→fra) mais deux volumes monolingues : un volume estonien et un volume français. Il n'y a pas de volume d'axies. On a des liens du volume estonien vers le volume français, mais pas l'inverse.

Les entrées du volume estonien contiennent :

- l'essentiel des informations estoniennes,
- certaines gloses françaises (voir ci-dessous),
- certains liens vers leurs équivalents français (voir ci-dessous),
- des exemples estoniens et français.

[1] La définition de *principe de précaution* dans la Convention Alpine est différente de sa définition dans le droit français.

Les traductions des articles estoniens sont réalisées par des liens vers le volume français, qui contient une telle traduction sous la forme du PCDATA d'un nœud `<g:mot-princ>`. Les gloses précisant les sens du mot sont marquées par les balises `<g:avant>` (mot ou terme précédant le mot-vedette de la traduction française) et `<g:après>` (mot ou terme suivant le mot-vedette de la traduction française).

Le volume français ne contient que les mots-vedettes et les informations grammaticales du français (ex. genre des substantifs, pluriels irréguliers des substantifs et des adjectifs, féminins irréguliers des adjectifs, h aspiré, etc.).

Par exemple, le mot *jõevesi* en estonien se traduit par *eau* en français. Plus précisément, il s'agit de l'eau d'une rivière ou d'un fleuve.

Figure 15 : Un article du GDEF et sa structure XML

II.1.3.4 MOTÀMOT

Ce projet a été conçu au départ pour construire à terme une base lexicale de toutes les langues d'Asie du Sud–Est. La base actuelle contient pour l'instant des données dans seulement deux langues, le khmer et le français, avec une macrostructure PIVOT (un volume d'axies). Tous les volumes sont stockés en XML.

Pour la microstructure, chaque entrée est un vocable qui regroupe des lexies. Une entrée se compose d'un mot-vedette, d'une prononciation, de la catégorie grammaticale, et d'une liste de lexies.

Chaque lexie est décrite par une formule sémantique (voir l'exemple ci-dessous) ou par une glose libre. Il y a un lien vers une entrée (une axie) du volume pivot. Il y a aussi parfois un domaine, une liste d'exemples, des expressions idiomatiques, et un champ générique utilisé pour stocker des informations supplémentaires.

La Figure 16 reprend un exemple donné dans [Mangeot, M. & Nguyen, H. T., 2009]. Il y a un seul sens pour cet article *abandonner*, caractérisé par une formule sémantique (par exemple, *action sur un objet : humain ou animal X ~ entité Y*) et une glose libre (*laisser tomber quelque chose*).

La microstructure du volume pivot est très simple : une entrée axie contient un ensemble de liens vers des entrées des volumes monolingues. En cas de synonymie, il peut y avoir plusieurs liens d'une axie vers le même volume monolingue.

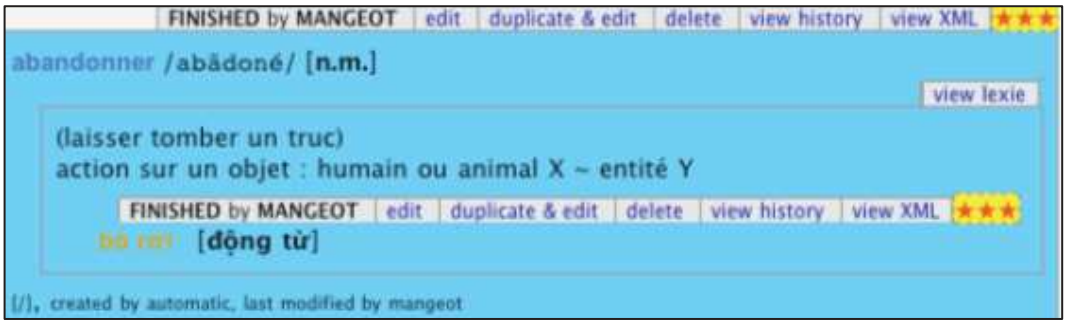


Figure 16 : Article abandonner dans MOTÀMOT

II.1.3.5 PIVAX-1

Comme nous voulons présenter PIVAX-1 de façon très détaillée, nous lui consacrons toute une section.

II.2 PIVAX-1

PIVAX-1 [Nguyen, H.-T. et al., 2007] est une nouvelle macrostructure à trois niveaux, définie et implémentée par H.-T. Nguyen dans le cadre de sa thèse [Nguyen, H.-T., 2009]. Elle permet d'implémenter plusieurs volumes monolingues pour chaque langue naturelle ou artificielle. Son implémentation a été basée sur Jibiki-1.

II.2.1 Motivations

Nous citons ici [Nguyen, H.-T., 2009]. « Ce projet a été motivé par les besoins de partage de données lexicales de systèmes de TA. Ce besoin vient du désir d'utilisateurs de systèmes commerciaux comme SYSTRAN, Reverso, METAL, etc. de partager leurs dictionnaires entre eux et entre systèmes[1]. Cependant, l'information lexicale à partager est limitée à cause de la protection de propriété pour des ressources importantes qui coûtent cher à développer, donc l'effort de ce côté-là n'a pas tout à fait réussi. La solution proposée repose sur l'idée d'avoir plusieurs volumes par espace lexical d'une langue. En plus, on souhaite développer une BDLex universelle pour tous les systèmes de TA, avec des architectures linguistiques arbitraires, mais c'est très difficile à réaliser, parce que les types d'information et leur organisation sont trop différents. Donc nous limitons le problème à la classe des systèmes de TA utilisant un même "espace lexical pivot". C'est pourquoi nous avons conçu et développé Pivax-1, un système contributif en ligne de BDLex qui permet de créer, maintenir et gérer les ressources lexicales de systèmes de TA basés sur un "pivot lexical". »

II.2.2 Structure de PIVAX-1

On présente la macrostructure et la microstructure de PIVAX-1 avec les exemples donnés dans [Nguyen, H.-T., 2009].

II.2.2.1 Macrostructure

La macrostructure de PIVAX-1 contient trois types de volumes: lexie, axème et axie.

Pour chaque langue naturelle présente ou chaque interlingua, on a un ou plusieurs volumes de lexies et leurs informations associées, et un unique volume d'axèmes. Une lexie correspond à un sens de mot dans un dictionnaire. Le terme "axème" est construit à partir de "acception" et

[1] H.-T. Nguyen a fait un long stage (décembre 2005 - novembre 2006) chez Systran au début de sa thèse.

"monolingue", mais en fait un axème relie des lexies synonymes dans une même langue, tandis qu'une acception (monolingue) correspond à un sens "en usage" ou "dans la langue". Dans la BDLex, il y a un volume unique d'axies (acceptions interlingues). Une axie relie des axèmes correspondant à des sens équivalents.

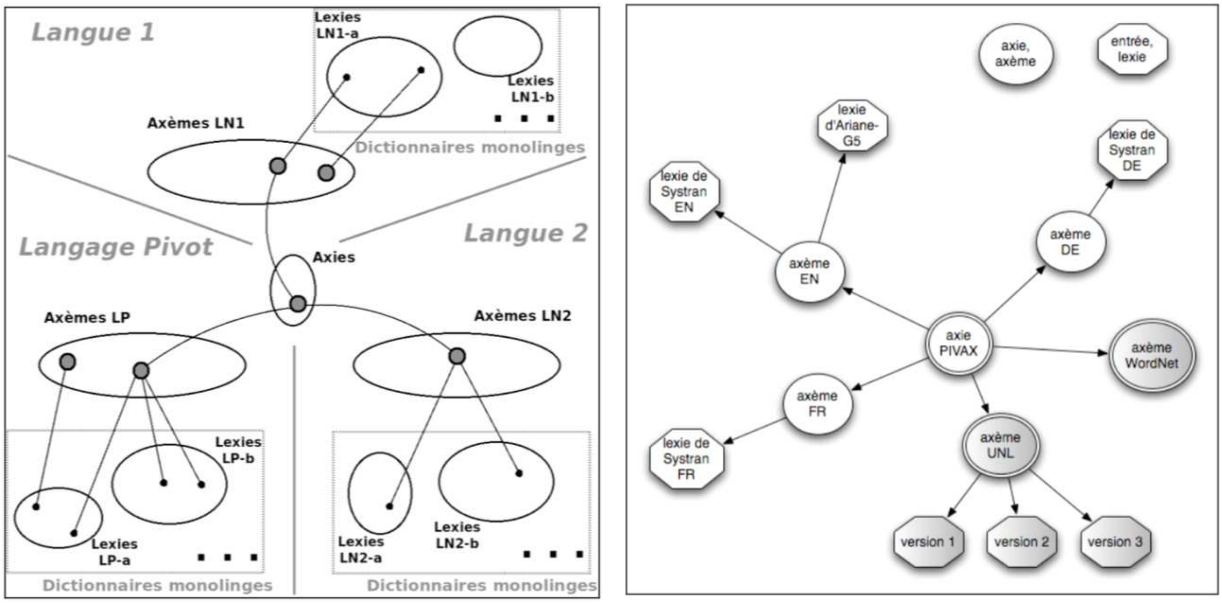


Figure 17 : Macrostructure de PIVAX et exemples de volumes

II.2.2.2 Microstructure

Les axèmes et axies sont simplement des liens, qui sont représentés chacun par l'ensemble des identificateurs de lexies ou axèmes correspondants.

Pour simplifier la programmation, on impose en PIVAX une même microstructure pour tous les volumes d'un même espace lexical.

Dans les volumes de LN, une entrée contient un lemme, sa classe (POS), son identificateur de sens de mot (le cas échéant), un commentaire (pour les autres développeurs) et le détail approprié de l'information propre à chaque volume et ses codes. Chaque entrée peut contenir des métadonnées (la date de modification, l'auteur, l'état de traitement, le niveau de protection, définissant les parties accessibles par le public : droit d'écriture, de lecture et d'exécution).

```
<d:data>
  <p:lexie p:id="lexie.systran.test.12004" p:process_status="UNPROCESSED" p:status="UNKNOWN"
    p:owner="Systran" p:score="0.00003">
    <p:lemma p:access="Public">fier</p:lemma>
    <p:class p:access="Public">Adj</p:class>
    <p:comment> "fier" can also be a V ("se fier à") </p:comment>
    <p:proper_information p:access="Hidden">
    /* Systran proprietary codes */
    </p:proper_information>
  </p:lexie>
</d:data>
```

Pour les volumes interlingues (volumes d'UW, volumes implémentant une ontologie), les entrées sont toujours définies au niveau sémantique, et peuvent donc toujours être appelées "lexies". Plus précisément, la microstructure d'un volume d'UW (lexèmes interlingues d'UNL) se compose de : mot-vedette, contenu, définition, exemples, note et autre information[1].

[1] C'est présenté comme dans sa thèse, mais dans son programme, tout n'est pas réalisé.

```
<uw id="unl.upp.abandon.7" status="UN_KNOWN">
        <headword>abandon</headword>
        <pos>VERB</pos>
        <content>abandon(icl>do, agt>thing, obj>thing)</content>
        <definition>stop maintaining or insisting on </definition>
        <examples>
                <example>of ideas or claims</example>
                <example>He abandoned the thought of asking for her hand in marriage</example>
                <example>Both sides have to abandon some claims in these
                  negotiations</example>
        </examples>
        <more-info>
                <info> </info>
        </more-info>
</uw>
```

```
<cdm-elements>
        <cdm-volume xpath="/unl_volume"/>
        <cdm-entry xpath="/unl_volume/uw"/>
        <cdm-entry-id xpath="/unl_volume/uw/@id"/>
        <cdm-headword xpath="/unl_volume/uw/headword/text()" d:lang="unl" index="true"/>
        <cdm-pos xpath="/unl_volume/uw/pos/text()" d:lang="unl"/>
        <cdm-definition xpath="/unl_volume/uw/content/text()" d:lang="unl" index="true"/>
        <cdm-example xpath="/unl_volume/uw/examples/example/text()" d:lang="unl"/>
</cdm-elements>
```

Figure 18 : Exemple d'un volume UNL de PIVAX-1 et de ses pointeurs CDM

II.2.2.3 Algorithme de calcul des liens

H.-T. Nguyen a créé des liens d'axies vers axèmes et d'axèmes vers lexies, mais pas dans les directions inverses. Dans PIVAX-1, il n'y a donc pas de liens à partir des volumes de lexies. Dans son implémentation de l'algorithme de calcul des liens de la Figure 19, il a créé quatre méthodes différentes pour trouver les traductions d'une entrée source (lexie source).

1. `getAxemesPointingTo` : trouver les axèmes par les lexies dans les tables des pointeurs d'axèmes.
2. `getAxisPointingTo` : trouver les axies par les axèmes dans les tables des pointeurs d'axies.
3. `findAxemeByAxie` : trouver les axèmes par les axies dans les tables des pointeurs d'axies.
4. `findLexieByAxeme` : trouver les lexies par les axèmes dans les tables des pointeurs d'axèmes.

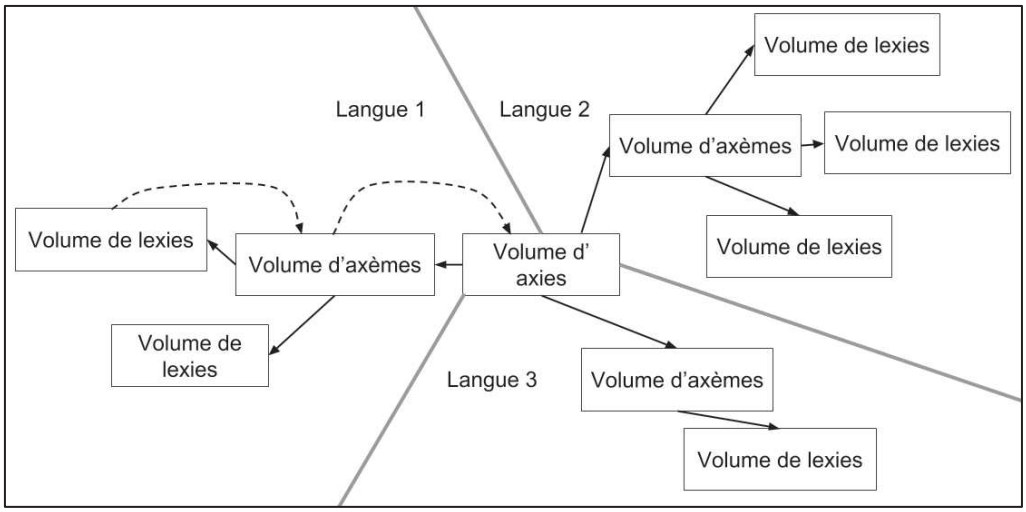


Figure 19 : Algorithme de calcul des liens dans PIVAX-1

II.2.2.4 **Interface**

Cette interface de consultation en colonnes a été inspirée par PARAX (voir I.1.4.1). Elle permet de modifier la largeur de chaque colonne (par – et +) et de changer l'ordre des colonnes (par < et >). Les entrées reliées à une même axie ou à un même axème sont présentées en leur affectant une même couleur.

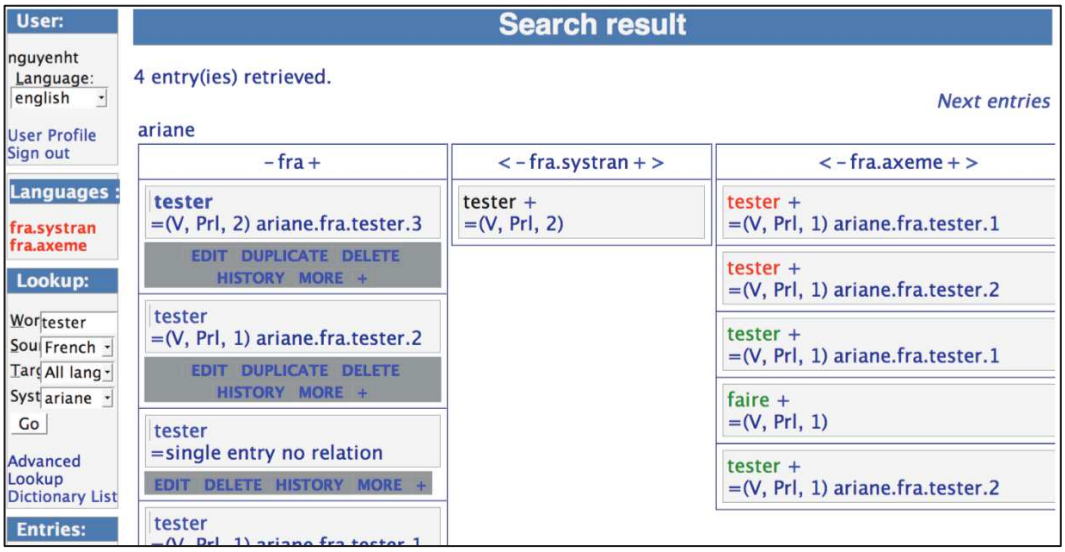


Figure 20 : Interface de consultation en colonnes de PIVAX

II.2.2.5 **Début de programmabilité**

L'objectif à long terme est de transformer PIVAX-1 en un EDL (Environnement de Développement Linguiciel) avec certains LSPLEX (Langage Spécialisé pour la Programmation Lexicale). Les utilisateurs (non-informaticiens) pourraient programmer des tâches spécifiques selon leurs besoins. H.-T. Nguyen a proposé un langage narratif [Bellynck, V., 2001] pour la manipulation des graphes lexicaux. Ce langage offre des opérations simples pour créer ou supprimer des objets (lexie, axème, axie et les liens). Il permet également de définir des macros pour grouper des suites d'opérations. Par exemple,

```
Simple :
Créer_nœud (entrée, type, méta-données) ;
Créer_lien (entrée_1, entrée_2, alias) ;
entrée = entrée dans un volume de PIVAX ;
type = lexie | axème | axie ;
méta-données = suite_d_attributs_valeurs ;
suite_d_attributs_valeurs (
     ( attribut = valeur ;) ? attribut = valeur ;
)
Supprimer_nœud (entrée);
Supprimer_lien (entrée_1, entrée_2);

Macro :
NOM_DE_MACRO (paramètres) (variables locales) {
     Suite_d_opérations_simples;
}
Suite_d_opérations_simples {
     (Simple ?;) Simple;
}
```

H.-T. Nguyen a spécifié une interface Web pour visualiser directement ces manipulations sur les graphes lexicaux (on a les graphes initiaux, ensuite on peut programmer et mettre à jour la

programmation, puis on obtient les graphes finals), mais il n'a pas eu le temps de les implémenter.

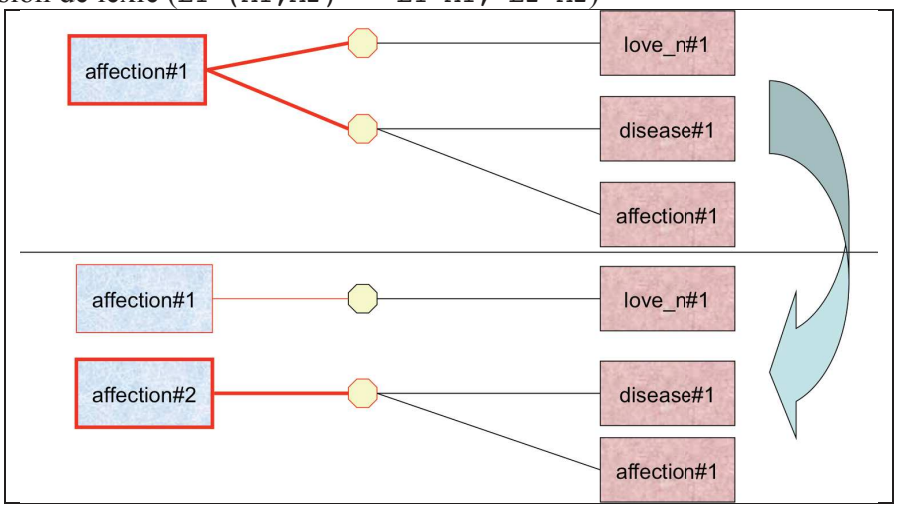
H.-T. Nguyen a aussi proposé un algorithme global pour unifier les UW d'UNL. Comme écrit plus haut, les UW sont créés par différents groupes, et du coup on a souvent des UW différents pour la même "acception interlingue". Par exemple :

- `access(icl>reach>do, agt>thing, obj>thing)` dans le volume U++C
- `access(icl>reach, agt>thing, obj>thing)` dans le volume UNLKB.

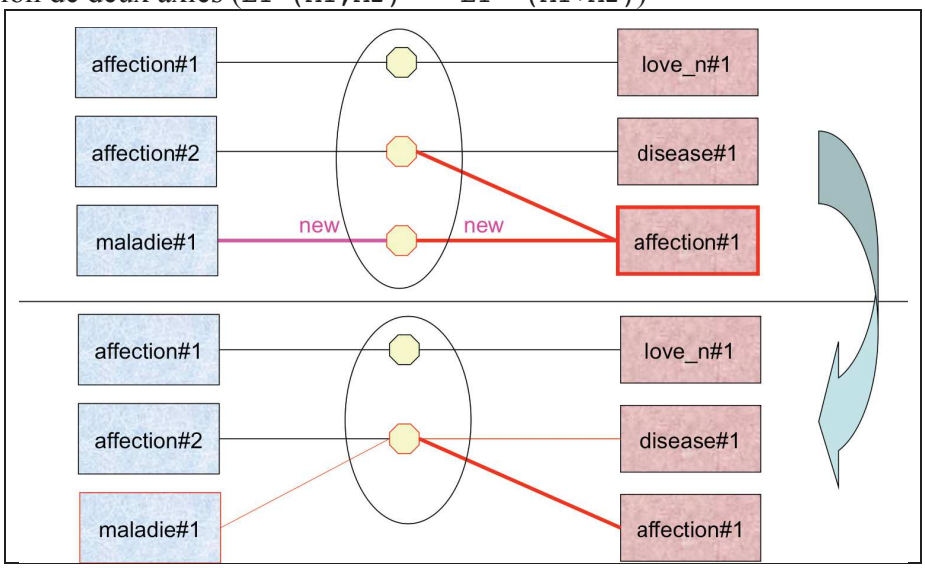
Son algorithme est basé sur trois niveaux de comparaison : seulement le mot-vedette, le mot-vedette et les restrictions, les informations spécifiques (pas implémenté).

D'autre part, H.-T. Nguyen a prototypé trois "règles de transformation" que M. Mangeot avait spécifiées dans sa thèse [Mangeot, M., 2001] en 2001 et que Ch. Boitet avait détaillées au séminaire du projet PAPILLON en 2005 [Boitet, C., 2005]. Ces règles ont pour but de restaurer la cohérence dans une BDLex de type PAPILLON-NADIA. Par exemple, une lexie ne peut pas être reliées à deux axèmes ou axies différents. Pour ce cas, on a trois actions possibles :

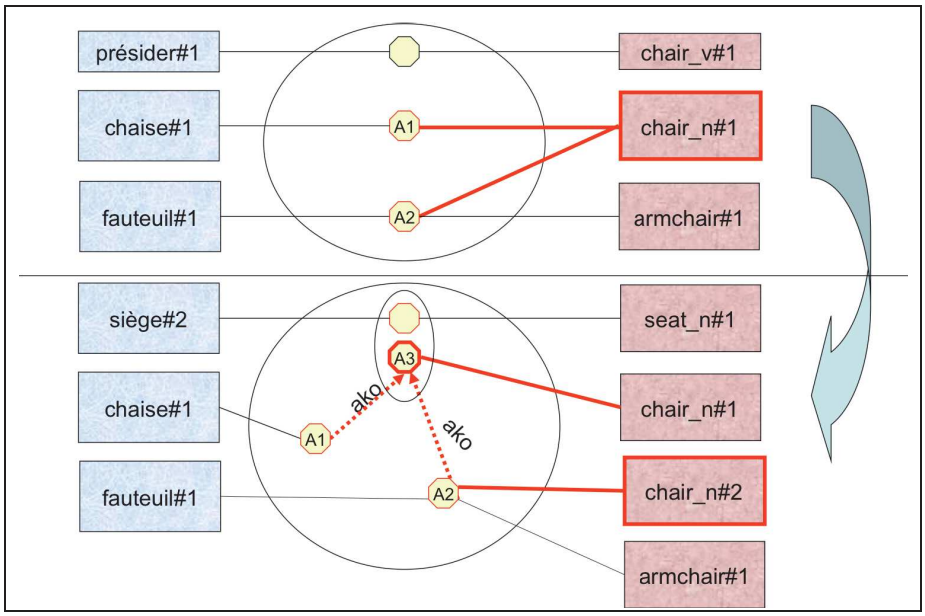
- Division de lexie (`L1–(A1,A2) → L1-A1; L2-A2`)



- Fusion de deux axies (`L1–(A1,A2) → L1 –(A1+A2)`)



- Création d'une autre axie et d'une lexie de plus (`L1–(A1,A2) → L1-A3 ; L2–A2 ; ako((A1,A2), A3)`)



II.2.3 Utilisations de PIVAX-1

II.2.3.1 Dans OMNIA

Dans le cadre du projet ANR OMNIA [Rouquet, D. & Nguyen, H. T., 2009], PIVAX-1 a été utilisé pour construire une "mini-grammaire" en systèmes-Q à partir d'une liste de lemmes.

On lemmatise d'abord les textes source en utilisant NOOJ ou DELAF, puis on annote les textes lemmatisés par des lexèmes interlingues, qui sont les UW d'U++C construits à partir de WORDNET.

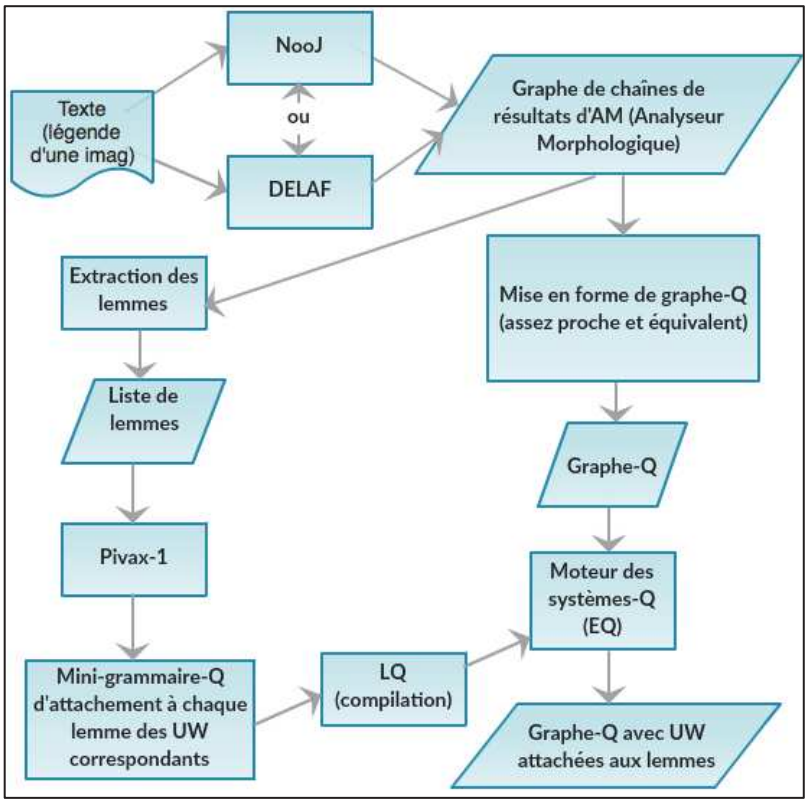


Figure 21 : Processus de traitement du projet OMNIA

Il s'agit non seulement de mots simples, mais aussi de mots composés. Par exemple, la suite de mots `waiting room` peut être segmentée de 3 façons : `waiting`, `room`, `waiting room`. Après la lemmatisation, on transforme le résultat en graphe-Q, qui est un format compact d'échange de résultats multiples de lemmatisation. Voir la Figure 22.

Les lemmes trouvés (simples et composés) sont envoyés à PIVAX-1, qui produit un dictionnaire lemmes-UW pour ce graphe-Q sous forme de règles-Q. L'étape suivante consiste à appliquer ce dictionnaire au graphe des lemmes, ce qui produit un graphe-Q contenant le graphe des lemmes, et de nouveaux arcs (et arbres) contenant une représentation des UW trouvés.

Dans cette application, PIVAX-1 fonctionne comme un serveur lexical qui produit un ensemble "d'articles" des UW sous la forme spécifique de règles-Q, pour chaque mot envoyé par le module d'annotation d'OMNIA.

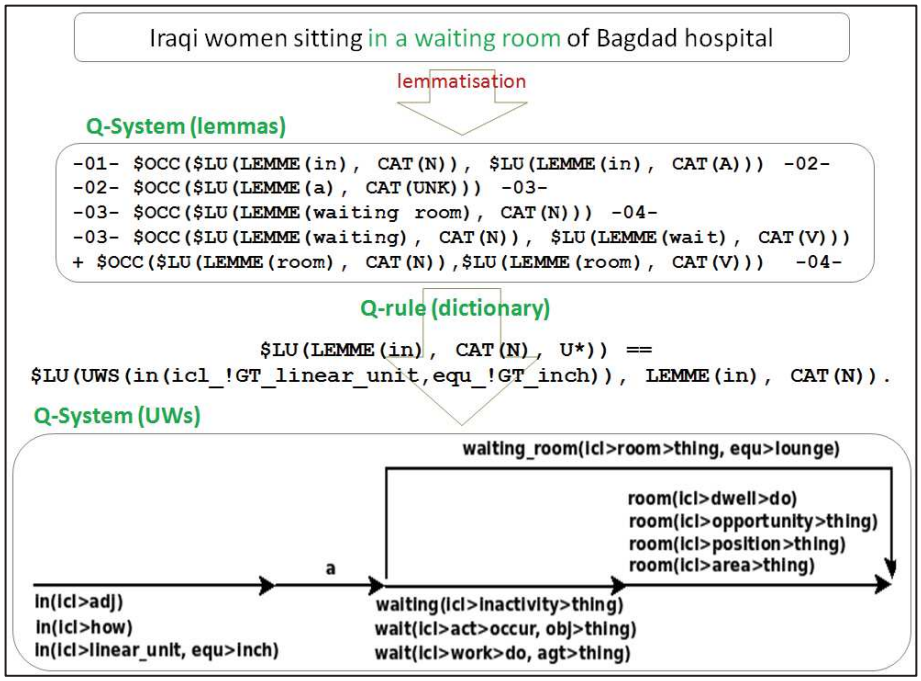


Figure 22 : Exemple d'annotation de texte dans OMNIA

II.2.3.2 Dans TRAOUIERO et GBDLEX-UW++

Pendant sa thèse, H.-T. Nguyen a utilisé PIVAX-1 pour en faire le support du projet U++C. Il a importé plusieurs ressources dans cette BDLex.

Table 3 : Ressources importées dans PIVAX-1 par H.-T. Nguyen

Ressource	*Type/Langue*	*Nombre des entrées*
UNL-DECO	UNL-FRA	39 389
PARAX	UNL-FRA	18 978
PARAX	UNL-CHN (données par le Pr. Shi en 2001)	9 315
PARAX	UNL-RUS	13 817
PARAX	UNL-ESP	3 833
UNLKB	UNL	21 618
UPM UW++	UW avec définition et exemples	207 009
UNDL	Projet EOLSS-UNL-FR/UW	21 354
IATE	Projet EOLSS/ENG, termes correspondant aux 21K UW récupérés par ROBODICO	255 305
IATE	Projet EOLSS/FRA, termes correspondant aux 21K UW récupérés par ROBODICO	258 175

La sous-tâche 4.4 GBDLEX-UW++ du projet TRAOUIERO avait pour but la construction d'une grande base lexicale avec lexèmes interlingues. Pendant ce projet, on a principalement construit deux ressources d'UW.

La ressource COMMONUNLDICT a été créée par le lexicographe et linguiste russe Vyacheslav Dikonov. Le type de BDLex de cette ressource (la macrostructure) est PIVAX-1. Il y a trois types d'entrées (vocables, axèmes et axies), et 720 K entrées au total. COMMONUNLDICT contient

8 langues (7 langues naturelles : français, anglais, hindi, malais, russe, espagnol, vietnamien et le langage UNL) et 17 volumes (8 volumes de données monolingues, 8 volumes d'axèmes monolingues et 1 volume d'axies "acceptions interlingues").

Tous les volumes d'un même type (vocable, axème et axie) ont la même microstructure. Ces structures ont été définies par V. Dikonov. Cette microstructure est différente de celle définie dans PIVAX-1.

La Figure 23 illustre **la microstructure d'un volume monolingue**. Chaque entrée de vocable permet de décrire toutes les informations détaillées, comme la partie du discours (POS), la prononciation, etc. Chaque vocable comprend une ou plusieurs lexies. Dans cette microstructure, l'attribut `entryref` permet de gérer les liens entre les lexies et les axèmes.

```
<p:vocable p:id="fra.Anglais.n">
    <p:lemma>Anglais</p:lemma>
    <p:pronunciation>ɑ̃.glɛ</p:pronunciation>
    <p:pos>n</p:pos>
    <!-- Sens : Relatif à l'Angleterre ou à ses habitants.-->
    <p:lexie p:id="CommonUNLDict.lexie.fra.Anglais.1">
      <p:entryref type="axeme" volume="CommonUNLDict_fra-axemes"
        p:idref="CommonUNLDict.axeme.fra.englishman(icl&gt;english&gt;person)"  lang="FRA"
        p:relation-mono="OTHER"/>
       <p:processors>
         <p:processor p:name="Ariane" p:access="Public">
           <p:procref type="entry" id="Anglais" var="CAT(CATN),GNR(MAS,FEM),N(NP)"
 lang="FRA"/>
         </p:processor>
       </p:processors>
    </p:lexie>
   <!-- Sens : Relatif à la langue anglaise.-->
   <p:lexie p:id="CommonUNLDict.lexie.fra.Anglais.2">
      ……
   </p:lexie>
</p:vocable>
```

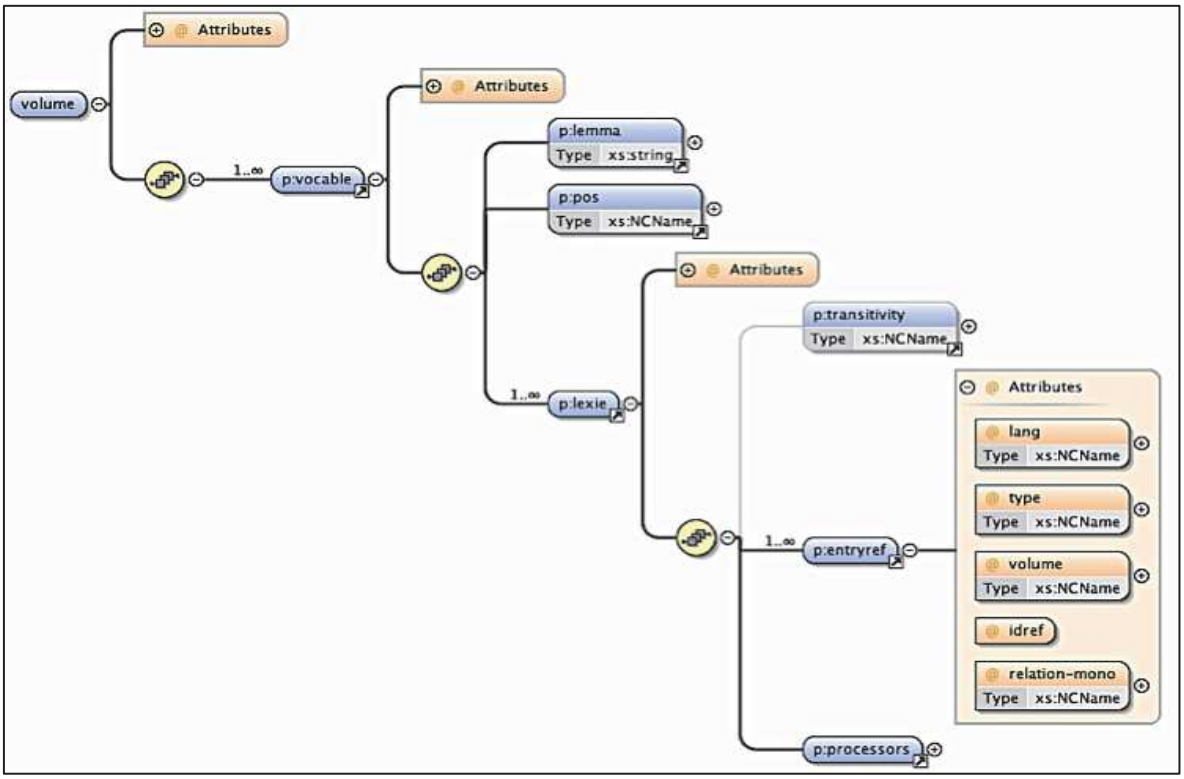


Figure 23 : Microstructure de volume monolingue de COMMONUNLDICT

La microstructure des volumes d'axèmes permet de décrire les liens avec les entrées de type lexie et les liens avec les entrées de type axie.

La microstructure des axies permet de décrire les liens avec les entrées de type axème.

Une autre ressource est UWPEDIA, créée par David Rouquet à partir de DBPEDIA. Cette ressource contient trois langues naturelles (français, russe, espagnol) et le langage UNL, avec 9M entrées au total. Sa macrostructure est la même que celle de PIVAX-1. Les microstructures sont simples.

Pour **un volume monolingue de vocables**, une entrée consiste en une lexie, un lemme, et un lien vers un axème.

Pour **un volume d'axèmes ou d'axies**, elle ne consiste qu'en des liens.

Une entrée du volume français

```
<p:vocable p:id="fra.Agriculture_de_précision">
        <p:lemma>Agriculture de précision</p:lemma>
        <p:entryref type="axeme" volume="UNL-UWpedia-FRA-axeme" lang="FRA"
          p:idref="http://uwpedia.org/fraaxeme/Precision_agriculture/Agriculture de précision"
          p:relation-mono="OTHER" />
</p:vocable>
```

Une entrée du volume d'axies

```
<p:axie p:id="http://uwpedia.org/axie/Carlo_Sforza" lang="UNL"
  p:writtenForm="Carlo_Sforza(iof>prime_minister)"
  unl:masterDef="Carlo_Sforza(iof>prime_minister{(icl>politician)})">
        <p:entryref type="axeme" volume="UNL-UWpedia-Axemes"
          p:idref="http://uwpedia.org/unlaxeme/Carlo_Sforza" lang="unl"/>
        <p:entryref type="axeme" volume="UNL-UWpedia-FRA-axeme"
          p:idref="http://uwpedia.org/fraaxeme/Carlo_Sforza/Carlo Sforza" lang="fra"/>
        <p:entryref type="axeme" volume="UNL-UWpedia-RUS-axeme"
          p:idref="http://uwpedia.org/rusaxeme/Carlo_Sforza/Сфорца, Карло" lang="rus"/>
        <p:entryref type="axeme" volume="UNL-UWpedia-ESP-axeme"
          p:idref="http://uwpedia.org/espaxeme/Carlo_Sforza/Carlo Sforza" lang="esp"/>
</p:axie>
```

Figure 24 : Exemples d'articles de UWPEDIA

Pour le nombre d'entrées de ces deux ressources, voir II.4.2.1. On avait prévu une BDLex de PIVAX-1 pour contenir ces données. Finalement, à cause de problèmes techniques (voir II.2.4), on n'a pas pu importer ces deux grosses ressources dans PIVAX-1 (on les a par contre importées dans PIVAX-2, voir II.4.3).

Grâce à l'organisation de la base PIVAX comme un grand réseau lexical, avec plusieurs volumes monolingues et un volume d'axèmes par "espace lexical", et un volume central d'axies, on pourra produire de nouveaux dictionnaires à partir des anciens dictionnaires bilingues par constructions d'axèmes et d'axies.

II.2.4 Qualités et limitations

Au début de cette thèse, il n'existait pas de serveur PIVAX-1 réellement utilisable. Après les expériences dans le projet OMNIA, le temps de réponse de PIVAX-1 pour un grand nombre de requêtes n'était pas satisfaisant. D'autre part, H.-T. Nguyen avait discuté les idées de la programmabilité de PIVAX par un linguiste-lexicographe-informaticien dans sa thèse, mais il manquait quand même des opérations pour des tâches spécifiques, par exemple, une sortie spécifique de mini-dictionnaires.

Par contre, PIVAX-1 est le premier système contributif de base de données lexicales en ligne qui permet de créer, maintenir et gérer les ressources lexicales de systèmes de TA basés sur un "pivot lexical", et hétérogènes dans le sens où leurs composants spécifiques à une langue peuvent être développés à différents endroits et avec différentes approches linguistiques et différents outils informatiques. La puissance de cette approche est surprenante : on a pu

"mettre dans PIVAX" des dictionnaires "normaux", des dictionnaires de TA, tous les dictionnaires UNL, WORDNET, et l'ontologie OMNIA (importée depuis PROTEGE[1]) grâce à sa structure à 3 niveaux.

II.3 Extensions fonctionnelles apportées par le passage à JIBIKI-2

Comme mentionné plus haut, au début de ma thèse, M. Mangeot avait prévu une grande mise à jour de JIBIKI. J'ai participé à cette évolution de JIBIKI-1 à JIBIKI-2 sous sa direction.

II.3.1 Liens riches

II.3.1.1 Motivations

Dans la BD de JIBIKI, il y a deux types de table : statique et dynamique. Les tables statiques sont créées pendant l'installation du système. Elles sont toujours les mêmes pour toutes les instances de JIBIKI. Il s'agit de 12 tables (voir la Figure 25, les tables en vert). Ces tables permettent de gérer le système (sauf les données lexicales) : gestion des utilisateurs, des groupes, des contributions, des feuilles de style.

Les tables dynamiques servent à stocker les données lexicales. Ces tables sont créées pendant la création ou l'import d'un nouveau volume. En JIBIKI-1, il y a deux tables pour chaque volume : une table des entrées et une table d'indexation (voir la Figure 25, les tables en bleu). Les noms de ces deux tables sont créés à partir du nom du volume. Par exemple, pour le volume `CommonUNLDict_fra`, nous avons la table `commonunldictfra` (table des entrées) et la table `idxcommonunldictfra` (table d'indexation).

La table des entrées contient quatre colonnes : identification, mot-vedette, code XML et version.

La table d'indexation (voir la Figure 26) contient toutes les informations indexées en utilisant le fichier de métadonnées (par des pointeurs CDM-CLASSIQUE ou des pointeurs particuliers). La colonne `entryid` associe un identificateur à une entrée dans la table des entrées. Dans cette version, il y a un seul pointeur qui sert au lien de traduction, appelé `cdm-translation-ref`.

On ne peut que créer un lien entre la source et l'identificateur d'une traduction, et on ne peut pas ajouter d'informations supplémentaires, notamment spécifier le volume cible du lien ! La possibilité de traiter des liens plus complexes manque vraiment.

[1] Protégé est un éditeur pour la création d'ontologies. Voir http://protege.stanford.edu

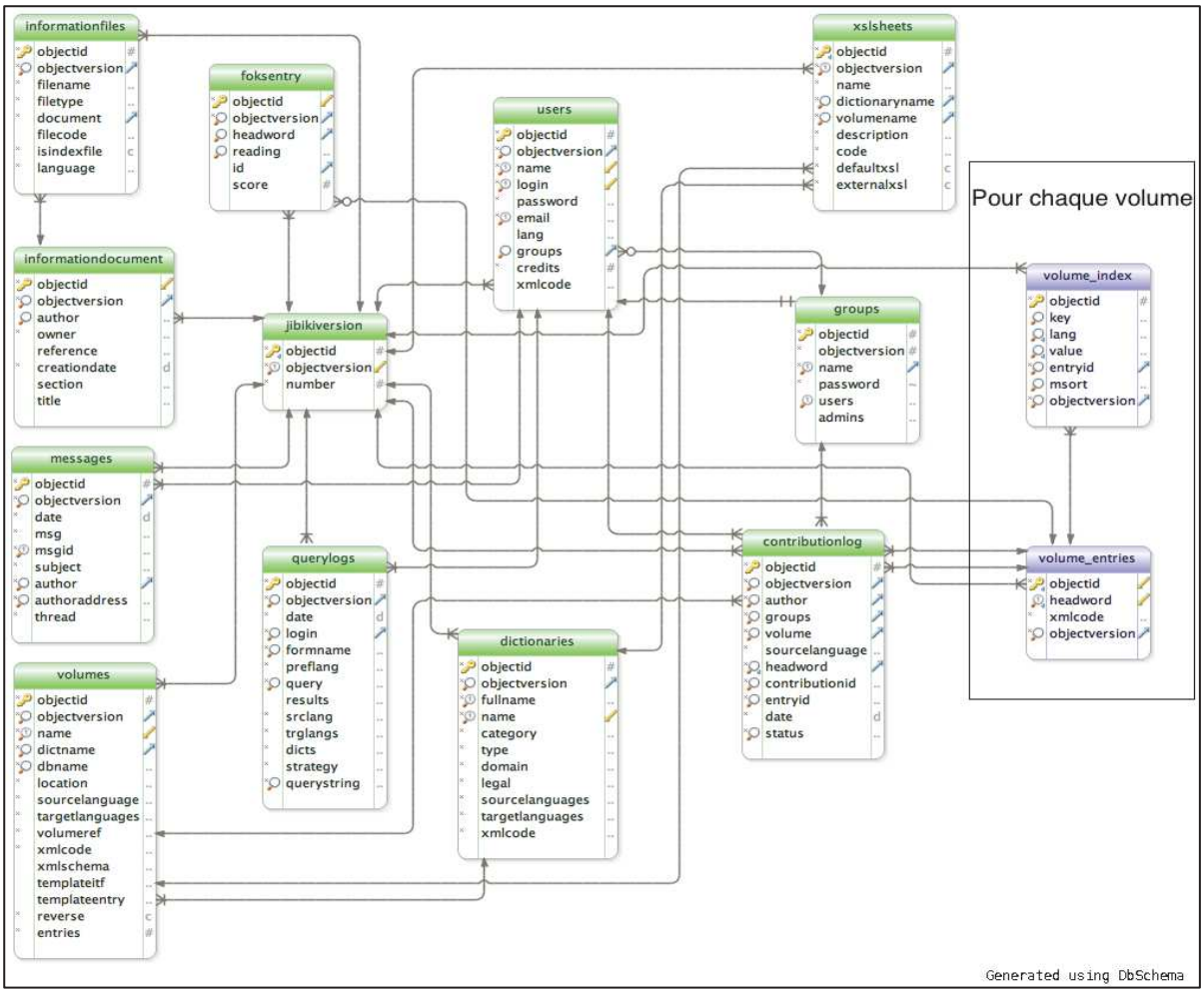


Figure 25 : Schéma de la base de données de JIBIKI-1

	key character varying(255)	lang character var	value character varying(255)	entryid numeric(19,0	msort character var	objectid [PK] numeric(	objectversion integer
1	cdm-pos	eng	v	1774081	engv	1774083	0
2	cdm-headword	eng	eat	1774081	engeat	1774084	0
3	cdm-contribution-creation-da	#NA	2015/03/18 15:30:17	1774081	#NA2015/03/	1774085	0
4	cdm-contribution-author	#NA	automatic	1774081	#NAautomati	1774086	0
5	cdm-contribution-status	#NA	not finished	1774081	#NAnot fini	1774087	0
6	cdm-contribution-id	#NA	eng.eat.1774081.c	1774081	#NAeng.eat.	1774088	0
7	cdm-entry-id	#NA	eng.eat.1	1774081	#NAeng.eat.	1774089	0
8	cdm-translation-ref	fra	fra.manger.2	1774081	frafra.mang	1774090	0
9	cdm-translation-ref	fra	fra.bouffer.1	1774081	frafra.bouf	1774091	0

Figure 26 : Exemple de données dans la table d'indexation

II.3.1.2 **Implémentation**

Pour traiter des liens plus compliqués, nous avons enrichi l'ensemble des balises CDM par une description plus riche des liens (voir la Figure 27). Cet ensemble enrichi est nommé CDM-LINKS. Pour chaque lien, plusieurs informations peuvent être indexées :

- l'identifiant de l'entrée source.
- l'identifiant de l'entrée cible.

- l'identifiant de l'élément XML de l'entrée source contenant le lien. Par exemple, le numéro de sens dans le cas d'une entrée polysémique avec un lien de traduction pour chaque sens. Cela permet de retrouver précisément l'origine du lien.
- le nom du lien. Celui-ci est utilisé pour distinguer des liens de types différents dans une même entrée, par exemple un lien de traduction et un lien de synonymie.
- la langue cible (code à trois lettres ISO-639-2/T).
- le volume cible.
- le type de lien. Certains sont prédéfinis car ils sont utilisés par les algorithmes de calcul des liens riches (traduction, axème, axie), mais il est possible d'en utiliser d'autres.
- une étiquette dont le texte est libre.
- un poids dont la valeur doit être un réel dans `]-10,+10[`.

```
<cdm-elements>
  <cdm-volume    xpath="/p:volume"/>
  <cdm-entry    xpath="/p:volume/p:vocable"/>
  <cdm-entry-id    xpath="/p:volume/p:vocable/@p:id"/>
  <cdm-headword    xpath="/p:volume/p:vocable/p:lemma/text()"/>
  <cdm-headword-variant    xpath="/p:volume/p:vocable/p:altspelling/text()"    d:lang="eng"/>
  <cdm-pos xpath="/p:volume/p:vocable/p:pos/text()" />
  <cdm-sense-id    xpath="/p:volume/p:vocable/p:lexie/@p:id"/>
  <links>
     <link name="axeme" xpath="/p:volume/p:vocable/p:lexie/p:entryref">
          <type xpath="@type" />
          <volume xpath="@volume" />
          <value xpath="@p:idref" />
          <lang xpath="@lang" />
          <label xpath="@p:relation-mono" />
     </link>
  </links>
</cdm-elements>
```

Figure 27 : Exemple d'utilisation de CDM-LINKS

Ces liens peuvent être établis entre deux entrées d'un même volume ou entre deux volumes différents. Un même volume peut regrouper des entrées reliées à plusieurs volumes. Pour réaliser l'implémentation des liens riches, nous avons séparé la table des liens de la table de CDM-CLASSIQUE. Voir la Figure 28 (nouvelle table en rouge pour stocker les liens riches).

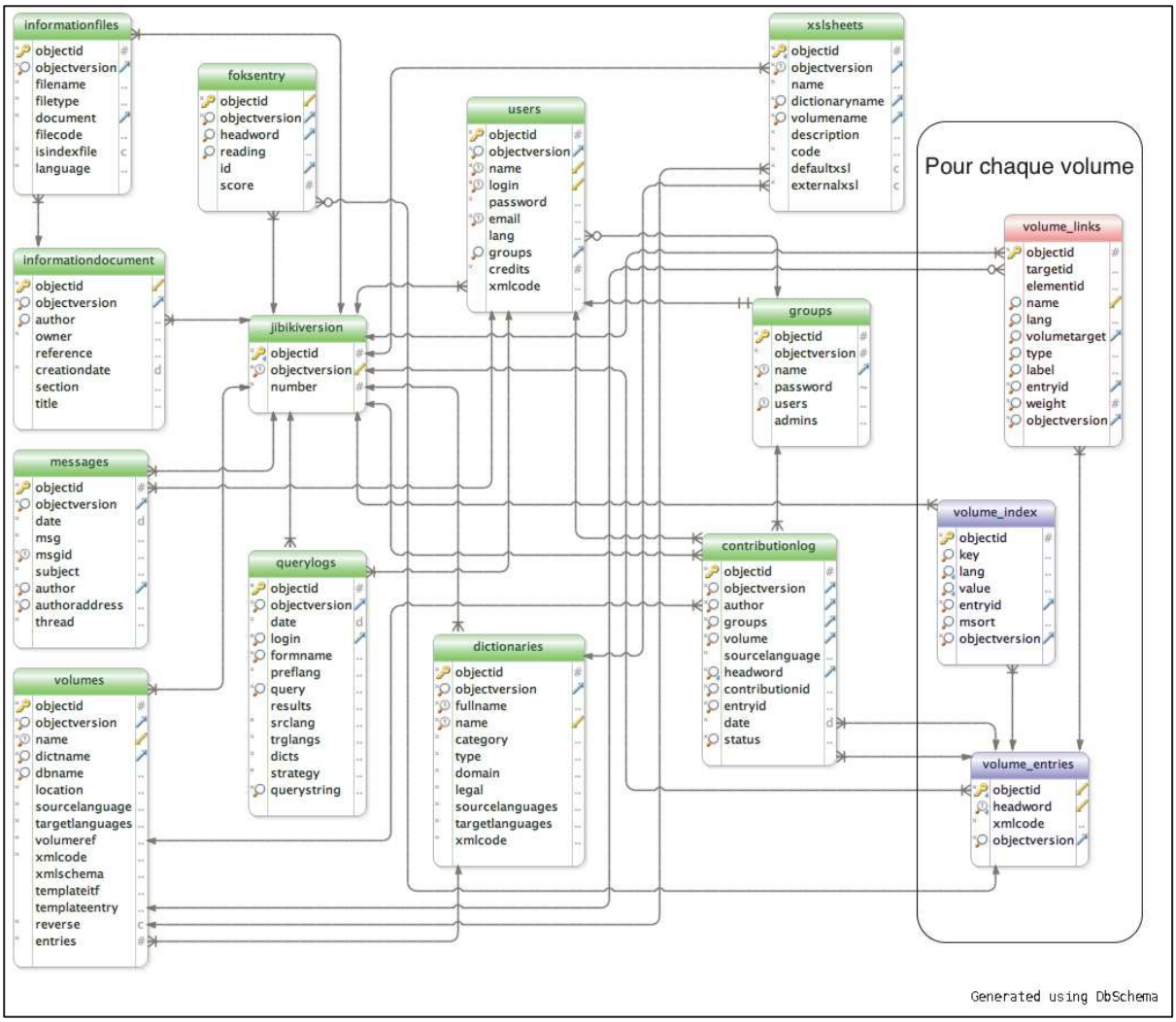


Figure 28 : Schéma de la base de données de JIBIKI-2

II.3.2 Listes non bornées

Dans JIBIKI-1, on n'a que deux types de recherche : recherche générique et recherche avancée. On en a ajouté un troisième dans cette nouvelle version, la recherche par volume. Cette fonction simule l'utilisation d'un dictionnaire papier. Si un mot est entré, les mots-vedette voisins dans l'ordre alphabétique sont affichés dans la partie gauche de la fenêtre. Cela permet de proposer les mots les plus proches pour une entrée incorrecte ou incomplète.

Cette partie est équipée d'un "ascenseur infini". Il est possible de naviguer dans tout le volume selon l'ordre alphabétique. Il est possible également de visualiser le contenu d'un article dont le mot-vedette est affiché dans la partie gauche en cliquant dessus. L'article s'affichera alors dans la partie droite.

Cette nouvelle fonction a été réalisée par M. Mangeot dans JIBIKI-2. Comme PIVAX-2 (voir II.4) a été développé sur la base de JIBIKI-2, il en a hérité naturellement.

Le temps de réponse et la qualité de cette fonction sont satisfaisants. Voici quelques expériences.

- Sur le serveur MOTÀMOT/JIBIKI-2, la recherche d'une entrée et l'affichage d'un mot du volume français (13K entrées) prennent moins d'une seconde.

- Sur le serveur PAPILLON/JIBIKI-2, la recherche d'une entrée et l'affichage d'un mot du volume FEM_FRA (19K entrées) prennent moins d'une seconde.
- Sur le serveur PIVAX-2/JIBIKI-2, la recherche d'une entrée et l'affichage d'un mot du volume UNL_UWPEDIA-FRA-LEXIE (772K entrées) prennent environ 4 secondes.
- Visualiser un article au début, au milieu ou à la fin de la liste est instantané.
- Pour aller d'un article à un autre à 10000 articles de distance, la manipulation de l'ascenseur est aussi fluide que celle du Finder du Mac dans un dossier à 50000 fichiers.

Figure 29 : Interface de liste non bornée

II.3.3 Possibilité générique de recherche avec lemmatisation

La recherche avec lemmatisation est implémentée dans l'interface de "recherche avancée" par l'ajout d'une condition `"le lemmatiseur est xxx"` pour un mot-vedette donné.

Cette fonction a été réalisée deux ans plus tard, au moment du travail sur LEXTOH (l'intergiciel de lemmatisation, voir IV.2). JIBIKI-2 envoie une requête à l'API de LEXTOH pour récupérer les résultats de lemmatisation[1], puis recherche les lemmes dans sa BDLex. On peut utiliser tous les lemmatiseurs implémentés dans LEXTOH. Les noms des lemmatiseurs sont proposés librement dans un formulaire. L'interface d'affichage est en deux colonnes, comme celui de la liste non bornée. La colonne de gauche affiche tous les résultats trouvés qui correspondent aux conditions. La colonne de droite affiche le contenu détaillé si on clique dessus.

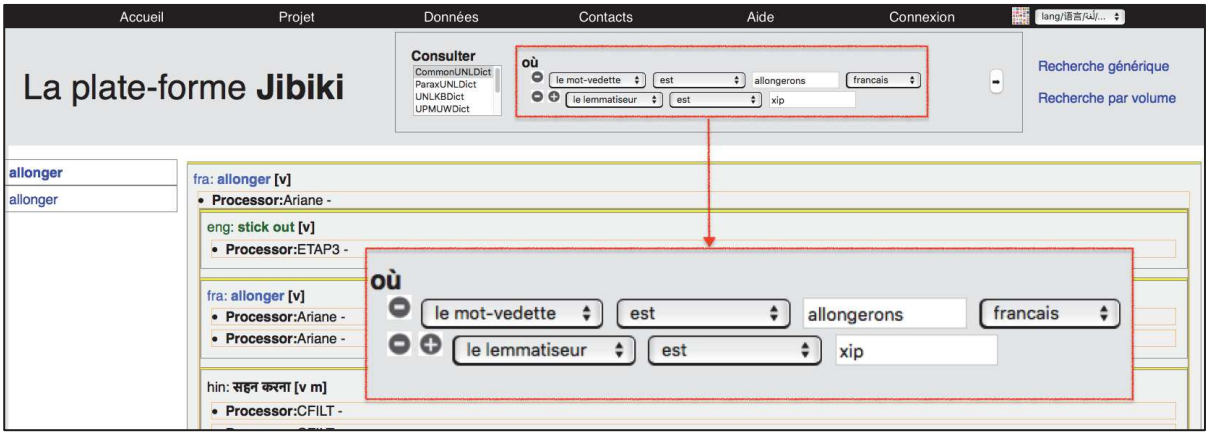


Figure 30 : Interface de recherche avec lemmatisation

[1] Au niveau de l'implémentation, on a utilisé le jar JSOUP. Cet outil est en source ouvert. Il permet de récupérer le résultat de LEXTOH par une seule commande : `Document doc = Jsoup.connect("URL d'API Lextoh").get();`.

II.4 PIVAX-2 : opérationnalisation et extension de PIVAX-1

PIVAX-2 est la réimplémentation de PIVAX-1 sur JIBIKI-2. PIVAX-2 hérite donc de toutes les fonctions et caractéristiques de JIBIKI-2. Dans cette section, on montrera les améliorations obtenues dans PIVAX-2 : (1) algorithme générique de calcul des liens, (2) passage à l'échelle et accélération, (3) support au projet GBDLex-UW++ et mise à disposition de ressources.

II.4.1 Algorithme générique de calcul des liens[1]

L'algorithme de calcul des liens programmé par H.-T. Nguyen pour PIVAX-1 est opérationnel (voir II.2.2.3), mais, à cause de l'impossibilité de spécifier un volume cible lorsqu'on décrit un lien, son algorithme est spécifique à la macrostructure PIVAX-1 et ne peut pas être utilisé pour un autre type de macrostructure[2]. De plus, H.-T. Nguyen a dû effectuer des développements spécifiques en modifiant le code JAVA de la plate-forme JIBIKI. Son code n'était donc pas réutilisable.

Avec JIBIKI-2/PIVAX-2, nous avons souhaité traiter de manière générique les liens entre tous les types de volume, de façon à éviter de modifier le code JAVA de la plate-forme pour traiter une macrostructure particulière. C'est pourquoi nous avons ajouté des liens supplémentaires.

- Dans les volumes de lexies, nous avons des liens vers les axèmes, de type `axeme`.
- Dans les volumes d'axèmes, nous avons des liens vers les axies, de type `axie` et vers les lexies, de type `final`.
- Dans le volume d'axies, nous avons des liens vers les axèmes, de type `axeme`.

Cette extension concerne non seulement de la macrostructure PIVAX, mais aussi d'autres macrostructures comme PIVOT et DIRECT.

Notre algorithme est en trois parties.

1. Recherche dans la base des entrées satisfaisant une liste de critères.
2. Recherche récursive des entrées reliées aux entrées précédemment sélectionnées.
3. Affichage des entrées avec les entrées reliées.

Pour la recherche récursive et l'affichage des entrées reliées, il faut savoir où s'arrêter, car on peut facilement imaginer une ressource remplie de liens entre entrées et si la recherche est récursive, elle ne s'arrête jamais. On le fait par la condition de détermination.

```
if (cond1||cond2||cond3){continuer la recherche récursive} else {s'arrêter}
```

Voici par exemple ce que nous avons fait pour deux types de recherche, de manière générique.

1. Cas de liens allant d'un point à un autre dans le graphe des liens.
   Par exemple :

   `entrée A (Volume A) → entrée B (volumeB) → entrée C (volume C)`

   Le lien `A→B` est marqué comme `indirect` ou `null`. Le dernier lien `B→C` est marqué comme étant un lien `final`, donc la recherche s'arrête[3].
2. Cas de liens de type `axie` qui déterminent un arbre dans une partie du graphe, l'axie étant la racine de l'arbre.
   Par exemple :

[1] La rédaction de cette section a été faite en collaboration avec M. Mangeot.

[2] Par exemple, le type prolexème et le type proaxie. Nous présenterons ces deux types dans le chapitre 3.

[3] En effet, il est tout-à-fait possible d'avoir les liens `B←C` dans le volume C. Mais on n'ira pas chercher plus loin.

`entrée A (Volume A)` → `entrée B (volume B)` → `entrée Axie (volume Axie)` → `entrée B' (volume B')` → `entrée A' (volume A')`

Le lien `B→Axie` est marqué comme étant un lien `axie`, donc ensuite on redescend dans l'arbre des liens. Le lien `B'→A'` est marqué comme étant `final`, donc la recherche s'arrête.

Ces types de recherche s'effectuent de manière générique. Il n'y a donc pas de code spécifique à chaque macrostructure. Quand une nouvelle macrostructure se présente, l'idée est de voir si l'on ne peut pas utiliser les algorithmes génériques et sinon, de voir quels sont les développements minimaux à faire pour permettre une recherche dans cette macrostrucutre. Par exemple, on peut ajouter un nouveau type de volume dans la condition de détermination.

II.4.2 Passage à l'échelle et accélération

II.4.2.1 Données lexicales supportées par PIVAX-2

Dans PIVAX-2, on a réimplémenté toutes les ressources disponibles de PIVAX-1 grâce à l'aide de H.-T. Nguyen (voir II.2.3.2). On a aussi mis dans la base de nouvelles ressources : celles du projet ITOLDU (voir I.1.3.2.4) et deux très grosses ressources (voir II.2.3.2) : une produite par V. Dikonov (COMMONUNLDICT) et l'autre par D. Rouquet (UWPEDIA).

Table 4 : Nombre des entrées COMMONUNLDICT

Volume	*Langue*	*Entrées*
CommonUNLDict_axi	axi	82 804
CommonUNLDict_eng	anglais	45 471
CommonUNLDict_eng-axemes	anglais	82 069
CommonUNLDict_esp	espagnol	7 080
CommonUNLDict_esp-axemes	espagnol	22 254
CommonUNLDict_fra	français	27 537
CommonUNLDict_fra-axemes	français	48 312
CommonUNLDict_hin	hindi	31 255
CommonUNLDict_hin-axemes	hindi	50 380
CommonUNLDict_msa	malais	37 342
CommonUNLDict_msa-axemes	malais	31 699
CommonUNLDict_rus	russe	28 475
CommonUNLDict_rus-axemes	russe	45 020
CommonUNLDict_unl	unl	82 804
CommonUNLDict_unl-axemes	unl	82 804
CommonUNLDict_vie	vietnamien	6 585
CommonUNLDict_vie-axemes	vietnamien	8 819

Table 5 : Nombre des entrées UWPEDIA

Volume	*Langue*	*Entrées*
UNL-UWpedia-ESP-lexie	espagnol	559 488
UNL-UWpedia-ESP-axeme	espagnol	559 488
UNL-UWpedia-FRA-lexie	français	772 068
UNL-UWpedia-FRA-axeme	français	772 068
UNL-UWpedia-RUS-lexie	russe	451 336
UNL-UWpedia-RUS-axeme	russe	451 336
UNL-UWpedia-UNL-lexie	unl	1 823 943
UNL-UWpedia-UNL-axeme	unl	1 823 943
UNL-UWpedia-axie	axi	1 831 219

II.4.2.2 **Évaluations comparatives des temps de réponse**

a. *Environnement*

Comme PIVAX-1 n'était plus facilement accessible au début de ce travail, nous l'avons réinstallé sur la même machine que PIVAX-2 et importé plusieurs ressources nécessaires.

Nous avons testé les deux sur les mêmes volumes de données (donc les mêmes tailles de BD). Les deux machines (client et serveur) sont implémentées dans un même intranet du LIG. Pour les environnements des tests, voir la table ci-dessous.

Table 6 : Environnement des tests de PIVAX-1 et PIVAX-2

Adr. serveur	Adr. Client	Date/Heure	Config. serveur	Config. client	Description Script
PIVAX-2 129.88.46.100:8998 PIVAX-1 129.88.46.100:8995	129.88.67.160	9/04/2015 10h-17h et 10/04/2015 10h-17h	Debian 6.0.10 Processeur 3.3 GHz Intel Core i3 Mémoire 8 GB	Mac OS X 10.9.5 Processeur 2.2 GHz Intel Core i7 Mémoire 4 GB 1333MHz DDR3	Script en JAVA. Le temps d'exécution comprend : lire le fichier des entrées, envoyer les requêtes, récupérer les résultats, écrire le fichier log.

b. *Tests*

Pour les tests, on utilise trois fichiers pour les entrées (4 entrées, 400 entrées et 4000 entrées), qui ont été créés de façon aléatoire par un script JAVA à partir des volumes importés. On a fait 10 expériences pour chaque fichier et pour chaque version de PIVAX. Les entrées concernent 4 langues : chinois, espagnol, russe et UNL.

Les résultats sont montrés dans la Table 7. Nous avons choisi deux critères : le temps utilisé (instant de la dernière réponse reçue – instant d'envoi de la première requête) et le nombre de réponses correctes. Le temps de réponse de PIVAX-2 est environ la moitié de celui de PIVAX-1. D'autre part, point important, PIVAX-1 renvoie beaucoup d'erreurs système (`chargement impossible`).

Table 7 : Résultat des évaluations comparatives des temps de réponse

Fichier de test	Système	Temps moyen en millisecondes	Temps en min/sec	Nombre d'erreurs	Nombre de résultats corrects
entries4Test.txt	PIVAX-1	980	0 min, 0 sec	0	4
entries4Test.txt	PIVAX-2	508	0 min, 0 sec	0	4
entries400Test.txt	PIVAX-1	61 832	1 min, 1 sec	214	186
entries400Test.txt	PIVAX-2	32 724	0 min, 32 sec	2	398
entries4000Test.txt	PIVAX-1	899 889	14 min, 59 sec	2 577	1 423
entries4000Test.txt	PIVAX-2	479 763	7 min, 59 sec	7	3 993

c. *Petite analyse*

On peut analyser ces résultats de deux points de vue principaux.

(1) M. Mangeot avait développé une API REST pour JIBIKI-1 avant mon arrivée au labo. Cette API a été conservée et mise à jour dans JIBIKI-2. Mais PIVAX-1 n'utilisait pas cette API. C'est pourquoi, pour les tests sur PIVAX-1, nous devions charger une page entière, y compris les

images (dont les logos), puis récupérer le résultat dans un fichier HTML. L'API de PIVAX-2 nous permet de récupérer directement les résultats en XML sans charger des informations inutiles.

(2) Comme on l'a expliqué au II.3.1, les informations portées par les liens de PIVAX-1 sont stockées dans la table d'indexation avec toutes les autres informations de CDM. Mais, dans PIVAX-2, les informations portées par les liens sont stockées séparément dans la table de "links". Son nombre d'enregistrements est bien inférieur à celui de la table d'indexation de PIVAX-1. Par exemple, pour le volume COMMONUNLDICT_ENG, le nombre d'enregistrements de la table d'indexation est 272 826, et le nombre d'enregistrements de la table de `links` est 82 096. Donc, pour une macrostructure complexe comme celle de PIVAX, la recherche par liens dans la BD est effectivement améliorée.

II.4.3 Support au projet GBDLEX-UW++ et mise à disposition de ressources

II.4.3.1 Support au projet GBDLEX-UW++

On a déjà mentionné ce projet et ses ressources lexicales plus haut (voir I.1.3.2.2, II.2.3.2 et II.4.2). On a beaucoup travaillé sur le projet GBDLEX-UW++ dans le cadre du projet TRAOUIERO, et on a importé toutes ses ressources disponibles dans PIVAX-2.

Grâce aux retours de V. Dikonov (linguiste et lexicographe de l'IPPI), qui a le plus travaillé sur ses données sous PIVAX-2, nous avons détecté de nombreux problèmes. Par exemple, on a cherché le mot-vedette anglais *milk* en UNL et en anglais, et on a trouvé tous les UW et les traductions des LN, y compris les mots russes. Puis on a cherché les mots-vedette *traire* et *lait* en français, ça marchait aussi. Cependant, quand on a cherché les mots-vedette russes (доить = *to milk* et молоко = *milk*), on n'a trouvé aucune traduction associée.

Ce problème provient de la pauvreté des données, et aussi du manque de progammabilité de PIVAX-2. C'est parce que les liens de recherche sont orientés, voir la Figure 31. En effet, il manque des liens des lexies russes vers les axèmes russes (dans le volume des lexies russes) et/ou des liens des axèmes russes vers les axies interlingues (dans le volume des axèmes russes).

V. Dikonov avait oublié la création de ces liens quand il avait construit la ressource, et ensuite il n'a pas voulu (ou pas pu) la modifier. Il faut dire que c'est impossible à faire à la main pour une grosse taille et que c'est toujours difficile à programmer par un linguiste.

Cela nous montre l'importance de la programmabilité (par langage narratif ou par manipulation graphique directe) de notre système.

C'est ce que H.-T. Nguyen a spécifié dans le dernier chapitre de sa thèse (voir II.2.2.5).

Il serait vraiment intéressant d'implémenter une telle fonctionnalité programmable, mais cela nous aurait pris plusieurs mois (à temps partiel, en parallèle avec le travail chez L&M).

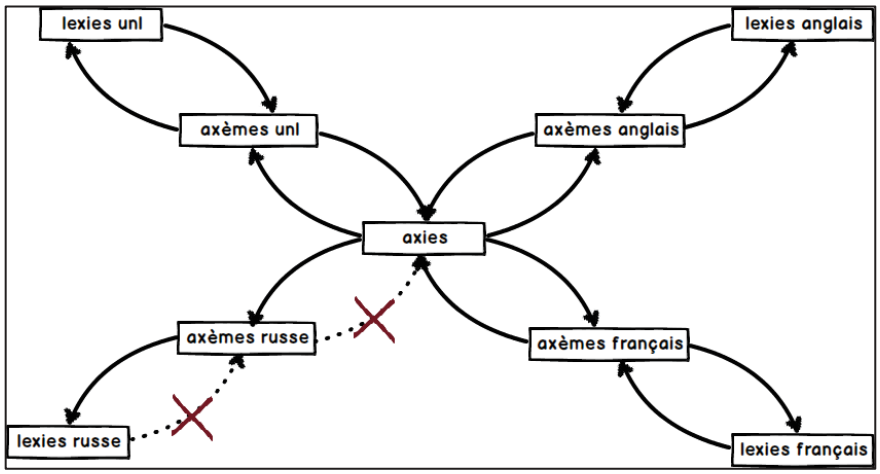


Figure 31 : Problèmes de la ressource de V. Dikonov

II.4.3.2 **Mise à disposition de l'outil et de données sur le Web**

Le serveur PIVAX-2 est implémenté sur un serveur intranet du GETALP (takara.imag.fr) ; on peut trouver sa configuration dans la Table 6. Le lien pour y accéder par l'intranet est http://takara.imag.fr:8998/pivax/Home.po. On peut aussi y accéder depuis Internet par http://getalp.imag.fr/pivax/Home.po grâce à une redirection.

D'une part, on a hérité de toutes les fonctions de JIBIKI-2, et d'autre part on a récupéré toutes les fonctions de PIVAX-1. Pour l'interface de recherche générique, on a implémenté deux types d'affichage : l'affichage en colonnes de PIVAX-1 (column display)[1] et l'affichage par défaut réalisé par JIBIKI-2 (row display).

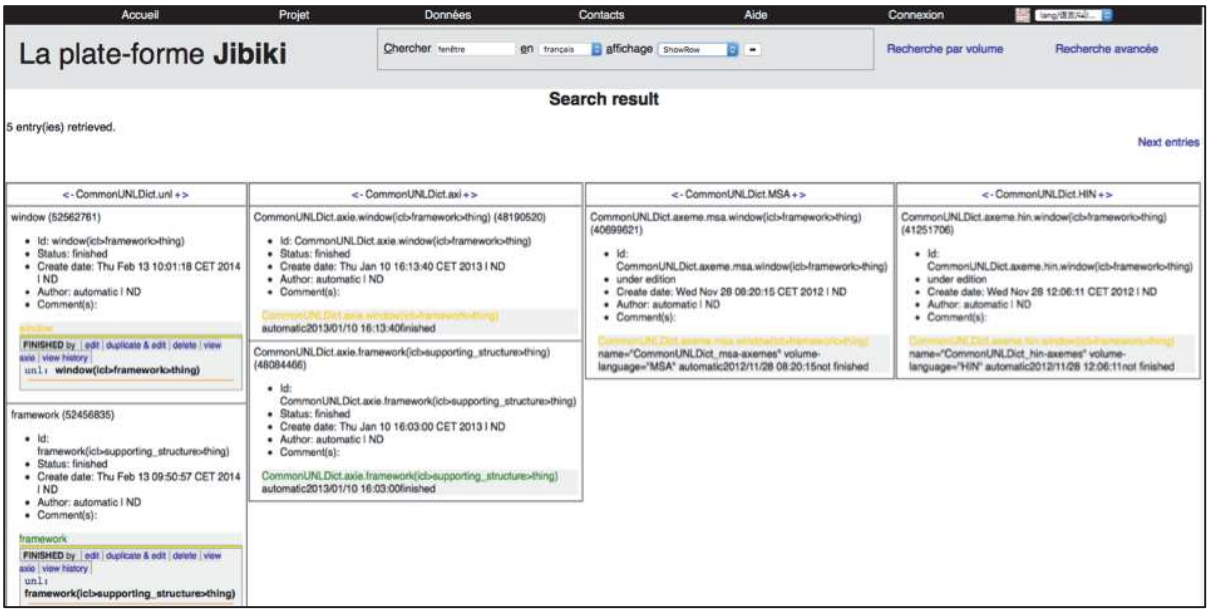


Figure 32 : Interface d'affichage en colonnes des PIVAX-2

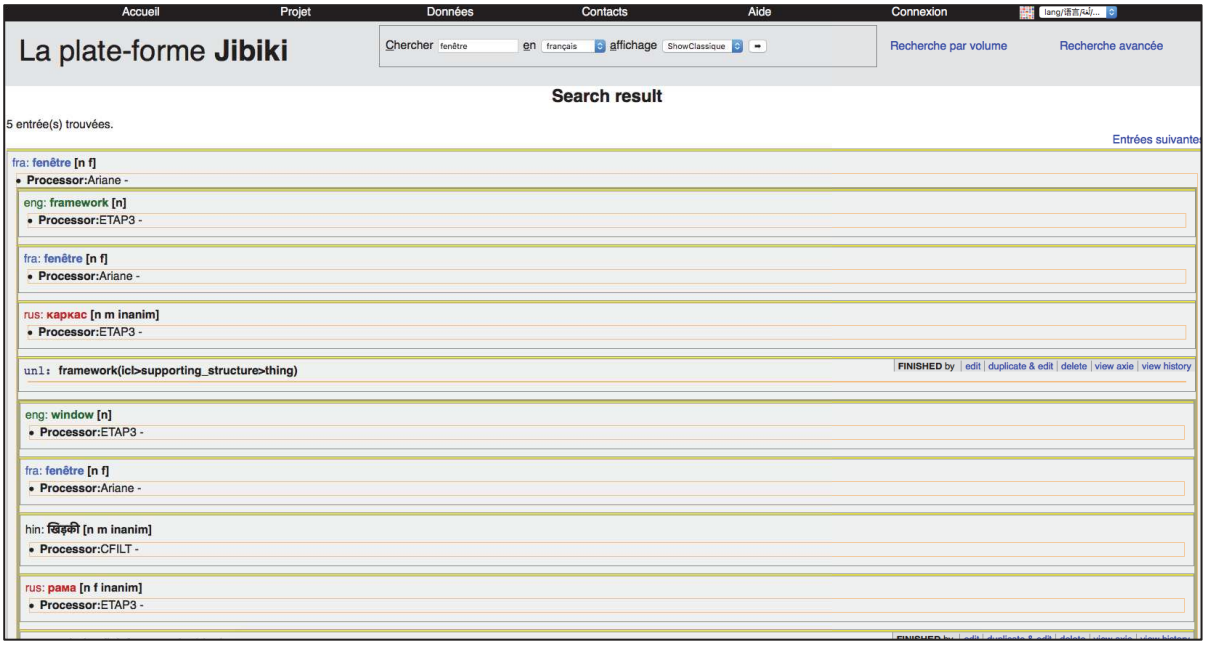


Figure 33 : Interface d'affichage classique (en lignes) de PIVAX-2

[1] Pour l'implémentation de cette interface en colonnes, voir l'Annexe 6.

Chapitre III Une nouvelle architecture intégrant les données lexicales générales, terminologiques et "situées" : PIVAX-3

Introduction

Ce chapitre présente une contribution originale à la lexicographie computationnelle, qui a donné lieu à une publication [Zhang, Y. & Mangeot, M., 2013] à LTT en 2013. Le point de départ a été un besoin précis de L&M, à savoir la gestion d'un certain type d'abréviations, les acronymes, pour certains clients.

Les acronymes en question sont des unités lexicales souvent terminologiques, mais pas toujours. Un même "prolexème" (collection d'unités lexicales synonymes dans une certaine situation, spatio-temporelle et/ou technique et/ou entrepreneuriale) n'a pas nécessairement de réalisation de type acronyme (ou abréviation, ou apocope, ou mot-valise) dans différentes langues. Du point de vue informatique, il faut pouvoir traiter ces unités dans la même base de données que les termes techniques et que les termes généraux.

Nous avons proposé un modèle pour réaliser l'intégration de ces 3 types d'unités lexicales dans une même BDLex. Pour des raisons informatiques, L&M ne pouvait pas intégrer cette solution à sa BDLex ; la solution réalisée pour L&M n'a donc été que partielle. Par contre, au laboratoire, nous avons pu l'implémenter complètement en PIVAX-3/JIBIKI-2, et produire un démonstrateur.

Dans la première section, nous analysons le problème posé par L&M et sa demande précise. Dans la deuxième section, nous étudions les éléments de la solution. Ensuite, dans la troisième section, nous présentons les solutions (la solution pour L&M et la solution générale), et les démonstrations. Enfin, nous discutons les autres extensions envisageables.

III.1 Analyse d'un problème posé par L&M

III.1.1 Présentation du problème rencontré par L&M

III.1.1.1 Contexte

On a déjà mentionné les besoins réels de L&M au I.2.2.2.2. Vers février 2012, L&M a été confrontée au problème suivant : deux clients (EDF et Wesco) avaient à traduire des textes comportant beaucoup d'acronymes. Ces acronymes figuraient dans leurs fichiers terminologiques, qui avaient été importés dans LIBELLEX, mais les traducteurs n'obtenaient pas le développement de ces acronymes, quand la traduction "acronymique" était absente. Ensuite, en 2013, deux autres clients (ExaleadSuggest et Louis Vuitton) ont demandé de traiter leurs terminologies monolingues avec des relations plus complexes.

L'outil de gestion de terminologie de LIBELLEX permet l'import, l'export, la suppression, la consultation et la validation. Il y a deux interfaces, pour la terminologie monolingue et pour la terminologie bilingue, qui diffèrent l'une de l'autre surtout pour la consultation et la validation. Pour l'import et l'export, on peut utiliser les formats d'échange de METRICC (TBXMETRICC et TEIMETRICC, voir I.2.2.1 et I.2.2.2). Mais ces formats sont dédiés au traitement automatique et sont difficiles à comprendre et à utiliser par les clients (par les humains en général).

LIBELLEX propose un nouveau format d'échange, TSV (Tabulation-Separated Values)[1], et nous avons principalement utilisé ce format TSV pour les tâches liées à LIBELLEX (voir les exemples dans la section suivante).

Juste au début de ma thèse, vers mi-avril 2012, LIBELLEX a réalisé une interface graphique pour la terminologie monolingue et pour afficher les réseaux sémantiques. Cette fonction a été initialement réalisée par Mikaël Morardo. Au début, les liens sont créés entre deux termes seulement s'ils partagent des contextes similaires. Une matrice nœuds/contextes est construite dynamiquement en utilisant l'analyseur syntaxique FRMG [Villemonte de la Clergerie, É. et al., 2009]. M. Morardo a présenté sa méthode et son algorithme dans [Morardo, M. & Villemonte de La Clergerie, É., 2013].

Figure 34 : Interface de consultation graphique pour la terminologie monolingue dans LIBELLEX

III.1.1.2 Extraits de la "ressource" des clients

Début 2012, les ressources importées en format TSV contenaient quatre types de fichier : (1) monolingue sans phrases d'exemples en une seule colonne : langue en ISO 639-1 (ex. `fr`), (2) monolingue avec phrases d'exemple en deux colonnes (ex. `fr[Tab]fr_sent`), (3) bilingue sans phrases d'exemples en deux colonnes (ex. `fr[Tab]en`) et (4) bilingue avec phrases d'exemples en quatre colonnes (ex. `fr[Tab]en[Tab]fr_sent[Tab]en_sent`).

	A	B	C	D
1	fr	en	fr_sent	en_sent
2	Résultat de base par action	Basic earnings per share	Résultat de base par action (en euros)	Basic earnings per share (in EUR)
3	Comité s' est réuni deux fois	Committee met	Ce Comité s'est réuni deux fois en 2005.	The Committee met twice in 2005.
4	marché a cru de 6,3%	period , the market increased	Entre 2004 et 2005, le marché a cru de 6,3%.	In the 2004-2005 period, the market increased 6.3%.
5	stratégie	strategy	stratégie & développement	strategy & development
6	Euromaster	Euromaster	et Sociétés de distribution Euromaster	and Euromaster distribution Companies

Figure 35 : Ressource bilingue importée (avec phrases) dans la BDLex de LIBELLEX

Voici quelques exemples de ressources transmises par les clients après mon arrivée.

[1] TSV est un format texte représentant des données tabulaires sous forme de "valeurs séparées par des tabulations". Chaque ligne correspond à une rangée du tableau et les cellules d'une même rangée sont séparées par une tabulation. — Wikipédia

A	B	C	D
Acro_eng	Def_eng	Acro_fra	Def_fra
EFAD	European Federation of the Association of Dietitians	FEAD	Fédération Européenne des Associations de Diététiciens
SNCF	Société Nationale des Chemins de Fer	SNCF	Société Nationale des Chemins de Fer
EERI	Estonian Energy Research Institute	IERE	Institut Estonien de Recherche sur l'Énergie
PWA	PieceWise-Affine	PWA	affines par morceaux
ANR	National French Research Agency	ANR	Agence Nationale de la Recherche
CHOC	CHallenge in Combinatorial Optimization	CHOC	CHallenge in Combinatorial Optimization
CIGC	Computing and Computational Grids	CIGC	Computing and Computational Grids
OLSR	shortest path base flat routing protocol	OLSR	protocole de routage du plus court chemin
LLC	linear logic concurrent	LCC	contraintes en logique linéaire
MRF	Markov Random Field	MRF	Markov Random Field
RRAM	Robust Rate Adaptive Multicast mechanism	RRAM	Robust Rate Adaptive Multicast mechanism
LB-ARF	leader-based mechanism	LB-ARF	leader-based mechanism
MCMC	Monte Carlo Markov Chains	MCCM	Monte Carlo par Chaîne de Markov
HOACs	Higher-order active contours	CAOS	contours actifs d'ordre supérieur

Figure 36 : Exemple de ressource prétraitée dédiée à des acronymes[1]

	A	B	C	D	E	F	G	H	I
1	Traçabilité source	entité nommée	parent français	français	anglais GB	Statut invalidé =1 validé défaut =2	synonyme français	synonyme anglais GB	commentaires
2	job02	nom de produit		Murakami	Murakami	1			Pas un nom de produit
3	job02	nom de ligne de produits		Polka Dots Plato	Polka Dots Plato	1			Abandonner/ utiliser à la place Polka Dots
4	job02 termino	vocabulaire		souliers	shoe	1			Dans le contexte commercial on utilise en général le pluriel.
5	Produits Louis Vuitton	nom de produit	PETITE MAROQUINERIE	4 CARTES DE CREDIT ET 2 TRANSPARENTS		2			
6	job01	vocabulaire	OUTIL DE MAROQUINERIE	ABAT-QUART	EDGE BEVELER	2			
7	Produits Louis Vuitton	nom de produit	SAC DE VILLE	ABBESSES		2			
8	Produits Louis Vuitton	nom de produit	SAC DE VILLE	ABELIA		2			
9	Produits Louis Vuitton	nom de produit	SOULIERS - LIGNES	ABIDJAN		2			
10	job01	nom de couleur	ORANGE	ABRICOT	APRICOT	2			
11	Produits Louis Vuitton	nom de produit	SOULIERS - LIGNES	ABSTRACT		2			
12	job01	nom de couleur	JAUNE	ACACIA	ACACIA	2			

Figure 37 : Exemple d'une ressource de Louis Vuitton prétraitée

	A	B	C	D	E
1	fr	fr_sents	fr_variants	fr_types_variants	
516	En cas de taxation d'office				
517	Contrats d'assurance maladie/retraite à gestion paritaire				
518	Modification des conditions de publicité du privilège du Trésor				
519	CRÉDIT EN FAVEUR DES DÉBITANTS DE TABAC				
520	organismes de logement social				
521	destiner				
522	activité agricole		activités agricoles	variant	
523	savoir				
524	statut				
525	directeur général des finances publiques	du 28 déc. 2011, art. 1er) Le directeur départemental ou régional ou le délégataire du directeur général des finances publiques peut résilier la c			
526	commission départementale des impôts	Décision de la commission départementale des impôts directs et des TCA			
527	dette fiscale	Le juge ne donne pas de base légale à	dettes fiscales	variant	
528	auteur				
529	capital				
530	conduire				
531	mouvement				
532	valeurs mobilières	193 bis Organismes de placement collectif immobilier 239 nonies, 749, 825 Organismes de placement collectif de valeurs mobilières (OPCVM) E			
533	contribuables modestes				
534	ignorer				
535	Vins, cidres, etc.				
536	Fraude fiscale. Relaxe				
537	heure				

Figure 38 : Exemple d'une ressource d'ExaleadSuggest prétraitée

III.1.1.3 **Demande précise de L&M**

L&M m'a demandé d'étudier ce problème, de trouver une solution implémentable à l'intérieur de LIBELLEX, et de l'implémenter. J'ai d'abord étudié et évalué la structure de la BDLex existante pour voir si elle permettait de traiter ce type de problème ou bien si l'on avait besoin d'installer une autre BDLex séparée (dans cette perspective, nous avons proposé un nouveau type de BDLex basé sur JIBIKI).

[1] C'est à 50% non traduit, et les traductions sont à 50% très mauvaises.

Ensuite, j'ai travaillé sur l'import et l'export de terminologies complexes en TSV, contenant des acronymes et diverses relations. Enfin, avec M. Morardo, j'ai réalisé l'affichage et l'intégration dans les interfaces existantes, principalement dans l'interface graphique.

III.1.2 Analyse des problèmes posés

III.1.2.1 Défauts de la ressource lexicale du client

L&M a souvent reçu des ressources très riches mais très incomplètes. Par exemple, les ressources lexicales de Louis Vuitton contiennent principalement des entités nommées associées à des lignes de produits et à des produits. Ces ressources initiales étaient hiérarchisées en XLSX, et il fallait les mettre en format "à plat" (en TSV) tout en gardant les liens de parenté, les correspondances bilingues, les synonymes, etc.

On aurait dû avoir une traduction en anglais pour chaque terme français, mais souvent il n'y en avait pas. Par exemple, un échantillon sur lequel j'ai travaillé contenait 10 857 termes français, mais seulement 2604 traductions anglaises.

On voit aussi des erreurs dans la Figure 36 : (1) l'acronyme *SNCF* et sa définition française ont été recopiés dans les colonnes anglaises, et (2) on indique un –s pour le pluriel d'un acronyme, toujours invariable en français (les *CNAM*, pas "les *CNAMs*").

De plus, les traductions françaises sont souvent fausses ou très mauvaises.

III.1.2.2 Problèmes conceptuels

Il y a des confusions entre les acronymes et les autres types d'abréviation. Plus généralement, on voit qu'il aurait fallu qu'au moins un(e) spécialiste de terminologie participe à la construction de ces ressources. D'où la nécessité d'étudier nous-même les aspects linguistiques, et plus précisément lexicologiques, liés aux différents "objets linguistiques" à traiter.

III.1.2.3 Problèmes venant de la structure de la BDLex de LIBELLEX

On a mentionné cette structure au I.2.2.1 et au I.2.2.2. L'absence de la notion d'acception interlingue (représentée en LEXALP ou en PIVAX par une "axie") associée à un concept dans le schéma conceptuel de la base lexicale de LIBELLEX est un problème profond. Elle rend très difficile la création des relations sémantiques.

III.1.3 Étude lexicologique et lexicographique

III.1.3.1 Nécessité d'un niveau conceptuel (lexies et axies)

Les liens entre les termes sont compliqués. Plusieurs termes différents peuvent être liés à un seul référent : *Jean-Paul II* et *Karol Jozef Wojtyla* en français, ou en anglais *John Paul II* et *Karol Jozef Wojtyla*.

Des pays parlant la même langue (ex : France et Suisse romande) peuvent également utiliser des mots différents pour le même concept. Par exemple, *chien renifleur* et *chien drogue*. Inversement, le même terme peut désigner des concepts différents : dans la province de langue allemande de Bolzano en Italie, le *Landeshauptmann* est le président du conseil provincial, avec des compétences beaucoup plus limitées que le *Landeshauptmann* autrichien, qui est à la tête de l'un des États (Länder) de la fédération autrichienne. Même chose pour *principe de précaution* en français (voir la Figure 14).

C'est pourquoi on a besoin de lexies (voir la *Définition 11*) et d'axies (voir la *Définition 12*).

III.1.3.2 Nécessité de la notion de "prolexème" pour les entités nommées

Les notions d'axie et de lexie ne suffisent pas à représenter toutes les situations liées aux noms propres, parce qu'on a des dérivés, des alias, des types différents d'abréviation d'un même nom propre, etc. Par exemple, pour la ville de Saint-Martin-d'Hères, on peut trouver : *Saint Martin d'Hères*, *St Martin d'Hères*, *Saint-Martin-d'Hères*, *SMDH*, ou *SMH*.

L'association d'un prolexème (voir la *Définition 18*) aux noms propres de même référent a été proposée pour traiter ce type de problème dans la thèse de M. Tran [Tran, M., 2006]. Un prolexème permet de relier les différentes formes d'un nom propre qui apparaissent dans les différents textes d'une langue donnée.

Il s'agit non seulement de noms propres, mais aussi d'expressions métaphoriques, ou de groupes nominaux, par exemple, *Paris* et *ville lumière*, *Obama* et *président des USA*.

III.1.3.3 Différence entre le niveau terminologique et le niveau des prolexèmes

Une terminologie contient des termes situés de façon relativement permanente. Un terme (nominal ou même verbal), ou un phrasème (ex: un gène "code pour" une protéine), a un sens spécifique par rapport aux énoncés relatifs à chaque domaine ou ontologie où il apparaît.

Par contraste, une entité nommée est un "désignateur", et son référent peut changer (et change fréquemment) en fonction du temps, du lieu, du contexte socio-économique et historique, etc. Un exemple très connu est *Président des USA* : George Bush. Oui, mais quand ? Il y a eu le père et le fils, à des périodes différentes.

III.1.3.4 Nécessité de distinguer les lexèmes spécifiques d'un "sous-langage"

On peut considérer une forme comme *L8R* comme un mot spécialisé pour le sens *later*, rencontré seulement dans le contexte des textos. Il s'agit ici non pas d'une entité nommée, mais d'un vocable du sous-langage des textos en anglais.

De même, dans le sous-langage des mels ou dans celui des textos, on trouve *A+* pour *À plus* ou *À plus tard*. Dans le contexte de textos, la meilleure traduction en anglais de *A+* sera donc *L8R* et pas *later*.

III.1.3.5 Possibilité de se référer à la théorie de la cognition située

Les idées présentées plus haut (au I.3.1.1.1) sont basées sur la théorie de la cognition située. Cette théorie provient initialement des travaux de Coseriu [Coseriu, E., 1998 ; Coseriu, E., 2001]. Citons ici la définition trouvée dans WIKIPEDIA :

Situated cognition is a theory that posits that knowing is inseparable from doing [Brown, J. S. et al., 1989] by arguing that all knowledge is situated in activity bound to social, cultural and physical contexts [Greeno, J. G. & Moore, J. L., 1993].

III.2 Éléments d'une solution

III.2.1 Systèmes dont on pouvait s'inspirer

III.2.1.1 CJK.ORG

CJK.ORG a été brièvement mentionné au I.2.3.3. L'institut CJK, sous la direction de J. Halpern, s'est concentré sur le problème de l'extraction intelligente d'informations pour traiter les variantes d'écriture de plusieurs langues, à partir de 1996 [Halpern, J., 2002]. Par exemple, en chinois et en japonais, il y a plusieurs formes d'écriture, et beaucoup de variantes pour certains caractères ou mots.

Pour l'écriture du chinois, on distingue le "chinois simplifié" et le "chinois traditionnel". Entre 1956 et 1986[1], les nouvelles autorités de la Chine (RPC) ont mis en œuvre une réforme de l'écriture préparée depuis bien avant la révolution. Elle a consisté à remplacer 2 274 caractères par des formes simplifiées, provenant de formes calligraphiques. Ni Taiwan ni le Japon ni la Corée n'ont adopté ces formes. Depuis une dizaine d'années, les formes traditionnelles sont de nouveau utilisées et enseignées. La raison principale semble être qu'il est plus difficile de se souvenir du sens des formes simplifiées que des formes traditionnelles, qui sont plus structurées et se prêtent mieux à des méthodes mnémoniques.

Les gens non informés pensent qu'il s'agit juste de la conversion d'un codage des caractères vers un autre codage des caractères. En fait, c'est beaucoup plus compliqué.

Il y a quatre difficultés principales. Les deux premières concernent les conversions des caractères et des mots, la troisième la conversion du sens, et la quatrième les variantes.

(1) Il y a beaucoup de sinogrammes simplifiés qui correspondent à plusieurs sinogrammes traditionnels (et vice versa, mais moins fréquemment). Voici trois exemples.

Chinois Simplifié (CS)	Chinois Traditionnel (CT)	Remarque
头 (tóu)	頭 (tóu)[2]	Correspondance injective (1-1)
发 (fā ou fǎ, polyphone)	髮(fǎ) et 發 (fā)	Correspondance 1-n
头发 (tóu fǎ)	頭髮 (cheveux, tóu fǎ)	頭發 n'est pas un mot.

(2) Un mot écrit en chinois simplifié peut correspondre à plusieurs mots écrits en chinois traditionnel. Pour le choix, il faut voir le contexte.

CS	CT	Remarque
阴 (yīn)	陰(yīn) et 陰 (yīn)	Correspondance 1-n
干 (gān ou gàn)	乾 (gān ou qián) et 干(gān ou gàn)	Correspondance 1-n
阴干(yīn gān)	陰乾 (sécher à l'ombre, yīn gān) 陰干 (terme de médecine chinoise, yīn gān)	Voir le contexte

(3) Pour certains sens, CS et CT utilisent des mots complètement différents. Voici un exemple.

CS	CT Taiwan	CT Hong Kong	Remarque
出租车 (chū zū chē)	計程車 (jì chéng chē)	的士 (dī shì)	La conversion de "caractère à caractère" : (CS) 出租车 → (CT) 出租車, produite par Google Translate, est fausse.

[1] La première version du "Chinese Character Simplification Scheme" a été publiée le 31 janvier 1956, et concernait environ 510 caractères. La deuxième version a été publiée en mars 1964 avec comme titre "Simplified Chinese characters list". En 1986 a été publiée la troisième version (c'est la version actuelle), qui contient 3 tables, respectivement de 350 caractères, de 132 caractères et de 1 753 caractères.

[2] Dans cette thèse, nous indiquons toujours la prononciation en mandarin standard.

(4) Il y a beaucoup de variantes en CT. Par exemple, 群 et 羣, 秋 et 穐, 匯 et 滙, 啟 et 啓, etc. D'autre part, la Chine continentale utilise un troisième système, le "chinois simplifié traditionnel" (CST) pour publier des journaux, des livres etc. pour les gens qui utilisent CT, par exemple "人民日報海外版" (Rén mín rì bào hǎi wài bǎn, People's daily overseas edition). Les caractères de CST sont définis dans la norme GB/T 12345-90. Ce ne sont pas tout à fait les mêmes que ceux de CT. Voir la table ci-contre.

CS	CST	CT
线	綫	線
绷	綳	繃

Le japonais est encore plus compliqué que le chinois. Il y a quatre jeux de caractères : kanji, hiragana, katakana et romaji. Ils sont le plus souvent mélangés. Par exemple, la phrase "金の卵を産む鶏" (Kin no tamago wo umu niwatori, poulet qui pond des œufs d'or) peut avoir 24 variantes d'écriture. En plus, il existe beaucoup de variantes, par exemple, (variante de Kanji) 發 et 発, ou (homophones) 柔かい (Yawaraka ~i) et 軟かい (Yawaraka ~i). Pour plus de détails, voir [Halpern, J., 2002 ; Halpern, J., 2006].

CJK.ORG utilise des tables de correspondance pour convertir entre les différents niveaux.

Conversion entre chinois simplifié et chinois traditionnel

- Tables "Code-level mapping" pour la conversion caractère à caractère.
- Tables "orthographic et lexemic mapping" pour la conversion mot à mot.
- Tables "orthographic mapping tables for proper nouns" pour les noms propres.
- Tables "orthographic/lexemic mapping tables for technical terminology" (surtout pour l'informatique).

Normalisation orthographique du chinois traditionnel vers le chinois simplifié

- Tables de normalisation de CT en CS.
- Tables de normalisation de CST en CS.

Base de données des variantes orthographiques en japonais

- Base de données complète des variantes orthographiques en japonais.
- Base de données des groupes homophones sémantiquement classés.
- Groupes de synonymes sémantiquement classés, pour l'expansion de ces synonymes (thésaurus japonais).
- Lexique anglais-japonais pour le CLIR (cross-language information retrieval, ou RI translingue).
- Règles d'identification des variantes non listées.

III.2.1.2 IATE

IATE (Inter-Active Terminology for Europe) [Ball, S., 2003] est la base de données terminologique que partagent les institutions de l'Union européenne. Elle concerne les 25 langues officielles de l'UE. L'interface actuelle permet de choisir parmi 21 grands domaines, eux-mêmes divisés en plus de 100 petits domaines. Il y a aujourd'hui environ 8,6 millions de termes dans la base d'IATE, répartis dans approximativement 1,4 million de fiches.

La base de données est organisée à trois niveaux : concept, langue et terme. Pour ajouter une nouvelle entrée, il faut l'associer à chaque niveau en utilisant une interface avancée de manipulation des données. Cette fonction est réservée aux terminologues et aux administrateurs. Le système permet également aux terminologues d'évaluer les termes par degré de fiabilité.

Il y a parfois des doublons pour un seul et même concept. C'est parce que plusieurs ressources terminologiques (EURODICAUTOM, TIS, EUTERPE, EUROTERMS, CDCTERM) ont été fusionnées dans la base de données IATE en 2004. Chaque institution avait auparavant sa propre base de données terminologiques.

Le système fournit aux terminologues des outils de "dédoublonnage", qui permettent la sélection, la suppression ou la concaténation des données à chacun de ces trois niveaux. Ce travail est toujours en cours.

III.2.1.3 EDR

EDR ELECTRONIC DICTIONARY[1] est un dictionnaire japonais-anglais, développé entre 1987 et 1993 par le projet EDR, organisé par le MITI (Ministry of International Trade and Industry) du Japon, auquel ont participé 8 grosses entreprises[2]. La base lexicale d'EDR est composée de dictionnaires de quatre types[3] et de deux corpus [Takebayashi, Y., 1993].

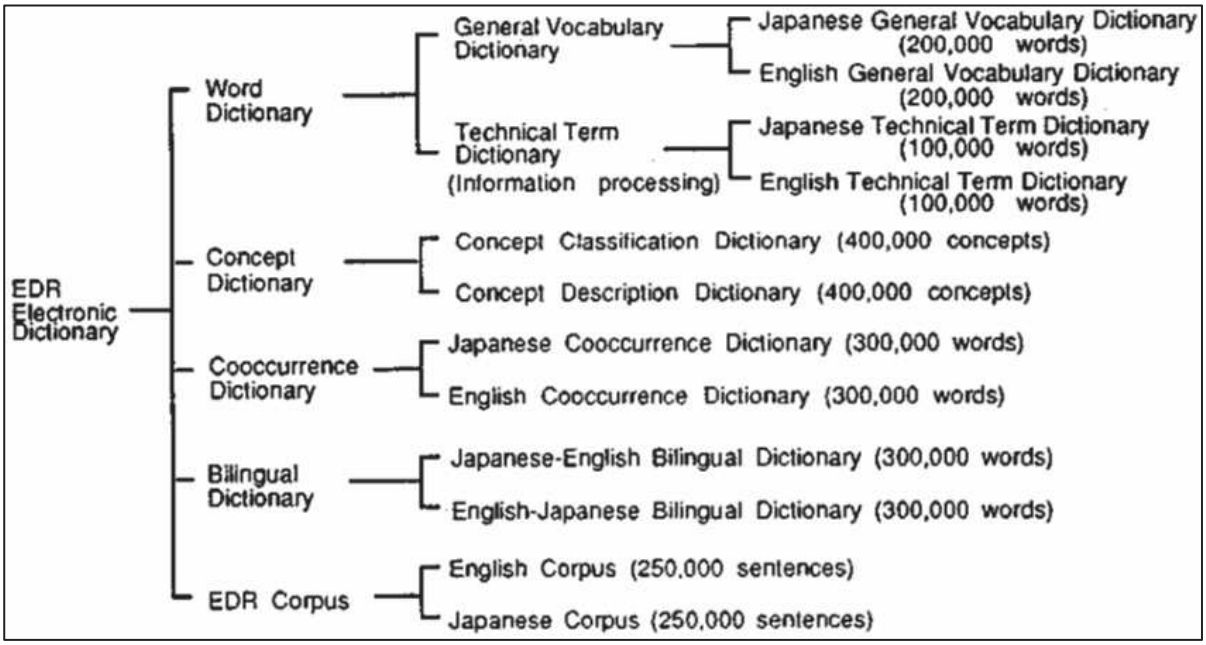


Figure 39 : Structure de EDR ELECTRONIC DICTIONARY

Les dictionnaires monolingues (Word Dictionaries) contiennent des informations grammaticales, des informations supplémentaires (l'usage, la fréquence, etc.) et des liens vers des concepts (dans le dictionnaire des concepts).

Les entrées du dictionnaire des concepts contiennent leurs définitions, des explications, ainsi que les relations entre deux concepts (dans le dictionnaire "Concept Classification"), par exemple `kind-of(concept1, concept2)`.

Les dictionnaires bilingues sont similaires aux dictionnaires papier. Ils définissent des correspondances de traduction.

Les dictionnaires de cooccurrences donnent des informations sur les usages, surtout les relations syntaxiques entre termes, par exemple, `eaten @d-object lunch`.

---

[1] https://www2.nict.go.jp/out-promotion/techtransfer/EDR/index.html?

[2] Fujitsu, Ltd., NEC Corporation, Hitachi, Ltd., Sharp Corporation, Toshiba Corporation, Oki Electric Industry Co., Ltd., Mitsubishi Electric Corporation, and Matsushita Electric Industrial Co., Ltd.

[3] Ou 5 types, si on ne fusionne pas le type monolingue général et le type monolingue terminologique dans un seul type.

III.2.2 PROLEXBASE et les prolexèmes

III.2.2.1 Le projet PROLEX

a. *Contexte*

Le projet PROLEX[1] a été lancé par le Laboratoire d'Informatique (LI) de l'université François-Rabelais de Tours en 1994. Son objectif était le traitement automatique des noms propres et la création d'un dictionnaire relationnel de noms propres.

PROLEX a produit PROLEXBASE, un système développé par Mickaël Tran dans le cadre de sa thèse [Tran, M., 2006]. PROLEXBASE est un dictionnaire électronique relationnel multilingue pour les noms propres.

On retranscrit ci-dessous les points importants et les notions de base introduites dans [Tran, M., 2006].

b. *Définition d'un nom propre*

M. Tran a listé plusieurs définitions différentes pour les noms propres, et il a finalement adopté la définition de [Jonasson, K., 1994].

Définition 22. **Nom propre** [Jonasson, K., 1994] : Toute expression associée dans la mémoire à long terme à une entité particulière en vertu d'un lien dénominatif conventionnel stable.

c. *Critères des noms propres*

Dans son analyse de la complexité du problème de l'identification des noms propres, il a défini 5 critères.

Critère de la majuscule à l'initiale. Cela dépend des langues et des cas. Pour le français, le critère de la majuscule s'applique seulement à l'écrit, mais ne concerne pas l'oral. L'emploi de la majuscule n'est pas limité aux noms propres, mais aussi à certains noms communs quand ils sont utilisés de façon "personnifiante" (ex : *la Mort*, *la Nature*). Dans le cas des mots composés, la majuscule n'apparaît pas toujours pour chaque élément (ex : *la tour Eiffel*, mais *Le Mans*).

Critères morphologiques. En français, les noms propres sont souvent invariables en genre et en nombre, mais il y a des exceptions et des incertitudes (ex : *les îles Spratleys* et *les îles Spratley*).

Critères syntaxiques. Les noms propres peuvent être ou non accompagnés d'un déterminant (ex : *Taiwan*, *la Thaïlande*, *Bornéo*, *les Philippines*).

Critères sémantiques. Il existe plusieurs théories quant à la signification des noms propres. Certains linguistes (S. Mill, K. Kripke, J. Molino, M. Noailly, K. Jonasson, etc.) les considèrent uniquement comme des étiquettes. Pour d'autres linguistes (E. Buyssens, F. Kiefer, M. Gross, etc.), ils ont un sens descriptif (faible ou fort). Enfin, d'autres linguistes les considèrent comme des prédicats de dénomination.

Critères pragmatiques. La signification d'un nom propre peut dépendre de son contexte d'utilisation (*Paris* → une ville de France, une ville des États-Unis, une ville du Canada, etc.).

[1] http://www.cnrtl.fr/lexiques/prolex/

d. *Typologies des noms propres*

M. Tran a présenté plusieurs typologies des noms propres. La plus importante est celle de Grass [Grass, T., 2000]. Cette typologie n'est pas exactement celle utilisée pour la réalisation de PROLEXBASE, mais elle lui a servi de base.

Anthroponymes : patronymes, prénoms, pseudonymes, ethnonymes, groupes musicaux modernes, gentilés[1], hypocoristiques, ensembles artistiques et orchestres classiques, partis et organisations, clubs sportifs, noms donnés aux animaux familiers (zoonymes).

Toponymes : pays, villes, microtoponymes, hydronymes, oronymes, installations militaires, monuments.

Ergonymes : marques, entreprises, établissements d'enseignement et de recherche, titres de livres, de films, de publications et d'œuvres d'art, objets mythiques.

Praxonymes : faits historiques, maladies, événements culturels.

Phénonymes : ouragans, zones de haute et de basse pression, astres et comètes, phénomènes climatiques (ex : *el Niño*).

III.2.2.2 Concepts essentiels venant de Coseriu

III.2.2.2.1 Types de relations de synonymie

La théorie linguistique d'Eugenio Coseriu [Coseriu, E., 1992] distingue trois sous-types de relation dans la relation de synonymie :

- la relation entre un signe linguistique et un objet.
- la relation entre un signe linguistique et d'autres signes linguistiques.
- la relation entre un signe linguistique et le contexte linguistique et situationnel.

III.2.2.2.2 Variations de la relation de synonymie en fonction de caractéristiques de la "situation"

[Coseriu, E., 1998] propose un "diasystème" décrivant les variations de la relation de synonymie en fonction de différentes dimensions :

- selon le temps (dimension diachronique).
- selon l'espace (dimension diatopique).
- selon les caractéristiques sociales des locuteurs (dimension diastratique). Par exemple, 神马(shén mǎ) et 什么(shén me)[2].
- selon les activités qu'ils pratiquent (dimension diaphasique).

Françoise Gadet [Gadet, F., 2003] a proposé une dimension en fonction du canal employé, oral ou écrit (dimension diamésique).

III.2.2.3 Aspects logiciels : PROLEXBASE

III.2.2.3.1 Concepts

Il y a deux notions principales à la base du projet PROLEX : le nom propre conceptuel et le prolexème.

---

[1] Noms des habitants d'un lieu (ex : *Bellifontain* pour habitant de Fontainebleau).

[2] C'est un exemple chinois. Initialement le mot *神*(divinité, shén)*马*(cheval, mǎ) n'a pas une vraie signification, mais les jeunes Chinois l'utilisent (surtout sur le Web) pour remplacer le mot *什么*(shén me, qui a plusieurs sens comme quoi, quel, quelconque etc.). C'est parce que les prononciations se ressemblent : *神马*(shén mǎ), *什么* (shén me).

a. *Le nom propre conceptuel*

Citons ici une partie de la présentation de [Tran, M., 2006].

Pour une langue donnée, des noms propres totalement différents sur le plan graphique peuvent renvoyer à un même et unique référent, et ce phénomène se retrouve généralement d'une langue à l'autre.

Nous définissons le nom propre conceptuel non pas comme le référent, mais plutôt comme un certain point de vue sur celui-ci. Ainsi les noms propres Allemagne en français, Alemania en espagnol, Deutschland en allemand, etc., seront associés à un même nom propre conceptuel, tandis que les noms propres République fédérale d'Allemagne en français, República Federal de Alemania en espagnol, Bundesrepublik Deutschland en allemand, etc. seront associés à un autre nom propre conceptuel. Ces deux noms propres conceptuels seront en relation de synonymie.

Pour définir ces différents points de vue, nous nous sommes basés sur un marquage diasystématique, qui provient des travaux sur la métalexicographie de [Coseriu, E., 1998].

b. *Le prolexème*

On a déjà mentionné la notion de prolexème au I.3.1.1.1 et au III.1.3.2. On peut considérer que le prolexème est une classe d'équivalence de synonymes de noms propres. M. Tran a défini des concepts secondaires pour le prolexème :

- les alias (les variantes, les abréviations, les sigles, les transcriptions etc.), par exemple, *Pékin – Bejing*, *Canal plus – Canal +*, *François Mitterrand – F. Mitterrand*.
- les dérivés (les noms relationnels et les adjectifs relationnels), par exemple, *Parisien* et *parisien*.

III.2.2.3.2 Relations

Après avoir identifié les différents concepts de noms propres, M. Tran précise les relations qui peuvent les relier.

- Synonymie : partage d'un même sens. Il en existe différents types :
  - diachronique (ex. *Zaïre* et *République démocratique du Congo*).
  - diastratique (les variations entre jeunes/personnes âgées, ruraux/urbains, professions différentes, niveaux d'études différents).
  - diaphasique (ex. *Paris* et *Ville lumière*).
- Méronymie : hiérarchisation sur plusieurs niveaux entre les éléments contenants (holonymes) et les éléments contenus (méronymes), par exemple, *arbre/forêt*, *matinée/journée*.
- Accessibilité : notion d'importance, d'entité significative. Par exemple, *Bangkok* est la capitale de la *Thaïlande*.
- Expansion classifiante : notion de caractérisation d'un terme (ex. *Dirigeant politique* et *Président*).
- Éponymie : la relation entre un nom propre et une forme lexicalisée. Elle sert à empêcher la reconnaissance abusive des noms propres. Par exemple, un *bic* = un *stylo-bille*, *Parkinson* ≠ nom propre dans *maladie de Parkinson*.

III.2.2.3.3 Ontologie des noms propres

M. Tran a pris en compte la méthodologie de construction de l'ontologie de Noy et McGuinness [Noy, N. F. & McGuinness, D. L., 2003]. Chaque nom propre conceptuel (pivot) est en relation d'hyperonymie avec un type et une existence.

Pour définir l'ontologie, M. Tran s'est inspiré de la typologie de Grass [Grass, T., 2000] (voir III.2.2.1). Les quatre premiers supertypes identifiés sont :

- les anthroponymes : trait humain ;
- les ergonymes : trait inanimé ;
- les pragmonymes : trait événement ;
- les toponymes : trait locatif.

Il y a aussi 29 sous-types que nous ne listons pas ici. Par exemple, le supertype ergonyme a des sous-types objet, œuvre, produit, vaisseau.

De plus, deux notions ont été ajoutées :

- la notion d'existence, pour préciser le domaine d'appartenance d'un nom propre (ex. historique, fiction, etc.).
- la relation d'hyperonymie (primaire et secondaire), qui décrit le phénomène d'inclusion. La relation d'hyperonymie primaire est la relation la plus usuelle. La relation d'hyperonymie secondaire est la relation complémentaire. Par exemple, le type "Entreprise" relie l'anthroponyme (par exemple Bouygues) en relation d'hyperonymie primaire avec l'organisme nommé d'après lui (par exemple, le groupe Bouygues) et relie l'ergonyme et le toponyme en relation d'hyperonymie secondaire. C'est parce que le terme "Entreprise" est d'abord vu comme un nom (ou l'entreprise elle-même), avant d'être considéré comme une fabrication humaine ou un lieu. Voici les exemples.
  (1) L'entreprise Bouygues a décidé que …
  (2) Il a réussi dans son entreprise avec …
  (3) Il est aujourd'hui au travail à l'entreprise…

III.2.2.3.4 Représentation à quatre niveaux

Il y a quatre niveaux.

Les deux premiers niveaux sont indépendants de la langue. Ce sont :

- le niveau méta-conceptuel : la typologie et l'existence.
- le niveau conceptuel : le nom propre conceptuel (qui constitue un "pivot" entre les langues) et les relations indépendantes des langues.

Les deux derniers niveaux sont dépendants d'une langue :

- le niveau linguistique : le prolexème, les alias, les dérivés et les relations qui dépendent de la langue (dont des fonctions lexico-syntaxiques de I. Mel'čuk).
- le niveau des instances : l'ensemble des formes fléchies d'un lexème d'une langue.

La Figure 40 regroupe les différents concepts utilisés.

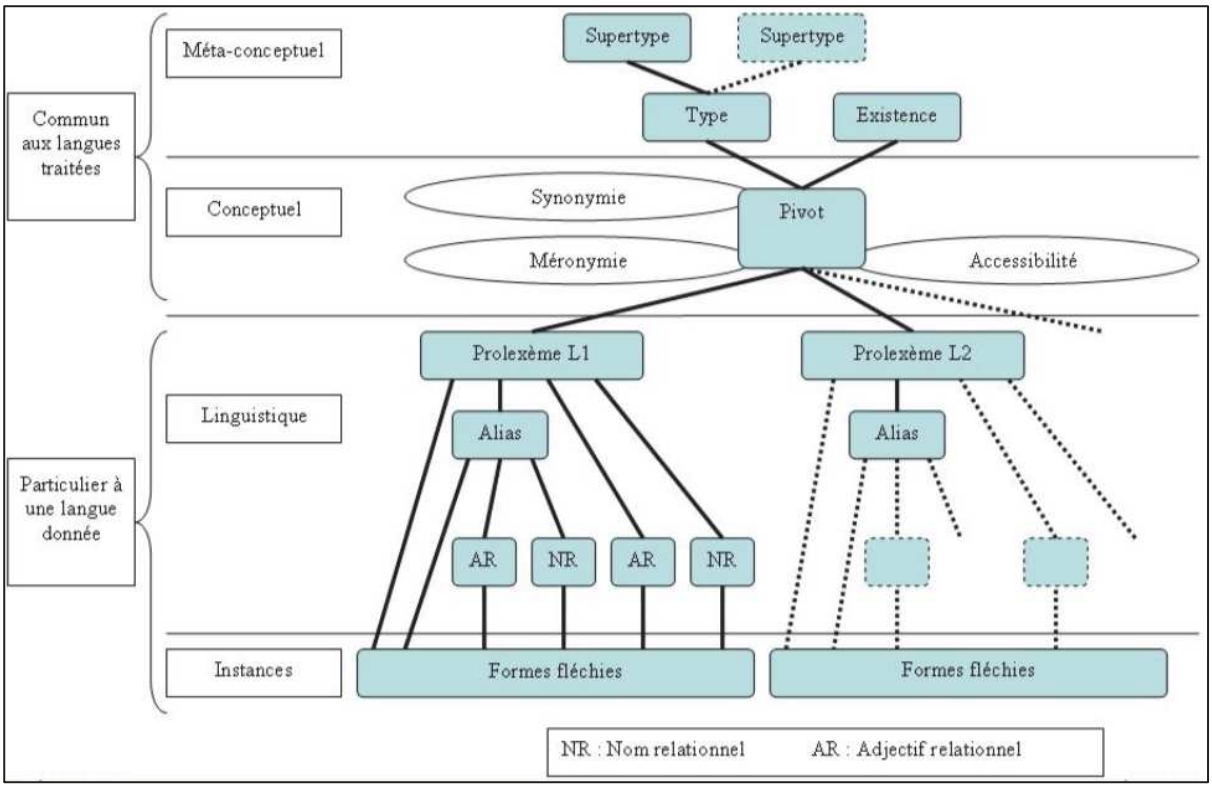


Figure 40 : Modèle à quatre niveaux de PROLEXBASE

III.2.3 Esquisse d'une solution

Notre travail s'est beaucoup inspiré de PROLEXBASE. Par contre, nous ne nous sommes pas limitée aux noms propres, mais nous avons étendu notre modélisation à tous les termes de différents degrés de situement (voir la *Définition 19*), y compris les verbes et les prédicats composés.

Au niveau linguistique, PROLEXBASE est déjà bien complexe, et nous ne voulions pas l'enrichir de ce point de vue. Nous avons préféré simplifier et ne reprendre qu'une partie des notions de PROLEXBASE (surtout l'idée de prolexème) dans notre prototype : PIVAX-3.

D'autre part, notre travail ne limite pas le nombre de langues. Donc la construction de dictionnaires non symétriques comme CJK.ORG ne nous convient pas. Pour la symétrie, nous avons repris les notions de lexie, d'axème et d'axie de PIVAX-2.

Pour l'implémentation, utiliser la plate-forme JIBIKI-2 était la meilleure solution. Nous avons profité des fonctions existantes (ex. gestion des contributions, interfaces etc.) et intégré la notion de prolexème dans la macrostructure de PIVAX-2, ce qui a donné PIVAX-3.

III.3 Conception et implémentation d'une solution basée sur les "liens riches"

III.3.1 1° prototypage chez L&M

On ne pouvait pas intégrer les prolexèmes dans la BDLex de LIBELLEX à cause de contraintes techniques. On ne pouvait pas non plus combiner PIVAX-3 avec LIBELLEX à cause de contraintes industrielles. Finalement, j'ai proposé et implémenté une solution ad hoc.

Dans cette section, on analyse les contraintes techniques et les contraintes industrielles, puis on présente la solution retenue, et une démonstration.

III.3.1.1 **Contraintes techniques**

III.3.1.1.1 Format d'échange de ressources lexicales

Comme l'a vu plus haut (au I.2.2) la BDLex de LIBELLEX a la même structure de BDLex que METRICC, et elle a été conçue à partir des mêmes formats d'échange. Nous présentons d'abord les formats d'échange, puis la BDLex correspondante.

a. *TBXMETRICC*

On a déjà mentionné les formats d'échange au I.2.2.1 et au I.2.2.2. Voici la structure XML des entrées terminologiques spécifiée dans la norme ISO 30042 (TBX standard).

Une entrée terminologique (`<termEntry>`) représente un concept, exprimé dans une ou plusieurs langues (`<langSet>`) au moyen d'un ou plusieurs termes (soit `<tig>`, soit `<ntig>`[1]).

Dans le format TBX standard, deux termes en relation de traduction sont considérés comme appartenant à un même concept ; par exemple, il sont encodés dans deux `<langSet>` différents, à l'intérieur d'une même balise `<termEntry>`.

Dans le format TBXMETRICC, deux termes en relation de traduction apparaissent dans des concepts (`<termEntry>`) différents. La relation de traduction est matérialisée au moyen d'une balise `<descrip>`, les reliant au niveau `<langSet>` et non au niveau `<termEntry>`.

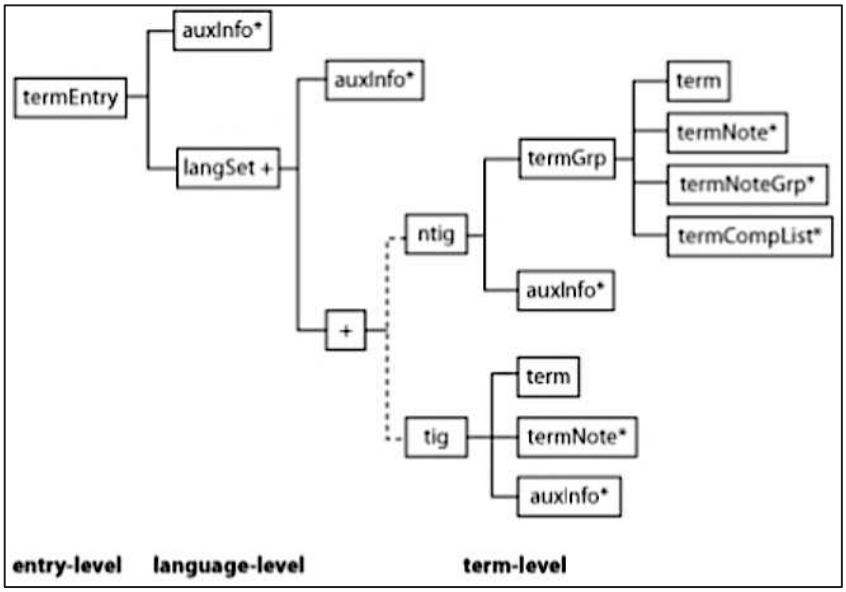


Figure 41 : Structure TBX standard

Dans certains articles, E. Delpech a présenté `<langSet>` en disant que c'est le niveau des sens, et que `<tig>` ou `<ntig>` est le niveau des termes (mot-vedette et variante).
Cependant, cette présentation est contraire à l'explication qu'elle donne dans la spécification interne du format TBXMETRICC (voir I.2.2.2.1).

Il nous semble qu'en fait `<langSet>` est le regroupement (complet ou partiel) des différents termes (`<tig>` ou `<ntig>`) de même sens.

D'une part, pour chaque balise `<langSet>`, les sous-balises `<tig>` ou `<ntig>` introduisent des termes de même sens. D'autre part, on peut avoir plusieurs `<langSet>` différents dans des `<termEntry>` différents pour une même langue, qui décrivent les mêmes sens. Il n'y a pas de relation monolingue entre deux entrées différentes.

Ainsi, TBXMETRICC ne fournit aucun moyen pour vraiment décrire un sens comme un objet unique. C'est une organisation un peu trouble.

[1] La balise `<ntig>` permet une représentation plus complexe d'un terme, notamment son découpage en éléments via la balise `<termCompList>`.

Nous sommes donc plutôt d'accord avec l'explication de E. Delpech au I.2.2.2.1. La relation de traduction au niveau `<langSet>` est l'équivalence sémantique entre termes simples ou composés, et c'est tout.

b. *TEIMETRICC*

TEIMETRICC permet d'encoder uniquement le découpage en phrases des textes dont sont extraits les glossaires METRICC. La structure d'entrée de TEIMETRICC est définie ci-dessous.

```
<TEI xml:id="IDENTIFIANT UNIQUE">
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title>TITRE DU TEXTE</title>
         </titleStmt>
         <!-- Liste des types-mime ici : http://fr.wikipedia.org/wiki/Type_mime -->
         <sourceDesc target="URI RELATIVE DU DOCUMENT" mimeType="TYPE MIME DU DOCUMENT">
            <p>description éventuelle du document original (elle peut être vide)</p>
         </sourceDesc>
      </fileDesc>
   </teiHeader>
   <text>
      <body>
         <s xml:id="IDENTIFIANT UNIQUE"><![CDATA[UNE PHRASE]]></s>
         <s xml:id="IDENTIFIANT UNIQUE"><![CDATA[UNE PHRASE]]></s>
         <s xml:id="IDENTIFIANT UNIQUE"><![CDATA[UNE PHRASE]]></s>
      </body>
   </text>
</TEI>
```

Figure 42 : Définition d'une entrée TEI

c. *TSV*

On a introduit TSV au III.1.1.2. L'import et l'export en format TSV ont été développés spécialement pour LIBELLEX après l'intégration de la BDLex de METRICC dans LIBELLEX. On peut noter que ce format n'est pas utilisé par METRICC.

III.3.1.1.2 Analyse de la structure de la base lexicale

On a brièvement présenté la structure de la BDLex de LIBELLEX au I.2.2.1 et au I.2.2.2. Nous devons ici aller plus dans le détail.

Il y a deux types de table : statique et dynamique. Les tables statiques sont créées une fois pour chaque BDLex lors de l'installation d'une instance de LIBELLEX. Voir l'Annexe 7, qui donne le schéma de la base de données. Les tables en bleu sont les tables statiques.

Les tables dynamiques (les tables en jaune dans l'Annexe 7) sont créées lors de l'import d'un nouveau glossaire. Il y a deux sous-types de table dynamique, les tables de description et les tables de liaison. Les tables de description permettent de stocker les contenus des entrées, et les tables de liaison permettent de stocker les informations de relation. En voici quelques-unes.

- `M_??_SETS`[1] : table stockant les informations correspondant à la balise `<langSet>`.
- `M_??_TIGS` : table stockant les termes (vedettes et variantes) des balises `<tig>` et `<ntig>`.
- `M_??_TIGS_SETS` : correspondances entre entrées de `SETS` et de `TIGS`.
- `M_??_CROSSLINGREL` : relation de traduction entre deux `SETS`.

[1] ?? est le numéro de ressource lexicale (on l'appelle glossaire), par exemple, `M_1_TIGS`, `M_37_TIGS`, etc.

III.3.1.2 **Contraintes industrielles**

Outre la mauvaise qualité des ressources terminologiques fournies à L&M par ses clients (voir III.1.2.1), nous avons rencontré des problèmes de coût de maintenance et des limites fortes aux évolutions possibles chez les clients.

III.3.1.2.1 Coût de maintenance

Au début, nous avons proposé d'utiliser JIBIKI comme plate-forme sous-jacente à LIBELLEX. L&M a refusé cette solution, parce qu'il n'y avait aucun permanent de L&M qui connaissait la plate-forme JIBIKI. Même si JIBIKI est en source ouvert, après ma thèse, il aurait fallu avoir au moins une personne pour la maintenance.

III.3.1.2.2 Limites aux évolutions chez les clients

D'autre part, les systèmes LIBELLEX sont installés indépendamment chez les clients comme des instances. Comme LIBELLEX fonctionnait déjà chez des clients, on ne pouvait pas faire de gros changements des bases lexicales dans les instances de LIBELLEX installées chez les clients. On n'aurait pu le faire que par des plugins, mais ça aurait toujours dû être compatible avec les ressources anciennes.

III.3.1.3 **Spécification et implémentation d'une solution ad hoc**

III.3.1.3.1 Solution ad hoc proposée

Nous[1] avons proposé une solution à deux niveaux, celui du modèle de BDLex et celui des instances spécialisées.

La BDLex a été enrichie avec un champ `type` de valeur libre dans plusieurs tables.

- Le `type` dans la table de stockage des termes (`TIGS`), peut être `mot-vedette`, `acronyme`, `abréviation`, `variante non typée`, etc.
- Le `type` dans la table de stockage des relations sémantiques (`LEXSEMREL`), peut être `parent` ou `enfant` (c'est le cas dans la base de Louis Vuitton).
- Ces valeurs de `type` sont faciles à changer/ajouter selon les besoins des clients.
- On a également enrichi les statuts de validation pour représenter la qualité.

On a déjà dit que le format TSV est utilisé principalement pour les imports des données des clients chez LIBELLEX. Nous avons développé une fonction d'import complexe à partir d'un fichier TSV pour améliorer les échanges de sources (les relations bilingues, les synonymes et les relations hiérarchiques parent/enfant etc). La Figure 43 montre l'interface d'import actuel de LIBELLEX.

D'autre part, selon les besoins des clients, on a développé plusieurs formats spécialisés (par exemple pour le client EXALEAD). La Figure 44 montre l'interface d'export actuel de LIBELLEX.

[1] Il n'y a pas que moi qui ai travaillé sur le sujet, mais aussi mes collègues. Le patron de L&M, F. Brown de Colstoun, communiquait avec les clients et spécifiait les besoins. Le chef E. Monneret était responsable technique, proposait les conceptions globales, puis vérifiait et validait les travaux. M. Morardo a développé l'interface d'affichage de consultation. Quant à moi, j'ai travaillé sur l'import, l'export, leurs interfaces et les traitements de la BDLex.

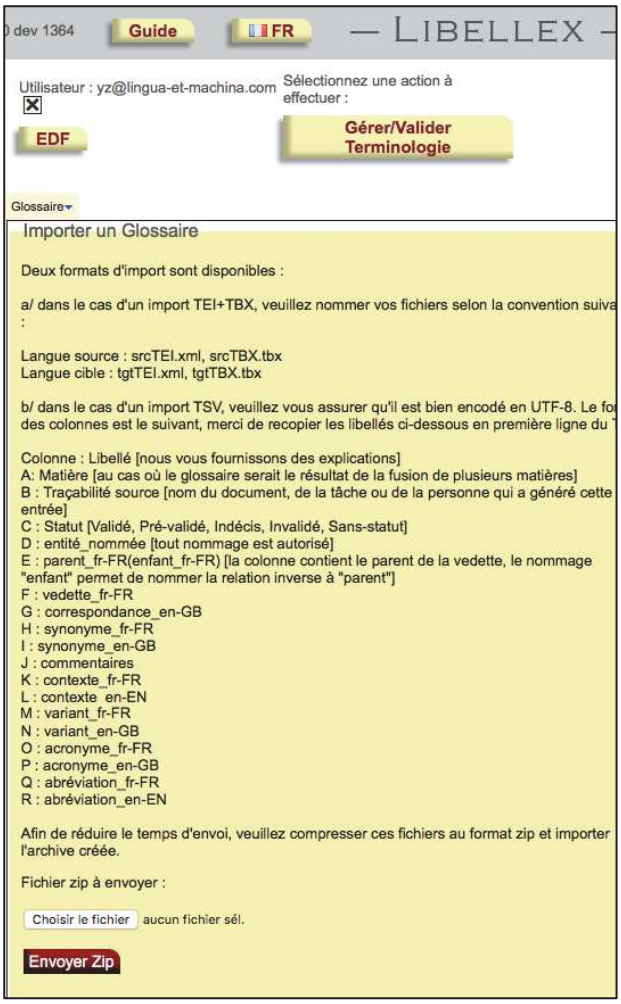


Figure 43 : L'interface d'import de LIBELLEX

Figure 44 : L'interface d'export de LIBELLEX

III.3.1.3.2 Résultats et validation

Dans certains cas, comme celui de Louis Vuitton, on s'est limité à une seule entrée dans la BDLex pour chaque terme. Par exemple, *SAC DE VILLE* ou *ORANGE* apparaissent dans plusieurs lignes dans la ressource (voir la Figure 37).

Si le terme est déjà créé dans la BDLex, on ne crée que la relation (relation parent/enfant et relation de traduction). Dans ce cas, on considère que le terme est au niveau du sens. Ce n'est certainement pas une solution totalement satisfaisante, et on ne peut pas faire la même chose pour tous les autres clients. Mais on a pu faire comme ça pour quelques autres clients.

III.3.1.4 **Démonstration**

La figure ci-dessous montre un exemple de Louis Vuitton : l'affichage pour la consultation du mot *bleu jean* avec les relations monolingues et la relation de traduction.

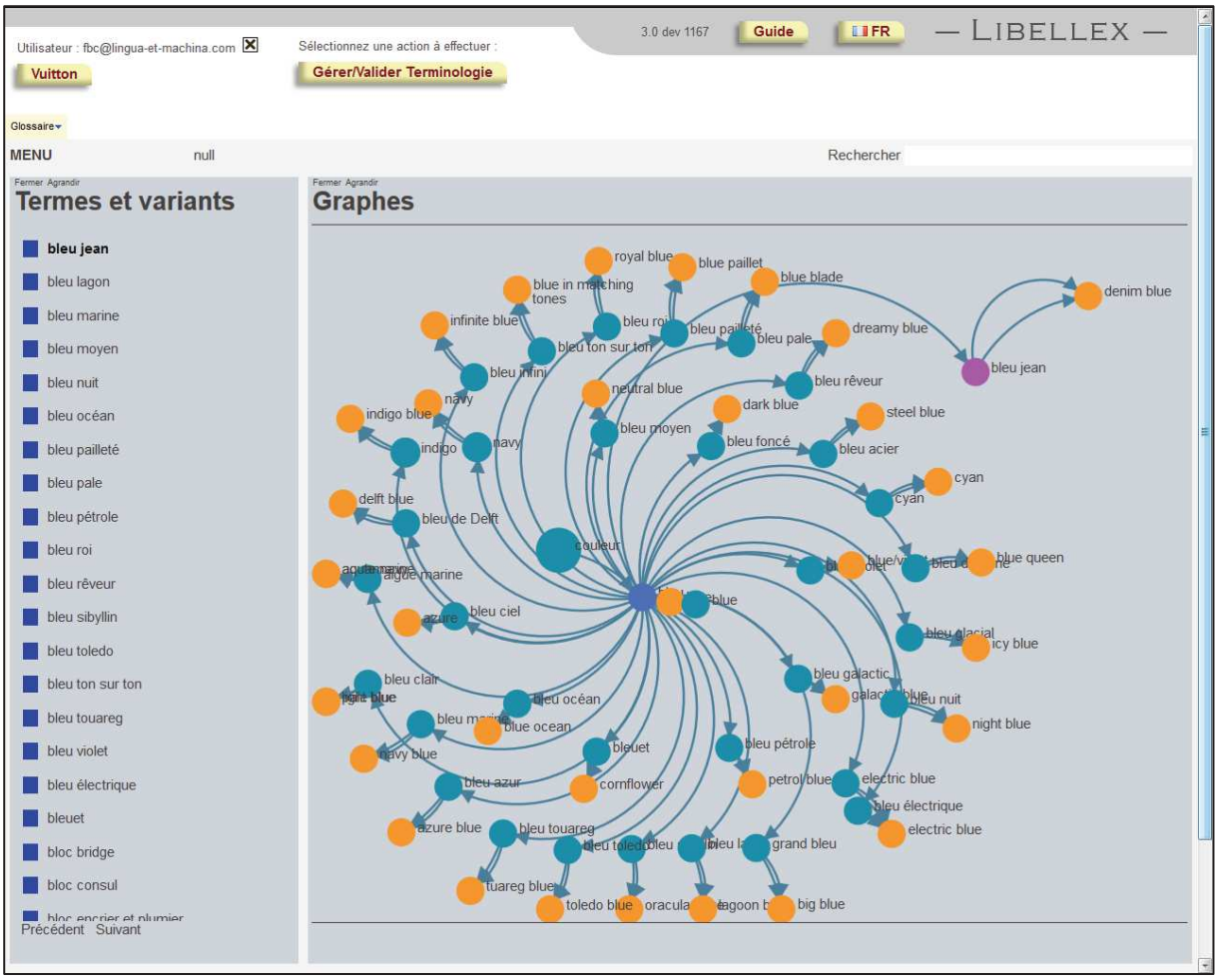


Figure 45 : Consultation de bleu jean sur l'interface de LIBELLEX

III.3.2 Une solution plus générale basée sur JIBIKI-2 : PIVAX-3

Au niveau du laboratoire, il a été possible d'aller plus loin en utilisant la plate-forme JIBIKI-2, qui permet d'implémenter de façon naturelle les différents types d'objets lexicaux et leurs liens. Cela nous a permis de produire un nouveau type de BDLex, PIVAX-3.

III.3.2.1 Extension de l'architecture de PIVAX-2

Notre but était d'unifier les 3 types de données lexicales : mots (simples ou composés) généraux, termes (liés à un domaine), et prolexèmes. Pour simplifier la conception, on a considéré qu'un terme est un type de prolexème.

III.3.2.1.1 Macrostructure

a. *Types de volumes repris de PIVAX-2*

On a repris les trois types de volumes de PIVAX-2 : lexie, axème et axie (voir II.2.2.1).

b. *Nouveaux types de volumes*

On a repris et enrichi la notion de prolexème et on a introduit une nouvelle notion, celle de proaxie.

Définition 23. ***Prolexème.*** Dans une BDLex PIVAX-3, il y a un seul volume de prolexèmes pour chaque langue. Dans ce volume, les prolexèmes regroupent les lexies qui représentent le même sens mais dont la réalisation syntaxique est différente (forme de surface, classe grammaticale, etc.).

Au contraire de M. Tran, notre notion de prolexème n'est pas limitée aux noms propres. Les liens bidirectionnels entre les lexies et leurs prolexèmes sont marqués avec une étiquette libre (par exemple, alias, acronyme, dérivation, définition, etc.).

Par exemple, l'entrée de type prolexème *fra.organisation_des_nations_unies.1* est reliée aux entrées de type lexie :

- *ONU*, par un lien étiqueté `acronyme`.
- *nations unies*, par un lien étiqueté `alias`.
- *onusien*, par un lien étiqueté `dérivation`.
- *organisation des nations unies*, par un lien étiqueté `définition`. Ce lien n'est pas la définition lexicographique du prolexème, mais caractérise seulement le terme préféré pour le décrire.

Définition 24. ***Proaxie.*** Il y a un seul volume de proaxies dans une instance de PIVAX-3. Les proaxies regroupent les prolexèmes de langues différentes partageant un même sens.

Les liens entre une entrée de proaxie et les entrées de prolexèmes sont bidirectionnels. Par exemple, dans un dictionnaire trilingue français-anglais-chinois, l'entrée de proaxie *proaxie.united_nations.1* relie les entrées :

- *fra.organisation_des_nations_unies.1* du volume des prolexèmes français,
- *eng.united_nations.1* du volume des prolexèmes anglais,
- *zho.联合国.1* du volume des prolexèmes chinois.

c. *Macrostructure complète*

Dans cette macrostructure, nous avons deux couches : une couche basique et une couche "Pro". Dans la couche basique, nous gérons trois types de volume : les volumes de lexies, les volumes d'axèmes et le volume d'axies. Dans la couche "Pro", nous gérons deux types de volume : les volumes de prolexèmes et le volume des proaxies.

Grâce à la couche basique, nous pouvons relier les lexies qui se correspondent exactement, comme l'acronyme français *ONU*, relié à l'acronyme anglais *UN*.

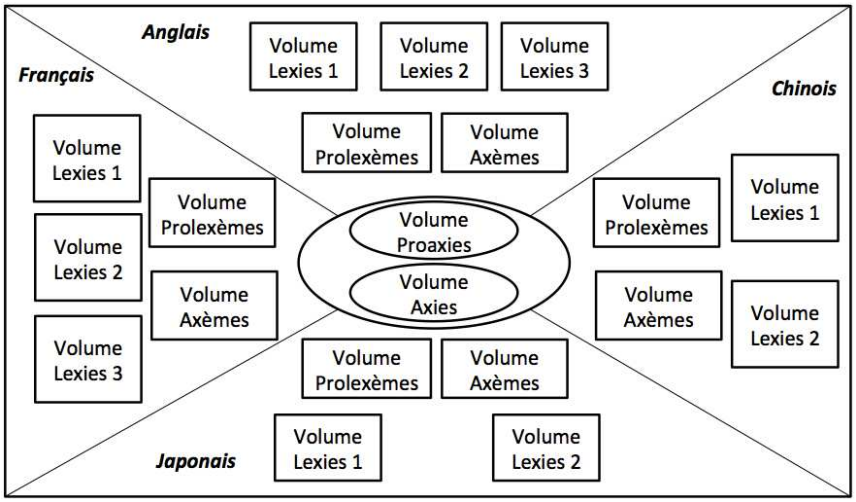


Figure 46 : Macrostructure de PIVAX-3

Grâce à la couche "Pro", nous pouvons proposer en traduction des lexies des langues cible de même sens. Par exemple, en chinois, il y a un seul mot *联合国* (lián hé guó) pour ce sens, et il n'existe pas d'acronyme. Donc on peut toujours proposer le même terme *联合国* pour la traduction de *ONU* et la traduction de *organisation des nations unies*. Voir la Figure 47.

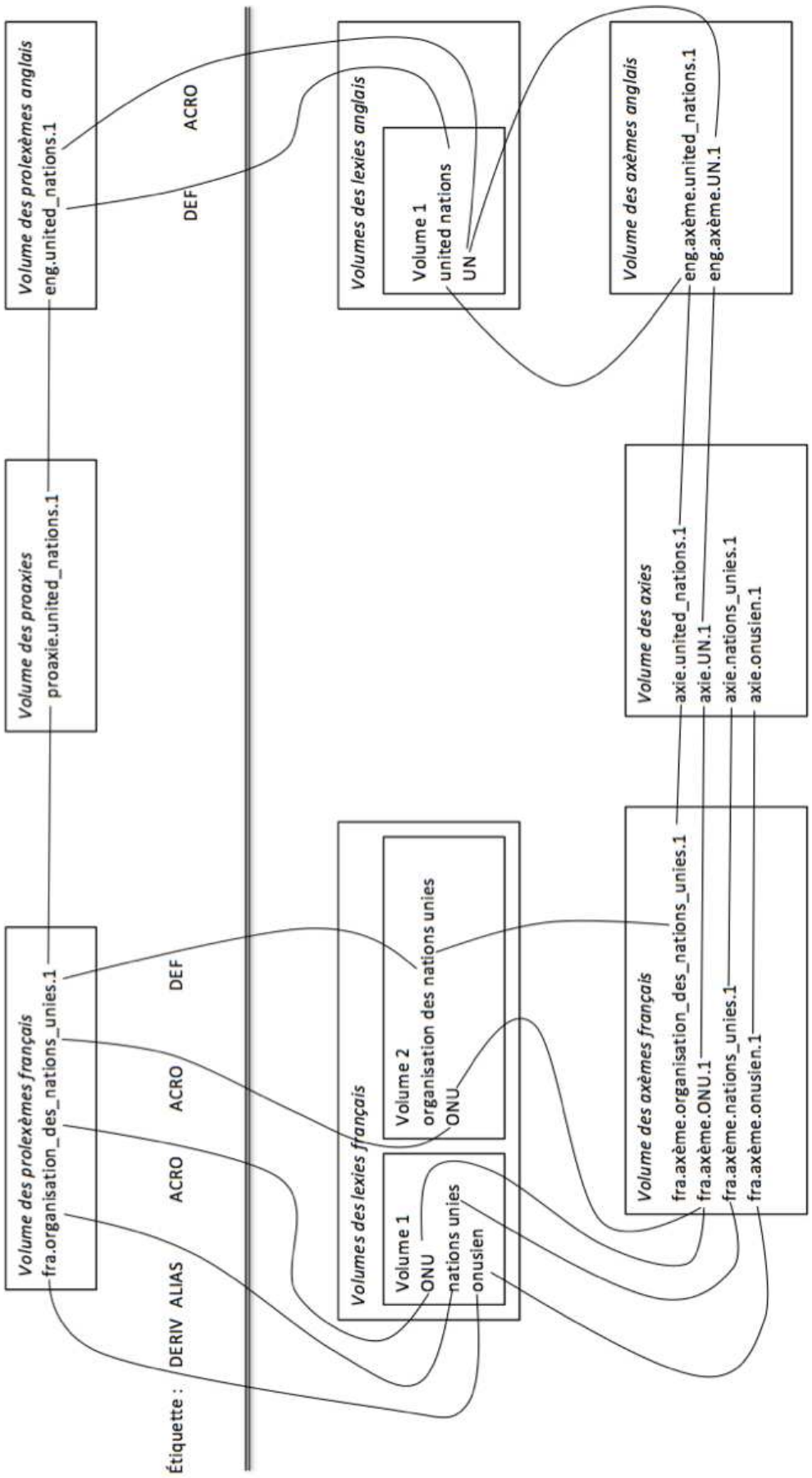


Figure 47 : Exemple des liens dans PIVAX-3

La notion d'étiquette a pour but de proposer les meilleures traductions. Par exemple, en japonais, *国際連合* (kokusai-rengō) est la lexie de même sens que *Organisation des Nations Unies*, et son acronyme est *国連* (kokuren). Cet acronyme utilise le premier et le troisième kanji de ce mot (composé), ce qui est différent des initiales de la lexie de définition (le cas de *ONU* et de *UN*). Il existe peut-être une langue qui a deux acronymes, l'un correspondant à l'acronyme des initiales, l'autre correspondant à une sélection de caractères ou de mots. Donc, nous avons décidé de ne pas relier ces deux acronymes de types différents à une même axie. Par contre, comme ce sont des acronymes, pour la traduction de *ONU*, *国連* est meilleur que *国際連合*. On donnera trois niveaux de précision de traduction au III.3.2.2.3.

III.3.2.1.2 Microstructures

a. *Microstructure des lexies*

Dans notre prototype, nous avons utilisé deux microstructures pour les volumes de lexies. La première est similaire à celle de PIVAX-2. Un volume a une entrée par lexie. Au niveau d'une lexie, on a le lemme, la partie du discours, les définitions multilingues, les informations des liens vers les prolexèmes et vers les axèmes.

```
<p:volume xmlns:p=……>
  <p:lexie p:id="Acro.fra.ONU.1">
    <p:lemma>ONU</p:lemma>
    <p:pos>n</p:pos>
    <p:definitions>
      <p:definition d:lang="fra">Initiales de « Organisation des Nations Unies ».
      </p:definition>
    </p:definitions>
    <p:entryref type="prolexeme" volume="Acro_fra-prolexeme "
      p:idref="Acro.prolexeme.fra.Organisation_des_nations_unies.1" lang="fra" p:relation-
      mono="ACRO"/>
    <p:entryref type="axeme" volume="Acro_fra-axeme" p:idref="Acro.axeme.fra.ONU.1" lang="fra"
      p:relation-mono=""/>
  </p:lexie>
  <p:lexie p:id="Acro.fra.Nations_unies.1">
    <p:lemma>Nations unies</p:lemma>
    <p:pos>n</p:pos>
    <p:definitions>
      <p:definition d:lang="fra">Alias de « Organisation des Nations Unies ».</p:definition>
    </p:definitions>
    <p:entryref type="prolexeme" volume="Acro_fra-prolexeme"
      p:idref="Acro.prolexeme.fra.Organisation_des_nations_unies.1" lang="fra" p:relation-
      mono="ALIAS"/>
    <p:entryref type="axeme" volume="Acro_fra-axeme"
      p:idref="Acro.axeme.fra.Nations_unies.1" lang="fra" p:relation-mono=""/>
  </p:lexie>
  ……
</p:volume>
```

Figure 48 : Exemple de la première microstructure des lexies

La seconde microstructure est conforme à la proposition de V. Dikonov. Un volume a une entrée par vocable. Au niveau d'un vocable, il y a une sous-entrée par lexie.

```
<p:volume xmlns:p=……>
  <p:vocable p:id="Acro.fra.vocable.CNAM.1">
    <p:lemma>CNAM</p:lemma>
    <p:lexie p:id="Acro.fra.CNAM.1">
      <p:pos>n.f.</p:pos>
      <p:definitions>
        <p:definition d:lang="fra"> Acronyme de la « Caisse Nationale de l'Assurance Maladie
          des travailleurs salariés ».</p:definition>
      </p:definitions>
      <p:entryref type="prolexeme" volume="Acro_fra-prolexeme"
```

```
        p:idref="Acro.prolexeme.fra.Caisse_nationale_de_l'assurance_maladie_des_travailleurs
        _salariés.1" lang="fra" p:relation-mono="ACRO"/>
      <p:entryref type="axeme" volume="Acro_fra-axeme" p:idref="Acro.axeme.fra.CNAM.1"
        lang="fra" p:relation-mono=""/>
    </p:lexie>
    <p:lexie p:id="Acro.fra.CNAM.2">
      <p:pos>n.m.</p:pos>
      <p:definitions>
        <p:definition d:lang="fra">Acronyme du « Conservatoire National des Arts et
          Métiers ».</p:definition>
      </p:definitions>
      <p:entryref type="prolexeme" volume="Acro_fra-prolexeme"
        p:idref="Acro.prolexeme.fra.Conservatoire_national_des_arts_et_métiers.1" lang="fra"
        p:relation-mono="ACRO"/>
      <p:entryref type="axeme" volume="Acro_fra-axeme" p:idref="Acro.axeme.fra.CNAM.2"
        lang="fra" p:relation-mono=""/>
    </p:lexie>
  </p:vocable>
</p:volume>
```

Figure 49 : Exemple de la deuxième microstructure (vocable > lexie)

b. *Microstructure des axèmes*

La microstructure des axèmes est simple. Une entrée ne comporte que des liens (vers une ou plusieurs lexies et vers une axie).

```
<p:axeme p:id="Acro.axeme.fra.ONU.1">
        <p:entryref type="final[1]" volume="Acro_fra"  p:idref="Acro.fra.ONU.1" lang="fra"
          p:relation-mono=""/>
        <p:axiref type="axie" volume="Acro_axie" p:idref="Acro.axie.UN.1" lang="axie"
          p:relation-mono=""/>
</p:axeme>
```

Figure 50 : Exemple de la microstructure d'un volume d'axèmes

c. *Microstructure des axies*

La microstructure des axies est également simple. Une entrée ne comporte que des liens vers les axèmes de chaque espace lexical (celui d'une langue naturelle ou éventuellement celui d'UNL).

```
<p:axie id="Acro.axie.United_Nations.1">
        <p:item-links link_group="g1">
                <p:item relation="" volume="Acro_eng-axeme" type="axeme"
                    p:idref="Acro.axeme.eng.United_Nations.1" lang="eng"/>
                <p:item relation="" volume="Acro_fra-axeme" type="axeme"
                    p:idref="Acro.axeme.fra.Organisation_des_nations_unies.1" lang="fra"/>
                <p:item relation="" volume="Acro_zho-axeme" type="axeme"
                    p:idref="Acro.axeme.zho.连合国.1" lang="zho"/>
                <p:item relation="" volume="Acro_jpn-axeme" type="axeme"
                    p:idref="Acro.axeme.jpn.国際連合.1" lang="jpn"/>
        </p:item-links>
</p:axie>
```

Figure 51 : Exemple de la microstructure des axies

d. *Microstructure des prolexèmes*

Une entrée de prolexème se compose de liens vers les lexies, avec les étiquettes, et de liens vers les proaxies, sans étiquette.

```
<p:prolexeme p:id="Acro.prolexeme.jpn.国際連合.1">
  <p:entryref type="final" volume="Acro_jpn"  p:idref="Acro.jpn.国際連合.1" lang="jpn"
    p:relation-mono="DEF"/>
```

[1] Le type final est implémenté pour le type lexie.

```
  <p:entryref type="final" volume="Acro_jpn"  p:idref="Acro.jpn.国連.1" lang="jpn" p:relation-
    mono="ACRO"/>
  <p:axiref type="proaxie" volume="Acro_proaxie" p:idref="Acro.proaxie.United_Nations.1"
    lang="proaxie" p:relation-mono= ""/>
</p:prolexeme>
```

Figure 52 : Exemple de la microstructure des prolexèmes

e. *Microstructure des proaxies*

La microstructure des proaxies est également simple. Une entrée de proaxie ne contient que des liens vers des prolexèmes situés en général dans plusieurs espaces lexicaux.

```
<p:proaxie id="Acro.proaxie.United_Nations.1">
  <p:link relation="" volume="Acro_eng-prolexeme" type="prolexeme" p:idref="Acro.prolexeme.
    eng.United_Nations.1" lang="eng"/>
  <p:link relation="" volume="Acro_fra-prolexeme" type="prolexeme" p:idref="Acro.prolexeme.
    fra.Organisation_des_nations_unies.1" lang="fra"/>
  <p:link relation="" volume="Acro_zho-prolexeme" type="prolexeme" p:idref="Acro.prolexeme.
    zho.联合国.1" lang="zho"/>
  <p:link relation="" volume="Acro_jpn-prolexeme" type="prolexeme" p:idref="Acro.prolexeme.
    jpn.国際連合.1" lang="jpn"/>
</p:proaxie>
```

Figure 53 : Exemple de la microstructure des proaxies

III.3.2.1.3 Utilisation de liens riches

Ce que nous appelons "lien riche" a été présenté au II.3.1. Nous présentons ici les utilisations des liens riches pour réaliser la gestion d'une terminologie (ensemble de termes "situés").

Pour modéliser les relations "situées", on a besoin des étiquettes portées par les liens entre les entrées de lexie et de prolexème. Voir le schéma Figure 47, et des exemples dans la Figure 48 et dans la Figure 52.

On a implémenté cette relation de "situement" dans un champ `étiquette` de valeur libre. Ce champ était prévu (mais jamais utilisé) dans la table `links` de JIBIKI-2 avec le nom `label`. Comme les liens sont orientés et bidirectionnels, nous avons dû stocker les étiquettes dans les tables des lexies et dans les tables des prolexèmes.

La Figure 54 ci-dessous montre des informations portées par certains liens de l'entrée *Acro.prolexeme.fra.Organisation_des_nations_unies.1*, stockées dans la table `links` du volume des prolexèmes français.

L'entrée *Acro.prolexeme.fra.Organisation_des_nations_unies.1* est stockée dans la table des entrées du volume des prolexèmes français. On a créé un lien en lui donnant un identifiant, ici *38946301* (une clé étrangère du champ `entryid` de la table `links` vers le champ `objectid` de la table des entrées), créé automatiquement par le système. Voir la Figure 28.

targetid character varying(255)	elementid character var	name character var	lang character var	volumetarget character varying	type character var	label character var	entryid numeric(19,0	weight numeric(3,2)	objectid [PK] numeric(	obj int
Acro.fra.ONU.1	''	final	fra	Acro_fra	final	ACRO	38946301	0.00	46650030	0
Acro.fra.Nations_unies.1	''	final	fra	Acro_fra	final	ALIAS	38946301	0.00	46650037	0
Acro.fra.Organisation_des_nations_unies.1	''	final	fra	Acro_fra	final	DEF	38946301	0.00	46650038	0
Acro.fra.onusien.1	''	final	fra	Acro_fra	final	DERIV	38946301	0.00	46650039	0
Acro.proaxie.United_Nations	''	proaxie	proaxie	Acro_proaxie	proaxie	''	38946301	0.00	46650040	0

Figure 54 : Exemple de l'utilisation d'étiquettes libres dans le volume des prolexèmes français

III.3.2.2 Implémentation de PIVAX-3

III.3.2.2.1 Traitement de la variété des microstructures

Dans PIVAX-1 (voir II.2.2.2), on ne peut utiliser qu'une seule microstructure pour tous les volumes d'un même espace lexical. Par contre, comme on l'a dit au III.3.2.1.2, PIVAX-3 permet

d'avoir des volumes de microstructures différentes dans le même espace lexical. Nous avons utilisé deux microstructures différentes pour les volumes des lexies françaises.

Pour l'implémentation purement technique, M. Mangeot a proposé, en plus du pointeur `entry`, qu'on ajoute un pointeur supplémentaire : `sens` (`cdm-sense-id`). Ci-dessous, les éléments CDM dans les deux fichiers de métadonnées correspondent aux exemples de la Figure 48 et de la Figure 49.

Éléments CDM correspondant à l'exemple de la Figure 48.

```
<cdm-elements>
  <cdm-volume      xpath="/p:volume"/>
  <cdm-entry       xpath="/p:volume/p:lexie"/>
  <cdm-entry-id    xpath="/p:volume/p:lexie/@p:id"/>
  <cdm-headword    xpath="/p:volume/p:lexie/p:lemma/text()"/>
  <cdm-pos         xpath="/p:volume/p:lexie/p:pos/text()" />
  <cdm-definition  xpath="/p:volume/p:lexie/p:definitions/p:definition/text()"/>
  <!-- cdm-sense-id    xpath=""-->
  <links>
    <link name="axeme" xpath="/p:volume/p:lexie/p:entryref[@type='axeme']">
      <type    xpath="@type" />
      <volume  xpath="@volume" />
      <value   xpath="@p:idref" />
      <lang    xpath="@lang" />
      <label   xpath="@p:relation-mono" />
    </link>
    <link name="prolexeme" xpath="/p:volume/p:lexie/p:entryref[@type='prolexeme']">
      <type    xpath="@type" />
      <volume  xpath="@volume" />
      <value   xpath="@p:idref" />
      <lang    xpath="@lang" />
      <label   xpath="@p:relation-mono" />
    </link>
  </links>
</cdm-elements>
```

Éléments CDM correspondant à l'exemple de la Figure 49.

```
<cdm-elements>
  <cdm-volume      xpath="/p:volume"/>
  <cdm-entry       xpath="/p:volume/p:vocable"/>
  <cdm-entry-id    xpath="/p:volume/p:vocable/@p:id"/>
  <cdm-headword    xpath="/p:volume/p:vocable/p:lemma/text()"/>
  <cdm-sense-id    xpath="/p:volume/p:vocable/p:lexie/@p:id"/>
  <cdm-pos         xpath="/p:volume/p:vocable/p:lexie/p:pos/text()" />
  <cdm-definition  xpath="/p:volume/p:vocable/p:lexie/p:definitions/p:definition/text()"/>
  <links>
    <link name="axeme" xpath="/p:volume/p:vocable/p:lexie/p:entryref[@type='axeme']">
      <type    xpath="@type" />
      <volume  xpath="@volume" />
      <value   xpath="@p:idref" />
      <lang    xpath="@lang" />
      <label   xpath="@p:relation-mono" />
    </link>
    <link name="prolexeme" xpath="/p:volume/p:vocable/p:lexie/p:entryref[@type='prolexeme']">
      <type    xpath="@type" />
      <volume  xpath="@volume" />
      <value   xpath="@p:idref" />
      <lang    xpath="@lang" />
      <label   xpath="@p:relation-mono" />
    </link>
  </links>
</cdm-elements>
```

Figure 55 : CDM correspondant aux deux exemples de microstructure de PIVAX-3

Si le pointeur `sens` (`cdm-sense-id`) est vide (cas de la Figure 48), on prend le pointeur `entrée` (`cdm-entry-id`) pour accéder à une entrée dans la BDLex.

Si le pointeur `sens` n'est pas vide (cas de la Figure 49), on l'utilise pour accéder à une entrée dans la BDLex. On garde toujours le pointeur `entrée`, de façon à pouvoir récupérer les informations de mot-vedette (`cdm-headword`). En effet, le mot-vedette n'est pas au niveau `sens`, mais au niveau `entrée`.

III.3.2.2.2 Algorithmes de calcul

a. *Réalisation informatique*

La réalisation informatique est basée sur plusieurs algorithmes. Nos algorithmes sont un peu compliqués, nous en donnons le pseudo-code à l'Annexe 8. Ici, on en donne seulement une brève présentation.

Le premier est l'algorithme de collecte des liens. Il permet de chercher tous les liens possibles dans l'ensemble des liens riches de tous les volumes pour une entrée recherchée, et de réaliser le parcours des liens riches. Il enchaîne les étapes suivantes :

- Chercher les lexies source et leurs liens.
- Chercher les liens de lexies source vers les axèmes source, puis vers les axies, ensuite vers les axèmes cible, enfin vers les lexies cible.
- Chercher les liens de lexies source vers les prolexèmes source puis vers les proaxies, ensuite vers les prolexèmes cible, à la fin vers les lexies cible, et comparer les étiquettes portées par les lexies/prolexèmes source et par les lexies/prolexèmes cible.

Le deuxième est l'algorithme de construction du résultat. Il s'agit principalement de notre stratégie des trois niveaux de traduction, qui sera présentée dans la section suivante III.3.2.2.3.

b. *Exemple : diagramme de calcul pour la recherche de "TGV"*

Quand on cherche le mot *TGV* vers l'anglais et le chinois, on trouve deux lexies : (1) *TGV* pour l'acronyme de *Train à Grande Vitesse*, et (2) *TGV* pour l'acronyme de *Transposition des Gros Vaisseaux* (terminologie médicale).

D'une part, on recherche les traductions en passant par les axèmes et les axies comme avec PIVAX-2, et on trouve une traduction en anglais : *TGV* pour l'acronyme de *Transposition of the Great Vessels*.

D'autre part, on recherche les traductions en passant par les prolexèmes et les proaxies, et on trouve une suite de liens et de traductions, voir la figure ci-dessous. Pour faciliter la lecture, on a utilisé des couleurs différentes et on a marqué des numéros pour chaque étape de la recherche par des liens.

Par exemple, 1.a et 1.b correspondent à la recherche des liens des lexies source vers les prolexèmes source. 2.a et 2.b correspondent à la recherche des liens des prolexèmes source vers les proaxies. De 3.a à 3.f, ce sont les étapes de la recherche des liens des proaxies vers les prolexèmes cible. De 4.a à 4.i, ce sont les étapes de la recherche des liens des prolexèmes cible vers les lexies cible.

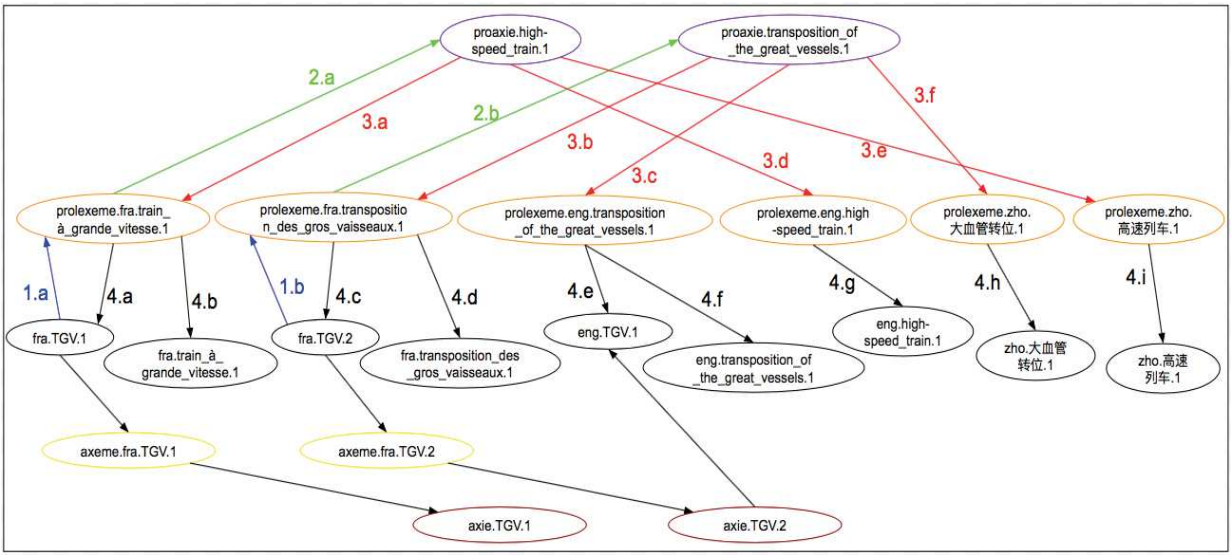


Figure 56 : Exemple de calcul des liens dans PIVAX-3

III.3.2.2.3 Niveaux des traductions

a. *Trois niveaux théoriques*

Nous proposons trois niveaux de traduction classés selon la précision obtenue.

(1) Le système trouve une lexie directement, en passant par le volume des axèmes et par le volume des axies. C'est le premier niveau de traduction, et le plus précis.

Pour l'exemple de la Figure 56, c'est le cas de *TGV* pour le sens d'acronyme *transposition des gros vaisseaux* en français vers le même sens *TGV* en anglais.

(2) Le système cherche le lien dans le volume des prolexèmes de la langue source avec une étiquette. Puis il parcourt le volume des proaxies, et ensuite le volume des prolexèmes et les volumes des lexies des langues cible. Il trouve une lexie avec la même étiquette. C'est le deuxième niveau, dit niveau intermédiaire.

Par exemple, à la fin de la section III.3.2.1.1, on a expliqué que, en japonais *国連* est meilleur que *国際連合* pour la traduction de *ONU*, parce que ces mots portent l'étiquette `acronyme`.

(3) Le système trouve les lexies par prolexème et proaxie sans prendre en compte l'étiquette. Ces lexies proposées constituent le troisième niveau, le moins précis.

Par exemple, dans la Figure 56, on trouve la traduction en chinois *高速列车* pour *TGV* et *train à grande vitesse*.

La quantité de lexies contenues dans le résultat augmente suivant les niveaux de traduction, du premier vers le troisième. C'est-à-dire qu'on a :

$$\{traductions\_1er\_niveau\} \subseteq \{traductions\_2e\_niveau\} \subseteq \{traductions\_3e\_niveau\}$$

b. *Trois niveaux d'affichage sur l'interface*

Pour faciliter la lecture, nous avons décidé :

(1) d'afficher l'étiquette, la langue et le mot-vedette dans le 1er et le 2ème niveau sur l'interface Web.
(2) d'afficher tous les détails (phrases exemples, définitions, POS, etc.) dans le 3ème niveau, y compris les lexies du même prolexème de la langue source.

(3) de ne pas afficher la traduction dans le 2ème niveau si elle a déjà été trouvée et est déjà affichée dans le 1er niveau.

III.3.3 Un exemple complet de gestion des acronymes

III.3.3.1 Exemple en quatre langues pour le sens "Organisation des Nations Unies"

III.3.3.1.1 Choix de l'exemple

Cette section présente notre méthode avec un exemple en quatre langues, pour le sens *Organisation des Nations Unies*.

(1) En français, il y a *Organisation des Nations Unies*, on peut aussi dire *Nations unies*, *ONU* ou *onusien*[1].
(2) En anglais, on a *United Nations* et son acronyme *UN*.
(3) En chinois, on a *联合国* (lián hé guó) qui est la seule lexie pour ce sens, et il n'y a pas d'acronyme.
(4) En japonais, on a *国際連合* (kokusai-rengō) et son acronyme *国連* (kokuren).

On choisit cet exemple pour les raisons suivantes :

(1) C'est un cas compliqué.
(2) On a déjà utilisé cet exemple dans les présentations ci-dessus, mais jamais complètement.
(3) Cet exemple (parties en anglais et en français) a été utilisé par M. Tran pour présenter PROLEXBASE.
(4) C'est un besoin initial de L&M.

III.3.3.1.2 Définition des étiquettes

Dans cet exemple, il n'y a pas que des acronymes, mais aussi d'autres types de noms propres, par exemple, alias et dérivés. Voir la table ci-dessous.

Table 8 : Étiquettes utilisées pour l'exemple "Organisation des Nations Unies"

Étiquette	Anglais	Français	Chinois	Japonais
Définition (DEF)	*United Nations*	*Organisation des Nations Unies*	联合国	国際連合
Acronyme (ACRO)	*UN*	*ONU*		国連
Alias (ALIAS)		*Nations Unies*		
Dérivé (DERIV)		*onusien*		

III.3.3.2 Modélisation de cet exemple dans PIVAX-3

III.3.3.2.1 Entrées relatives au "non-situé"

a. *Entrée de type "lexie"*

Nous avons déjà montré des exemples de lexies au III.3.2.1.2a. On gère deux types de lien pour chaque lexie : (1) type axème et (2) type prolexème. Les liens vers les prolexèmes contiennent une valeur non vide pour l'attribut `p:relation-mono`. Cette valeur est donc le degré de situement, c'est l'étiquette portée par le lien.

[1] On a repris les termes de PROLEXBASE.

```
<p:lexie p:id="Acro.fra.Nations_unies.1">
      ……
      <p:entryref type="prolexeme" volume="Acro_fra-prolexeme"
          p:idref="Acro.prolexeme.fra.Organisation_des_nations_unies.1" lang="fra" p:relation-
          mono="ALIAS"/>
      <p:entryref type="axeme" volume="Acro_fra-axeme"
          p:idref="Acro.axeme.fra.Nations_unies.1" lang="fra" p:relation-mono=""/>
      ……
</p:lexie>
```

Figure 57 : Exemple des liens de lexie Nations Unies dans la ressource lexicale[1]

Pour les CDM correspondants, voir la première partie de la Figure 55.

b. *Entrée de type "axème"*

```
<p:axeme p:id="Acro.axeme.fra.Organisation_des_nations_unies.1">
        <p:entryref type="final[2]" volume="Acro_fra"
           p:idref="Acro.fra.Organisation_des_nations_unies.1" lang="fra" p:relation-mono=""/>
        <p:axiref type="axie" volume="Acro_axie" p:idref="Acro.axie.United_Nations.1"
           lang="axie" p:relation-mono=""/>
</p:axeme>
```

```
<cdm-elements>
  <cdm-volume    xpath="/p:volume"/>
  <cdm-entry    xpath="/p:volume/p:axeme"/>
  <cdm-entry-id    xpath="/p:volume/p:axeme/@p:id"/>
  <!-- dml.xsd schema requires cdm-headword, but it is meaningless for axemes -->
  <cdm-headword    xpath="/p:volume/p:axeme/@p:id"/>
  <links>
     <!-- links between French axemes and Axies -->
     <link name="axie" xpath="/p:volume/p:axeme/p:axiref">
          <value xpath="@p:idref" />
          <lang xpath="@lang" />
          <volume xpath="@volume" />
          <type xpath="@type" />
     </link>
     <!-- links between French axemes and French lexies -->
     <link name="final" xpath="/p:volume/p:axeme/p:entryref">
          <value xpath="@p:idref" />
          <lang xpath="@lang" />
          <volume xpath="@volume" />
          <type xpath="@type" />
     </link>
  </links>
</cdm-elements>
```

Figure 58 : Exemple d'axème et ses CDM

c. *Entrée de type "axie"*

Les axèmes et les axies ont pour but de présenter le sens exact. C'est pour distinguer les acronymes de types différents entre *onu*, *un* et *国連*. Voir la Figure 59.

[1] On a gardé l'attribut `p:relation-mono` pour le lien de type axème dans cet exemple. C'est un résultat de création initiale de la ressource par un script. Sa valeur est toujours vide, on pourrait donc l'enlever.

[2] Le type `final` est l'implémentation informatique du type `lexie`.

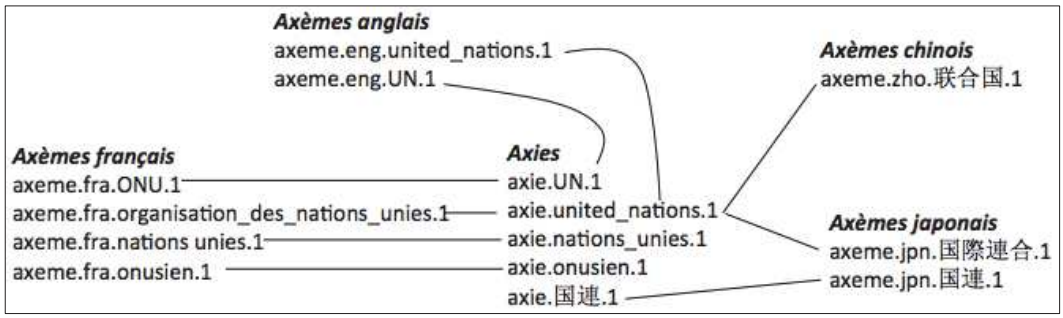


Figure 59 : Liens entre les axèmes et les axies

```
<p:axie id="Acro.axie.UN.1">
  <p:item-links link_group="g1">
    <p:item relation="" volume="Acro_eng-axeme" type="axeme"
      p:idref="Acro.axeme.eng.UN.1" lang="eng"/>
    <p:item relation="" volume="Acro_fra-axeme" type="axeme" p:idref="Acro.axeme.fra.ONU.1"
      lang="fra"/>
  </p:item-links>
</p:axie>
```

```
<cdm-elements>
    <cdm-volume     xpath="/p:volume"/>
    <cdm-entry     xpath="/p:volume/p:axie"/>
    <cdm-entry-id     xpath="/p:volume/p:axie/@id"/>
    <!-- dml.xsd schema requires cdm-headword, but it is meaningless for axies -->
    <cdm-headword     xpath="/p:volume/p:axie/@id"/>
    <!-- id of an axeme linked to the axie -->
    <links>
       <link name="axeme" xpath="/p:volume/p:axie/p:item-links/p:item">
                <value xpath="@p:idref" />
                <lang xpath="@lang" />
                <volume xpath="@volume" />
                <type xpath="@type" />
        </link>
    </links>
</cdm-elements>
```

Figure 60 : Exemple d'axie et ses CDM

III.3.3.2.2 Entrées relatives au "situé"

a. *Entrée de type "prolexème"*

Les liens entre les prolexèmes et les proaxies sont illustrés par la Figure 61.

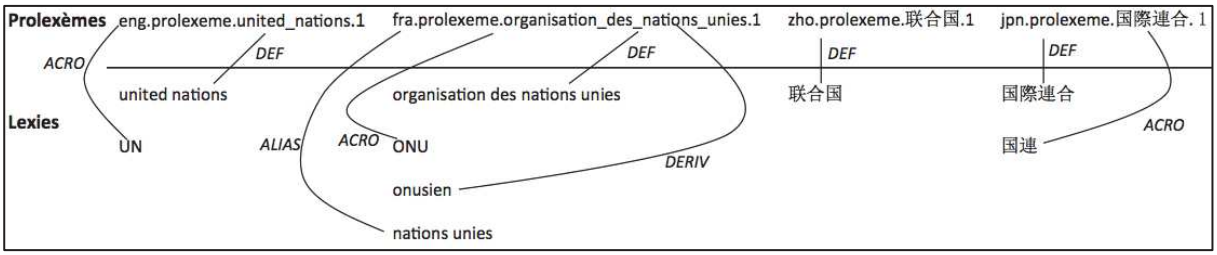


Figure 61 : Liens entre les prolexèmes et les proaxies

Nous avons également utilisé l'attribut `p:relation-mono` pour présenter l'étiquette portée par les liens des prolexèmes vers les lexies.

```
<p:prolexeme p:id="Acro.prolexeme.fra.Organisation_des_nations_unies.1">
    <p:entryref type="final" volume="Acro_fra"  p:idref="Acro.fra.ONU.1" lang="fra"
      p:relation-mono="ACRO"/>
    <p:entryref type="final" volume="Acro_fra"  p:idref="Acro.fra.Nations_unies.1" lang="fra"
      p:relation-mono="ALIAS"/>
```

```
    <p:entryref type="final" volume="Acro_fra"  p:idref="Acro.fra.onusien.1" lang="fra"
       p:relation-mono="DERIV"/>
    <p:entryref type="final" volume="Acro_fra"
       p:idref="Acro.fra.Organisation_des_nations_unies.1" lang="fra" p:relation-mono="DEF"/>
    <p:axiref type="proaxie" volume="Acro_proaxie" p:idref="Acro.proaxie.United_Nations"
       lang="proaxie" p:relation-mono=""/>
</p:prolexeme>
```

```
<cdm-elements>
    <cdm-volume    xpath="/p:volume"/>
    <cdm-entry    xpath="/p:volume/p:prolexeme"/>
    <cdm-entry-id    xpath="/p:volume/p:prolexeme/@p:id"/>
    <cdm-headword    xpath="/p:volume/p:prolexeme/@p:id"/>
    <!-- dml.xsd schema requires cdm-headword, but it is meaningless for prolexeme -->
    <links>
    <!-- links between French prolexemes and Axies -->
       <link name="proaxie" xpath="/p:volume/p:prolexeme/p:axiref">
                <value xpath="@p:idref" />
                <lang xpath="@lang" />
                <volume xpath="@volume" />
                <label xpath="@p:relation-mono" />
                <type xpath="@type" />
                </link>
    <!-- links between French prolexemes and French lexies -->
       <link name="final" xpath="/p:volume/p:prolexeme/p:entryref">
                <value xpath="@p:idref" />
                <lang xpath="@lang" />
                <volume xpath="@volume" />
                <type xpath="@type" />
       </link>
    </links>
</cdm-elements>
```

Figure 62 : Exemple de prolexème et ses CDM

b. *Entrée de type "proaxie"*

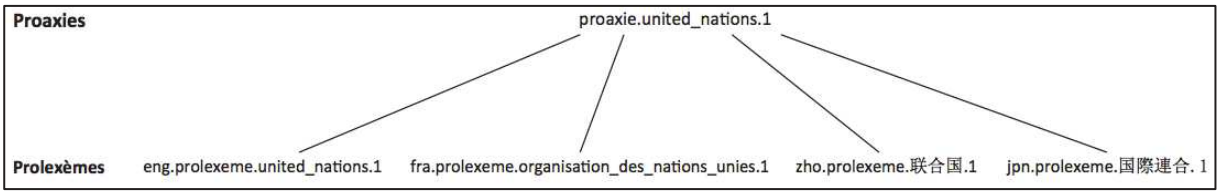


Figure 63 : Liens entre les proaxies et les prolexèmes

Voir la Figure 53 pour l'exemple de l'entrée de proaxie.

```
  <cdm-elements>
    <cdm-volume    xpath="/p:volume"/>
    <cdm-entry    xpath="/p:volume/p:proaxie"/>
    <cdm-entry-id    xpath="/p:volume/p:proaxie/@id"/>
    <!-- dml.xsd schema requires cdm-headword, but it is meaningless for proaxies -->
    <cdm-headword    xpath="/p:volume/p:proaxie/@id"/>
    <!-- id of an axeme linked to the axie -->
    <links>
       <link name="prolexeme" xpath="/p:volume/p:proaxie/p:item-links/p:item">
                <value xpath="@p:idref" />
                <lang xpath="@lang" />
                <volume xpath="@volume" />
                <type xpath="@type" />
       </link>
     </links>
  </cdm-elements>
```

Figure 64 : CDM correspondants des entrées proaxies

III.3.3.2.3 Diagramme de modélisation globale

Dans ce diagramme complet, pour faciliter la lecture, on a concentré la modélisation des types différents des entrées et leurs liens. On ne présente qu'un seul volume, mais un exemple avec plusieurs volumes a déjà été présenté dans la Figure 47.

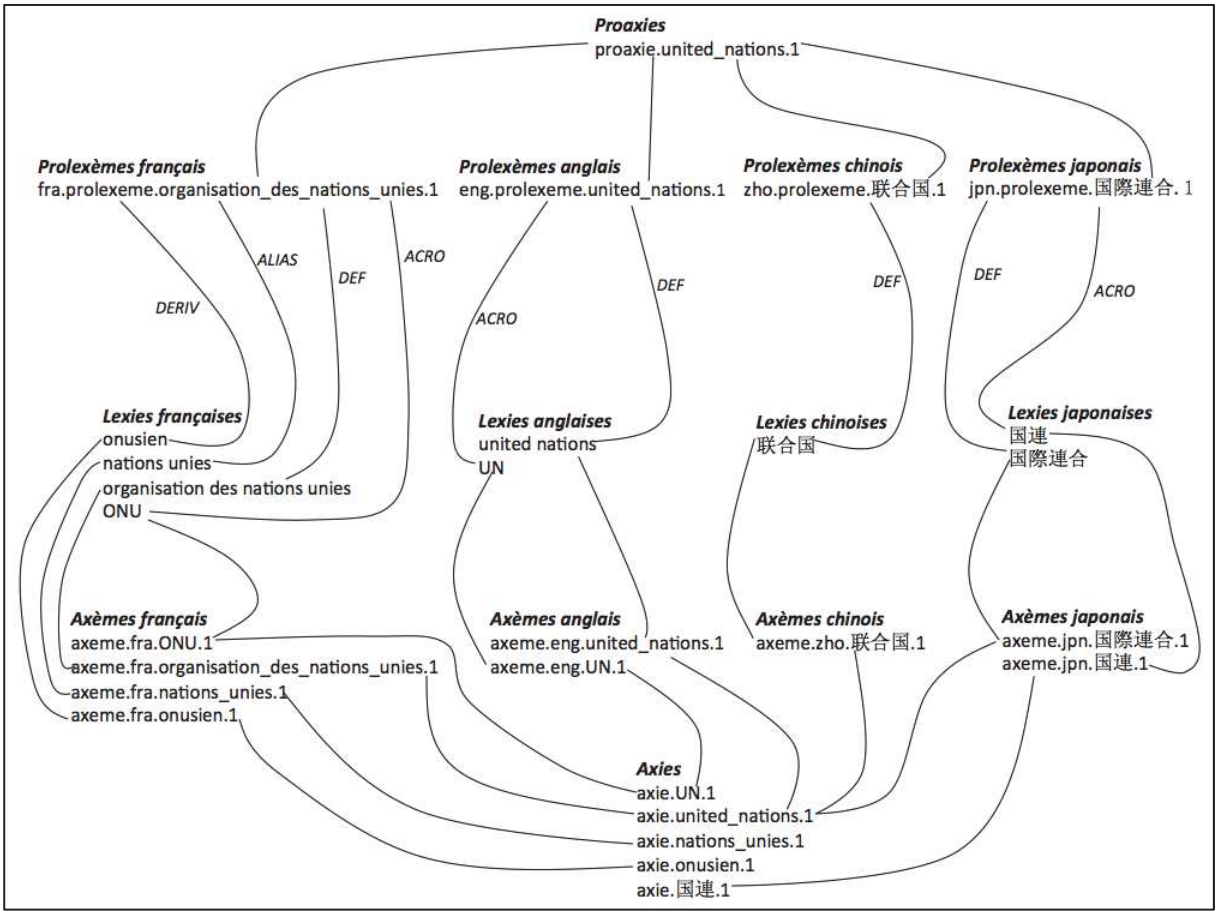


Figure 65 : Modélisation complète de l'exemple organisation des nations unies dans PIVAX-3

III.3.3.2.4 Identification des traductions à trois niveaux, théoriques et affichés

Nous avons expliqué les trois niveaux de précision d'une traduction au III.3.2.2.3.

Lorsqu'on cherche les liens de la lexie *ONU* du français vers l'anglais, vers le japonais et vers le chinois, on a trois niveaux théoriques :

- Le premier niveau de traduction : vers l'anglais, *ONU→UN*.
- Le deuxième niveau de traduction : vers le japonais, *ONU→国連*. Le système trouve un lien dans le volume des prolexèmes français avec l'étiquette `Acro`. Puis il trouve le lien dans les proaxies, ensuite il suit le lien de prolexème en japonais, enfin il arrive au volume des lexies japonaises, et il trouve une lexie avec l'étiquette `Acro`. Donc la lexie proposée du deuxième niveau de langue cible est cet acronyme. Le deuxième niveau de traduction comprend toujours le premier niveau de traduction. C'est-à-dire que *ONU* et *UN* ont la même étiquette `Acro`, donc le lien *ONU→UN* correspond également au deuxième niveau de traduction.
- Vers le chinois : *ONU→联合国*. Le système trouve les lexies par prolexème et proaxie sans étiquette correspondante. Ces lexies proposées constituent le troisième niveau, le moins précis. Le troisième niveau de traduction comprend les niveaux précédents.

Sur l'interface, selon les stratégies présentées au III.3.2.2.3b, on affiche :

- Le premier niveau : terme *UN* de l'anglais avec l'étiquette `Acro`.
- Le deuxième niveau : terme *国連* du japonais avec l'étiquette `Acro`.
- Le troisième niveau : tous les termes, y compris les termes reliés par un même prolexème de la langue source (*ONU, nations unies, onusien et organisation des nations unies*).

III.3.3.3 **Démonstrations**

III.3.3.3.1 Démo 1 : consultation du terme UN de l'anglais vers les autres langues

Il s'agit d'une consultation pour les trois niveaux de traduction.

Table 9 : Trois niveaux de traduction : terme UN de l'anglais vers toutes les langues

Niveau	Lexies trouvables en théorie				Lexies trouvées et affichées par l'interface			
	Français	Anglais	Chinois	Japonais	Français	Anglais	Chinois	Japonais
1	*ONU*				*ONU*			
2	*ONU*			*国連*				*国連*
3	*ONU, Nations unies, onusien, Organisation des nations unies*	*Q='UN'*	*联合国*	*国際連合, 国連*	*ONU, Nations unies, onusien, Organisation des nations unies*	*UN, United Nations*	*联合国*	*国際連合, 国連*

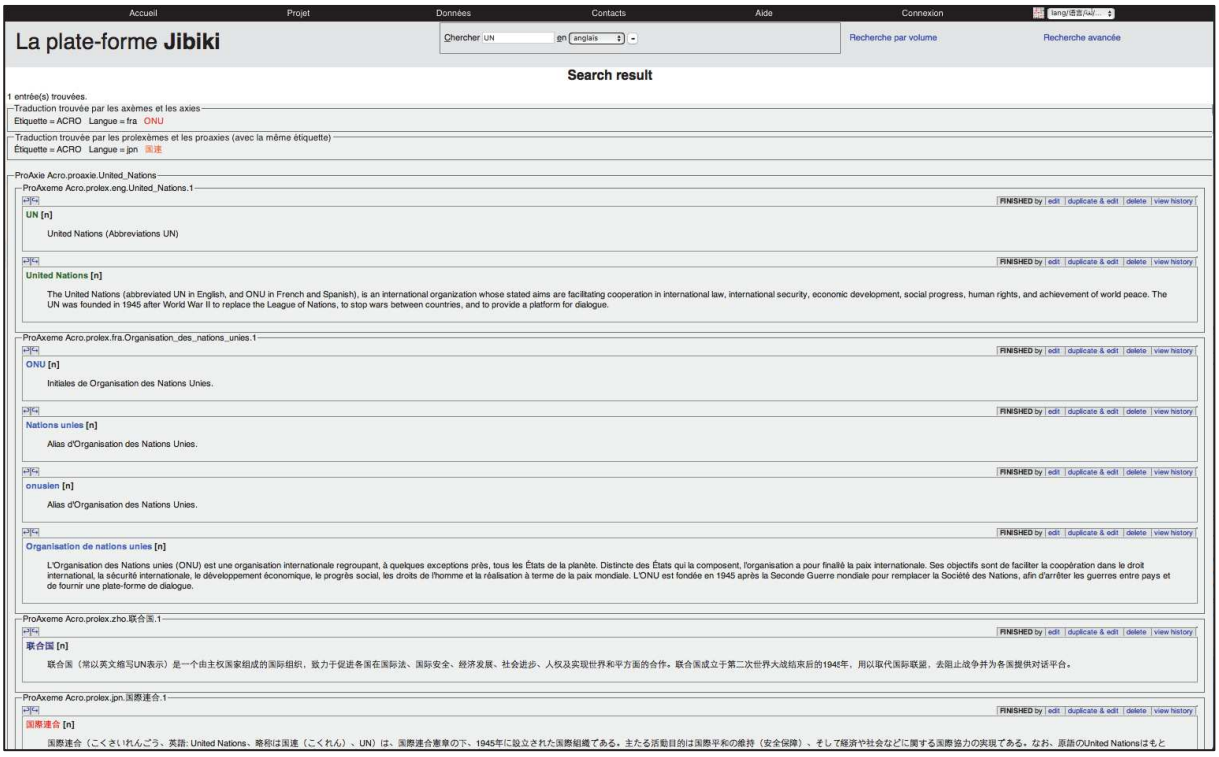


Figure 66 : Terme UN de l'anglais vers toutes les langues

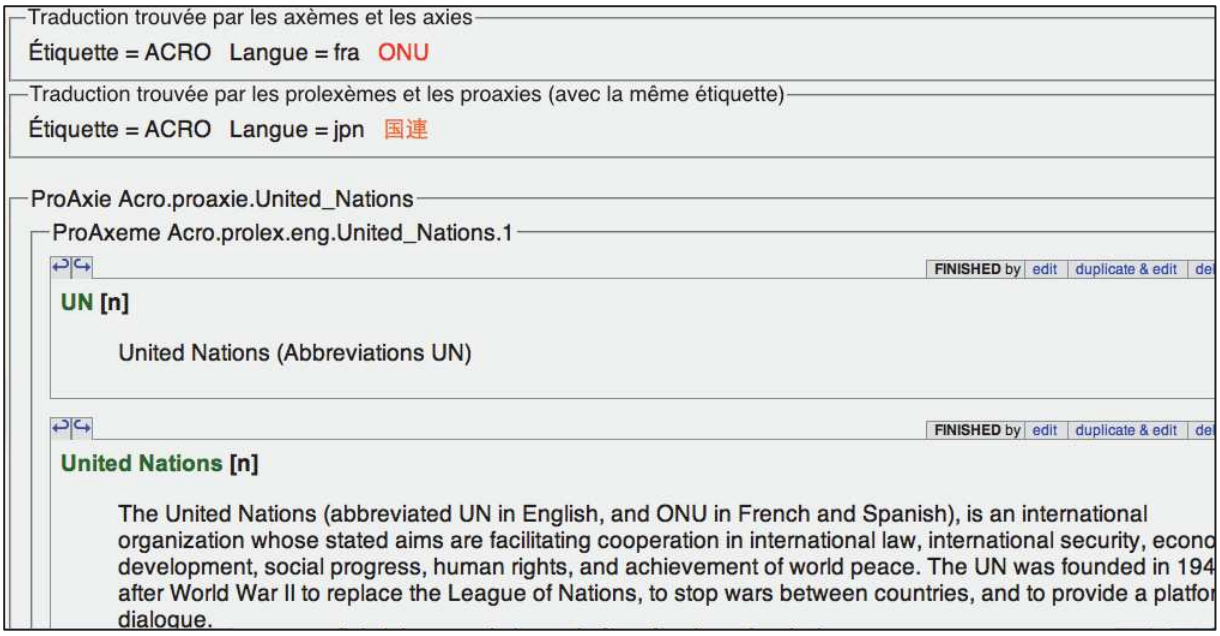


Figure 67 : Affichage agrandi de l'exemple "UN"

III.3.3.3.2 Démo 2 : consultation du terme 国連 du japonais vers les autres langues

Il s'agit d'une consultation aux 2ème et 3ème niveaux de traduction.

Table 10 : Trois niveaux de traduction : terme 国連 (kokuren) du japonais vers toutes les langues

Niveau	Lexies trouvables en théorie				Lexies trouvées et affichées par l'interface			
	Français	Anglais	Chinois	Japonais	Français	Anglais	Chinois	Japonais
1								
2	*ONU*	*UN*			*ONU*	*UN*		
3	*ONU, Nations unies, onusien, Organisation des nations unies*	*UN, United Nations*	*联合国*	*Q='国連'*	*ONU, Nations unies, onusien, Organisation des nations unies*	*UN, United Nations*	*联合国*	*国際連合, 国連*

Figure 68 : Terme 国連 du japonais vers toutes les langues

III.3.3.3.3 Démo 3 : consultation du terme onusien du japonais vers les autres langues

Il s'agit d'une consultation au 3ème niveau de traduction.

Table 11 : Trois niveaux de traduction : terme onusien du français vers toutes les langues

Niveau	Lexies trouvables en théorie				Lexies trouvées et affichées par l'interface			
	Français	Anglais	Chinois	Japonais	Français	Anglais	Chinois	Japonais
1								
2								
3	*Q= 'onusien'*	*UN, United Nations*	*联合国*	*国際連合, 国連*	*ONU, Nations unies, onusien, Organisation des nations unies*	*UN, United Nations*	*联合国*	*国際連合, 国連*

Figure 69 : Terme onusien du français vers toutes les langues

III.4 Autres extensions envisageables

PIVAX-3 est le premier prototype, la base actuelle est une preuve de concept qui comporte quelques exemples issus de PROLEXBASE et une toute petite partie des données protégées de Lingua et Machina. Nous souhaitons tester cette solution en passant à l'échelle sur de grosses bases telles que CJK (chinois, japonais, coréen, arabe, anglais) avec 24 millions d'entrées ou L'UNIFED MEDICAL LANGUAGE SYSTEM avec 5 millions de termes.

III.4.1 Vers l'intégration général-terminologique-situé

Dans l'avenir, nous prévoyons de l'intégrer pour l'utilisation dans un domaine linguistique spécifique, et de l'enrichir en utilisant les quatre variations du diasystème de Coseriu.

III.4.1.1 Intégration pour l'utilisation dans un domaine linguistique spécifique

Pour le futur, nous souhaitons faire évoluer cette macrostructure pour prendre en compte les différents sous-types de synonymie, et transposer le concept de prolexème pour que cette solution puisse être utilisée dans un autre domaine linguistique.

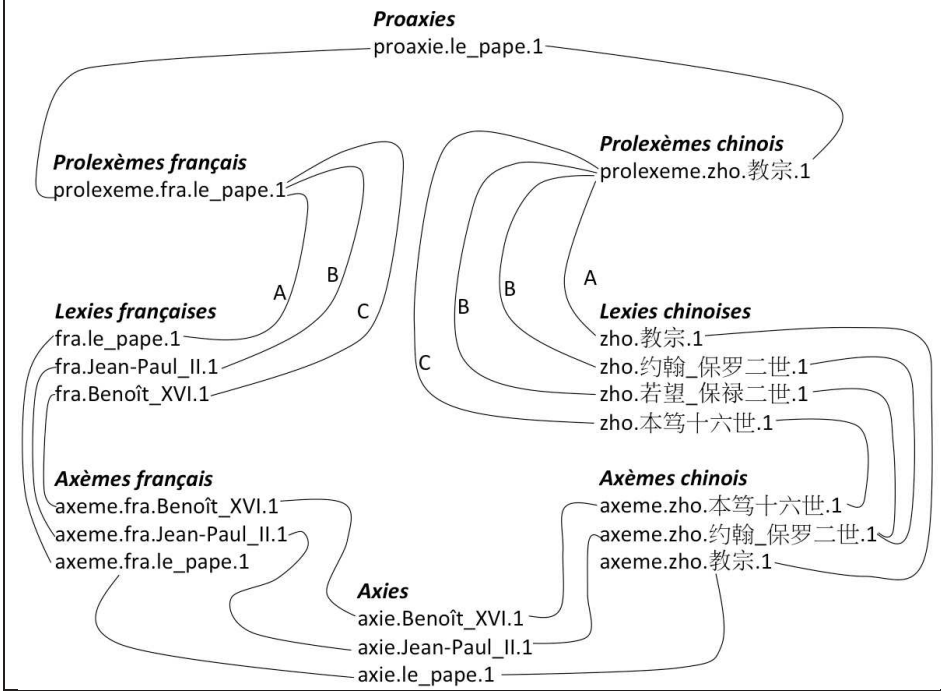
Par exemple, pour une ressource lexicale comprenant des textos, en français *A+* correspondrait à *À plus* ou *À plus tard* avec une étiquette `texto`, et en anglais *L8R* correspondrait à *later* avec l'étiquette `texto`.

III.4.1.2 Intégration pour l'utilisation des quatre dimensions du diasystème

Comme notre étiquette est une étiquette libre, pour chaque BDLex de type PIVAX-3, il faut bien définir les étiquettes utilisées pour conserver la cohérence.

Nous prévoyons de prendre en compte également les quatre dimensions du diasystème basé essentiellement sur ce qu'Eugenio Coseriu a proposé : diachronique (variété dans le temps), diaphasique (variété concernant les finalités de l'emploi), diatopique (variété dans l'espace), et diastratique (variété relative à la stratification socio-culturelle).

Pour ces cas complexes, on a proposé d'utiliser le descripteur de situement (voir la Table 1) comme étiquette. Voici un exemple.



Descripteur de situement (étiquettes)	
A	le_pape#général
B	le_pape#situé? date=16101978-02042005
C	le_pape#situé? date=19042005-28022013

Figure 70 : Modélisation de l'exemple "le pape" dans PIVAX-3

Les deux lexies chinoises *zho.约翰_保罗二世.1* et *zho.若望_保禄二世.1* sont synonymes. On peut les échanger dans tous les contextes. Donc, on a ces deux traductions au premier niveau de traduction pour *fra.Jean-Paul_II.1*.

III.4.2 Autres structures (ex: INNOVALANGUES-LEXINNOVA)

Mes co-directeurs de thèse ont proposé d'utiliser PIVAX-3 dans le sous-projet LEXINNOVA du projet INNOVALANGUES, dans lequel elle et il sont impliqués. Nous n'avons pas pu y travailler vraiment, mais pensons qu'il est intéressant de donner ici leur perspective d'utilisation et d'extension de PIVAX-3 dans ce projet.

III.4.2.1 Contexte du projet INNOVALANGUES-LEXINNOVA

Citons ici la brève présentation de LEXINNOVA dans sa page wiki interne.

Dans le cadre de l'ENPA (Environnement Numérique Personnalisé d'Apprentissage) d'Innovalangues[1], nous souhaiterions créer des outils sous licence ouverte (données et code), qui permettraient aux apprenants identifiés travaillant sur tablette ou ordinateur de créer leurs propres lexiques dynamiques multilingues d'apprentissage et de générer des exercices associés.

Les étudiants, tuteurs et enseignants pourraient également avoir accès aux corpus de groupes-classes donnés.

[1] INNOVALANGUES est un projet IDEFI-ANR visant à innover dans l'apprentissage et l'enseignement des langues. Il a été lancé officiellement le 14 juin 2012.

Ces corpus pourraient être issus de diverses sources : entrées des étudiants, corpus de documents de cours, d'exercices et jeux faits sur l'ENPA par l'étudiant. Chaque lexique individuel ou collectif résulterait d'une extraction de données provenant d'une base lexicale partagée.

Les lexiques individuels devraient pouvoir être récupérés par les étudiants à la fin de leur formation (export EXCEL ou XML).

La mémoire du parcours et des contextes associés pour chaque utilisateur (élève, enseignant, et même logiciel) est partagée avec les autres outils d'Innova. A chaque session, selon le profil de l'utilisateur et des informations propres à notre application, les données sont extraites des bases lexicales et corporale pour constituer la vue temporelle exploitée. Les actions de l'utilisateur conduisant à des modifications de ces bases sont prises en compte immédiatement lorsque c'est possible, sinon différées au moment où la connexion sera rétablie. Les fonctionnalités prévues ne conduisent qu'à des actions introduisant des nouvelles informations, il n'y a donc pas de conflits a posteriori.

III.4.2.2 **Proto-structure du dictionnaire**

Il existe déjà des outils dictionnairiques (MAGIC WORD[1], GAME OF WORDS[2], CHECK YOUR SMILE[3], KINEPHONES[4], SELF[5]) utilisés dans le cadre du projet INNOVALANGUES. Chaque outil a sa propre BDLex. On voudrait intégrer et unifier ces ressources dans une même structure de BDLex.

La proto-structure du dictionnaire est créée à partir de celles des outils existants et à partir des besoins pédagogiques du projet. Cela nous a menés à la microstructure suivante.

- forme fléchie (MAGIC WORD)
- lemme (GAME OF WORDS, MAGIC WORD)
- catégorie grammaticale (CHECK YOUR SMILE), ex. nom, verbe, etc.
- catégorie morphologique (MAGIC WORD), ex. classe du Bescherelle.
- définition(s) L2 (MAGIC WORD, CHECK YOUR SMILE)
- définition(s) L1 (CHECK YOUR SMILE)
- entrées liées sémantiquement (GAME OF WORDS)
- découpage syllabique (CHECK YOUR SMILE)
- syllabe accentuée (CHECK YOUR SMILE)
- transcription phonétique L2 (CHECK YOUR SMILE, KINEPHONES)
- transcription phonétique L1 (CHECK YOUR SMILE)
- transcription en couleurs (KINEPHONES)
- enregistrement sonore (CHECK YOUR SMILE)
- mot en contexte, par exemple collocations (SELF)
- association entre mot et niveau de compétence d'usage en production, par exemple le "profil en anglais" ou "english profile" (SELF)

---

[1] MAGIC WORD : vise à produire un premier prototype de jeu inspiré du Boggle. Voir http://gamer.innovalangues.net/magicword/

[2] GAME OF WORDS : permet de développer un jeu à partir des règles du Taboo. Voir http://gamer.innovalangues.net/gameofwords/

[3] CHECK YOUR SMILE : ce chantier vise à créer un site web collaboratif et ludo-éducatif pour l'apprentissage d'un lexique spécialisé.

[4] KINEPHONES : réalise un prototype pour l'enseignement-apprentissage de la phonologie du français et de l'anglais (British et US). Voir kinephones.u-grenoble3.fr.

[5] SELF : Système d'Evaluation en Langues à visée Formative. Voir http://self.innovalangues.net.

- sinogrammes
- composants des sinogrammes
- transcriptions des sinogrammes en mandarin et en japonais
- image

III.4.2.3 **Modélisation de la macrostructure avec des exemples**

On a modélisé la proto-structure aux niveaux forme, lemme, lexie, axème et axie. Voir la Figure 71. Dans cette figure, on n'a pas dessiné tous les liens. Il faudrait ajouter un axème correspondant à chaque lexie.

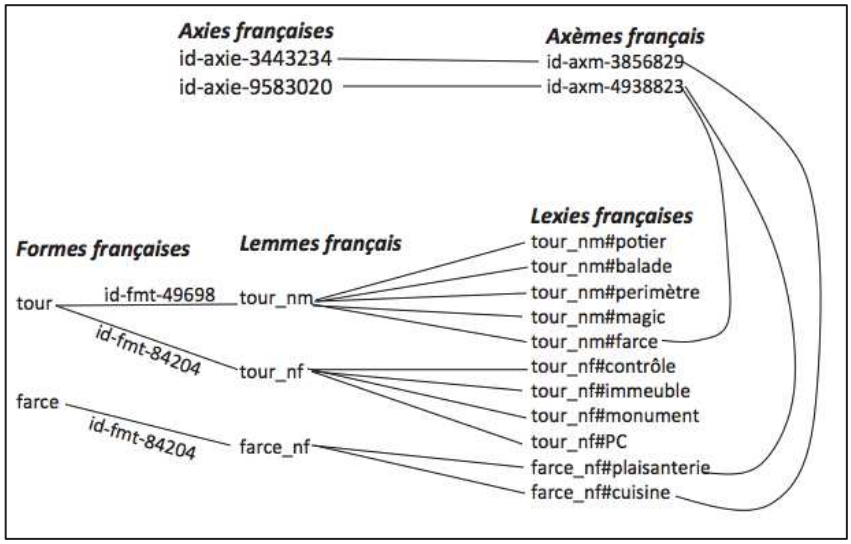


Figure 71 : Modélisation de BDLex pour LEXINNOVA

Pour chaque langue naturelle, on aura une combinaison des informations morphologiques qui permettent de générer un lemme (ce qui vient de l'analyse morphologique de la forme, éventuellement précisé par des interactions humaines). Dans notre exemple, Ch. Boitet a proposé d'utiliser le résultat d'un analyseur morphologique écrit en ATEF[1] comme "descripteurs morphologiques". Les liens entre forme et lemme portent des étiquettes commençant par `id-fmt`.

Table 12 : Exemple de descripteur morphologique

id-fmt	**valeur**
`id-fmt-49698`	`GNR=MAS; CAT=N; SUBN=NC; NBR=SIN`
`id-fmt-84204`	`GNR=FEM; CAT=N; SUBN=NC; NBR=SIN`

[1] On a déjà mentionné ARIANE-G5 au I.1.1.1. ATEF (Analyse de Textes en États Finis) est le LSPL (Langage Spécialisés pour la Programmation Linguistique) utilisé par ARIANE-G5 pour écrire tous les analyseurs morphologiques. Ce langage a été créé en 1972 par J. Chauché, P. Guillaume et M. Quézel-Ambrunaz.

Chapitre IV Outils génériques pour BDLex "actives"

Introduction

Un point important pour faire progresser le domaine des BDLex et de la lexicographie computationnelle en général est de les faire fonctionner de façon fiable et performante comme des serveurs Web, qui peuvent s'appeler mutuellement (comme PIVAX et SECTRA), et générer de grandes quantités d'appels. Par exemple, SECTRA peut demander à PIVAX de recalculer en tâche de fond et itérativement un "mini-dictionnaire" pour chaque segment d'une ou plusieurs mémoires de traductions. Cela demande la lemmatisation de tous les mots de ces segments. Mais cette fonction de lemmatisation peut elle-même être appelée par bien d'autres services, par exemple par un service de "lecture active" comme celui récemment réalisé par M. Mangeot en utilisant la BDLex japonais-français qu'il a construite à partir du dictionnaire de CESSELIN (85 000 entrées) et de la partie japonais-français du dictionnaire de Jim Breen (~15 000 entrées venant du sous-projet du projet PAPILLON de Jean-Marc Desperrier).

Dans ce chapitre, nous présentons d'abord la solution retenue (ACTIVEMQ) pour réaliser la gestion des tâches et des requêtes entre des serveurs lexicaux et d'autres serveurs (comme des serveurs de lemmatisation via LEXTOH, de traduction automatique via TRADOH, de gestion de corpus et mémoires de traductions comme SECTRA, et de lecture active ou de présentation bilingue).

Nous présentons ensuite l'intergiciel LEXTOH qui permet d'appeler un ou plusieurs services de lemmatisation, puis d'unifier et de filtrer leurs résultats, en utilisant le langage TRACOMPL du système ARIANE-G5.

Enfin, nous décrivons le logiciel CREATDICO qui permet de construire des "mini-dictionnaires" de formes et de formats variés, typiquement à partir d'un segment (phrase ou titre) ou d'un paragraphe.

IV.1 Gestion des travaux par ACTIVEMQ

IV.1.1 Motivations : besoins attestés et fonctionnalités désirées

IV.1.1.1 Besoins attestés

IV.1.1.1.1 Besoins de SECTRA et de TRADOH

SECTRA_W (Système d'Expoitation de Corpus de Traductions sur le Web) était disponible avant mon arrivée au laboratoire. Ce système a été développé par C.-P. Huynh durant sa thèse [Huynh, C.-P., 2010] et amélioré par H.-T. Nguyen [Nguyen, H.-T., 2009], L. X. Wang [Wang, L. X., 2015] et Ch. Boitet. Il est utilisé pour l'évaluation de systèmes de TA et pour la postédition collaborative de pages Web prétraduites par un ou plusieurs systèmes de TA.

Les prétraductions sont obtenues via des appels à TRADOH [Vo-Trung, H., 2004]. TRADOH est un intergiciel d'appel à un ou plusieurs services de traduction automatique. SECTRA envoie souvent une suite de requêtes à TRADOH. Il n'y a pas de gestion de files d'attente entre les requêtes de SECTRA et les réponses de TRADOH, ce qui provoque parfois un problème de décalage entre segments source et segments cible. Voir la Figure 72.

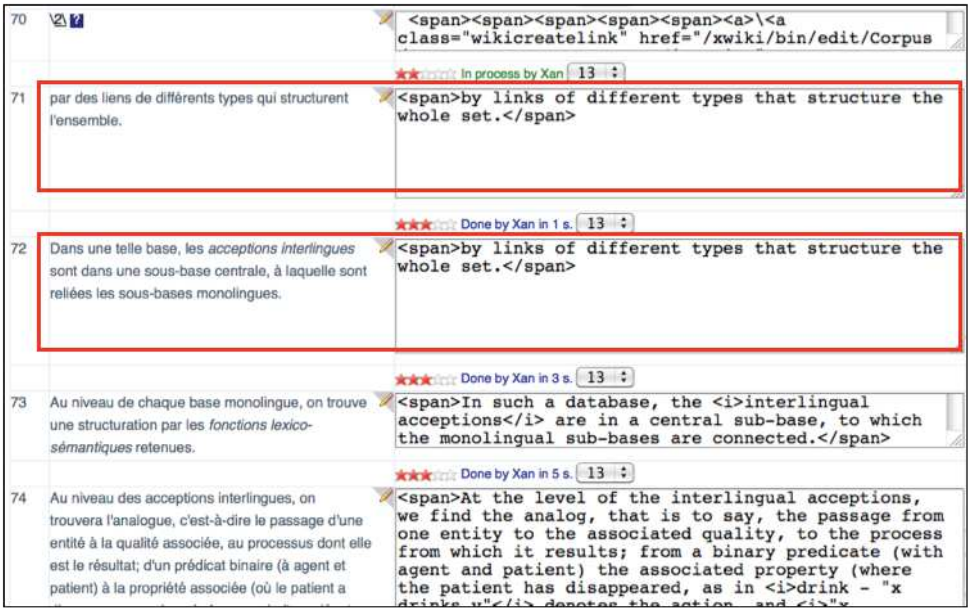


Figure 72 : Décalage de prétraductions dans SECTRA après plusieurs appels consécutifs à TRADOH

IV.1.1.1.2 Besoins de PIVAX pour les mini-dictionnaires

H.-T. Nguyen a essayé vers la fin de sa thèse de créer un mini-dictionnaire pour le projet et pour chaque segment dans SECTRA par des appels à JIBIKI-1/PIVAX-1. Cet effort n'a pas réussi à cause des mauvaises performances de JIBIKI-1/PIVAX-1 (voir I.1.4.3, I.1.4.3 et II.2.4) et du manque d'outil de gestion de files d'attente. Avec JIBIKI-2/PIVAX-3, même si les performances sont meilleures qu'avec JIBIKI-1/PIVAX-1, on ne peut toujours pas garantir un délai de réponse.

IV.1.1.1.3 Besoins de LEXTOH et CREATDICO

Les nouveaux intergiciels LEXTOH et CREATDICO ont les mêmes besoins. LEXTOH est un intergiciel de lemmatisation. CREATDICO est un intergiciel de création de dictionnaires relatifs à un ensemble de mots. Ils seront présentés en détail dans les sections suivantes.

Un exemple d'utilisation simple de LEXTOH par SECTRA est une demande d'analyse morphologique de chaque segment pour annoter les deux parties d'un corpus bilingue et préparer l'apprentissage d'un système MOSES de type "factoriel".

Un scénario d'utilisation de CREATDICO est la création de mini-dictionnaires pour SECTRA. SECTRA peut envoyer 1000 requêtes en même temps à CREATDICO. Chaque requête concerne la demande d'un mini-dictionnaire pour un segment. Pour traiter une requête, CREATDICO envoie le segment à LEXTOH pour demander son analyse morphologique. Après avoir reçu les lemmes, CREATDICO envoie la requête à PIVAX. Même si on peut consulter plusieurs lemmes dans une requête en utilisant l'API de JIBIKI[1], le délai de réponse est parfois dépassé.

IV.1.1.1.4 Besoins d'autres projets

Il y a également d'autres outils ou projets de notre équipe, comme SEGDOC [Kalitvianski, R., 2013], OMNIA [Rouquet, D. & Nguyen, H. T., 2009 ; Rouquet, D. et al., 2013], ITOLDU [Bellynck, V. et al., 2005] qui ont besoin d'un outil de gestion de flots de travaux.

[1] On peut demander la consultation de plusieurs lemmes dans une requête. C'est la nouvelle fonction de l'API de JIBIKI-2, réalisé par M. Mangeot en 2015. Voir http://jibiki.fr/jibiki/Api.po.

IV.1.1.2 **Fonctionnalités désirées**

IV.1.1.2.1 Gestion de flots de travaux (files d'attente et priorités)

Pour permettre de traiter des "flots de travaux" envoyés de n'importe où, il ne faut pas seulement des files d'attente, mais aussi des priorités. Une requête de consultation par un humain doit être traitée le plus vite possible.

IV.1.1.2.2 Possibilité de gagner en sécurité et d'assurer l'intégrité des données

Nous avons aussi la nécessité de sécurité et d'intégrité. Il ne faut pas perdre les requêtes après un redémarrage du serveur de type "boîte aux lettres". Il faut garantir que toutes les requêtes seront traitées, renvoyer automatiquement une requête s'il y a un grand délai impossible (on considère alors que la requête est perdue). Il faut aussi pouvoir traiter les requêtes de manière asynchrone.

IV.1.1.2.3 Surveillance des travaux en cours et compatibilité

Il faut pouvoir surveiller toutes les tâches de "boîte aux lettres" par une interface. En plus, comme on a plusieurs outils écrits dans plusieurs langages de programmation, on désire pouvoir traiter une grande variété de clients et de protocoles écrits en PHP, JAVA, C++. La simplicité d'utilisation est également importante.

Comme ce type de système demande un gros développement, nous avons cherché un outil qui pourrait nous convenir.

IV.1.2 Approches envisageables

IV.1.2.1 **Extension de BLEXISMA**

BLEXISMA (Base LEXIcale Sémantique Multi-Agents) est un système d'agents à gros grains, développé par Didier Schwab dans le cadre de sa thèse [Schwab, D., 2005], qui portait sur la désambiguïsation lexicale et les fonctions lexicosémantiques.

Pendant le projet TRAOUIERO (2011-2012), notre équipe avait prévu de développer une extension de BLEXISMA [Schwab, D., 2013]. L'objectif était de passer d'une organisation en clients et serveurs à une organisation en agents à gros grains. Au niveau de la réalisation, le but de cette tâche était de construire un dispositif de contrôle du système complet et d'organiser les modules principaux comme des agents à gros grains. Mais la personne compétente a été prise sur un autre projet et rien n'a été fait.

IV.1.2.2 **Reprise du "réseau CASH/LIDIA"**

ARIANE-G5 (système de développement et d'exploitation d'applications de traduction automatique) fonctionne grâce à un "réseau de machines virtuelles" sur un serveur IBM équipé de zVM/CMS. La réalisation du réseau CASH/LIDIA par Pierre Guillaume utilise comme dispositif d'interconnexion le "Spooling System" de VM et la mise en communication des lecteurs-perforateurs (READER PUNCHER) des machines virtuelles d'un même site VM/SP. Il s'agit donc de communications asynchrones et les problèmes de conflit d'accès sont laissés à VM/SP. La communication entre machines virtuelles se présente comme des échanges de fichiers (fichiers SPOOL) et de gestion de files d'attente (sur des lecteurs virtuels). Il s'est également avéré indispensable d'envisager une gestion de messages permettant une synchronisation rapide de l'état du réseau (machines serveurs ARIANE-G5 invitées à stopper en fin de leur traitement courant par exemple). Ce réseau intègre la (ou les) machines virtuelles gérant le "REMOTE SPOOLING" de façon à pouvoir éventuellement prolonger le réseau vers des sites extérieurs.

Ce système ne nous convenait pas, car nous voulions que tous les "agents" ou simplement "services" soient des services Web.

IV.1.2.3 Utilisation de JOBCENTER[1]

JOBCENTER est une plate-forme de gestion des travaux et d'exécution de travail distribué. Cet outil a été développé par Bio Med, et est en source ouvert.

Il y a deux types de composants dans ce système : client et serveur. Les deux types sont écrits en Java. Toutes les communications avec le serveur de JOBCENTER sont réalisées par un pilote du client. Le client peut envoyer un message à une file d'attente avec sa priorité.

Nos outils devraient appeler d'abord les composants clients de JOBCENTER pour communiquer avec le serveur. Les appels sont faits par le protocole HTTP en utilisant une API REST et XML.

JOBCENTER aurait été une très bonne solution. On ne l'a finalement pas utilisé, parce que son installation est lourde et qu'il y a peu d'exemples d'utilisation. De plus, il faut le déployer sur le serveur et sur chaque client. Cet outil demande également l'installation d'une base de données sur le serveur, dans laquelle les configurations des clients sont enregistrées. Après chaque ajout d'un client, il faut mettre à jour la base de données du serveur.

IV.1.2.4 Utilisation de ACTIVEMQ[2]

ACTIVEMQ est un intergiciel à messages utilisable pour le développement d'une architecture MOM (Middleware Oriented Messages). Cet outil a été créé par la fondation APACHE, et est en source ouvert. Il est écrit en Java en utilisant JMS (Java Message Service). Il supporte beaucoup de langages et de protocoles pour les clients. Il est utilisé par plusieurs grands projets comme APACHE CAMEL[3], JETTY[4], APACHE MIX[5], MULE[6] et également UIMA.

Cet outil nous convient très bien car il couvre tous nos besoins. Il est de plus encapsulé dans un fichier téléchargeable. Après une simple décompression, l'outil est utilisable, sans installation particulière. Le serveur ACTIVEMQ est déployé automatiquement par l'exécution d'une ligne de commande. Il n'y a pas besoin de le déployer sur les clients.

Il y a beaucoup d'exemples et de tutoriels détaillés sur le site officiel d'ACTIVEMQ. Il supporte aussi beaucoup de langages de programmation sur les clients (JAVA, PERL, PHP, PYTON, C, C++, C#, RUBY). De base, il est possible de se connecter en TCP, mais il est aussi possible d'échanger du texte via HTTP (REST ou SOAP). On l'a choisi aussi pour sa bonne intégration avec la plate-forme SPRING[7]. Cela nous a permis de diminuer le temps de programmation.

---

[1] https://github.com/yeastrc/jobcenter

[2] http://activemq.apache.org

[3] http://camel.apache.org

[4] http://www.eclipse.org/jetty/

[5] http://servicemix.apache.org

[6] https://www.mulesoft.com

[7] SPRING est une plate-forme en source ouvert pour construire et définir l'infrastructure d'une application java. Sans SPRING, pour appeler le serveur ACTIVEMQ, on doit programmer une centaine de lignes. Avec SPRING, on peut configurer les appels au serveur ACTIVEMQ par un fichier XML. Voir http://spring.io et http://activemq.apache.org/spring-support.html

IV.1.3 Intégration d'ACTIVEMQ

IV.1.3.1 Étude et analyse d'ACTIVEMQ

IV.1.3.1.1 Mécanismes de traitement de messages

Il y a deux mécanismes de traitement de messages : le mode diffusion/écoute (topic) et le mode message suivi (queue). Les différences sont montrées dans la Table 13.

Table 13 : Comparaison des mécanismes de traitement de messages d'ActiveMQ

Mécanisme	Diffusion/écoute (topic)	Message suivi (queue)
Statut d'enregistrement	Pas d'enregistrement	Le message est enregistré par défaut comme un fichier. On peut également l'enregistrer dans une base de données par une configuration.
Intégrité et sécurité	Si un client n'a pas lancé le programme d'écoute, il ne peut pas recevoir le message. Ce mécanisme ne supporte pas le traitement asynchrone.	Après réception d'un message adressé à un client, le système crée un symbole de traitement pour ce message. Si le programme de réception du destinataire ne s'exécute pas, le message sera enregistré, et sera envoyé quand le programme de réception du destinataire sera activé.
Destination	Un message est consommé une ou plusieurs fois par un ou plusieurs destinataires.	Un message est consommé une seule fois par un seul destinataire.

Après cette analyse, on a choisi le mode message suivi pour nos premières implémentations. Chaque client est créé comme destinataire de type `queue`.

IV.1.3.1.2 Attributs de message et fonctionnalités correspondantes

Chaque message contient une suite d'attributs : `Destination`, `ReplyTo`, `Type`, `DeliveryMode`, `Priority`, `MessageId`, `Timestamp`, `CorrelationID`, `Expiration` et `Redelivered`. Ces attributs nous permettent de spécifier les différents types de requête.

IV.1.3.2 Implémentation

IV.1.3.2.1 Architecture globale

On déploie une seule fois ACTIVEMQ comme serveur de gestion des travaux. Un système ou un outil comme LEXTOH, PIVAX, SECTRA, etc. est défini comme un client.

Chaque client doit fournir deux petites implémentations de ses spécifications : une implémentation de producteur (ou émetteur) et une implémentation de récepteur (ou consommateur, ou écouteur). Les producteurs sont lancés par un programme spécial ou par une commande au terminal pour envoyer les requêtes. Quand les requêtes sont envoyées, le programme du producteur s'arrête tout de suite. Les récepteurs activent leur programme d'écoute en boucle pour recevoir les requêtes. Voir la Figure 73. Chaque message est enregistré comme un fichier sur le serveur.

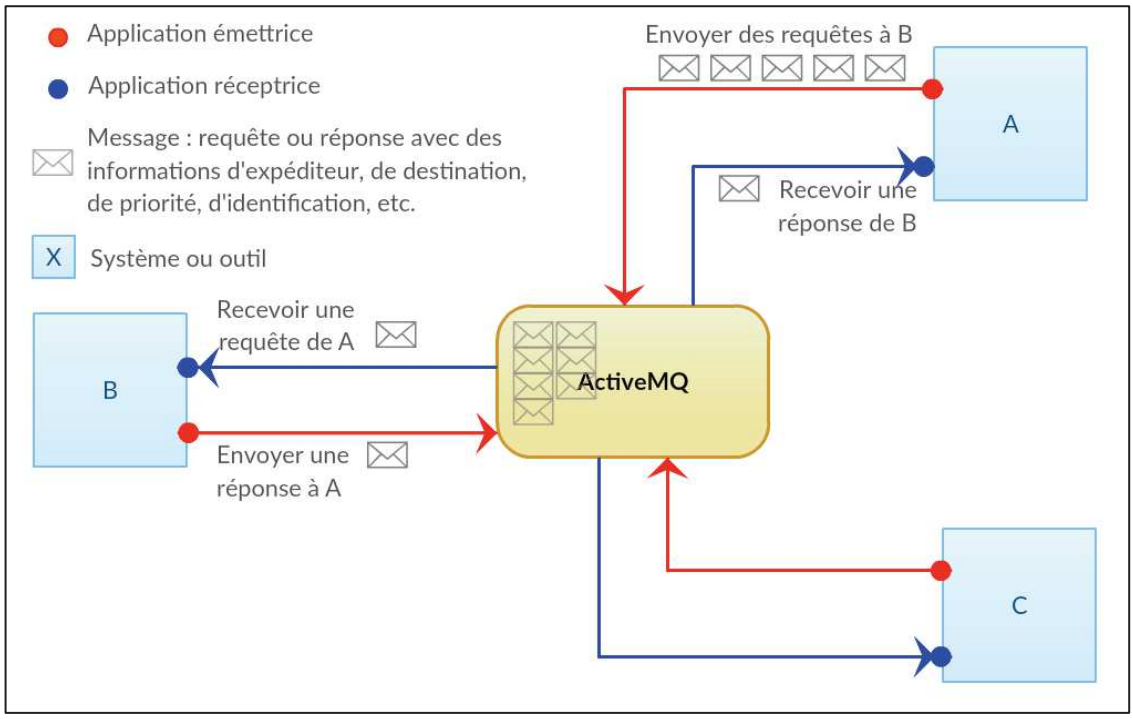


Figure 73 : Architecture "client- serveur" de gestion de travaux en utilisant ACTIVEMQ

IV.1.3.2.2 Interfaces et disponibilité

La Figure 74 montre la page d'accueil de l'interface de contrôle d'ACTIVEMQ. Notre serveur ACTIVEMQ est disponible sur la machine aximag2 (http://46.105.41.94:8161/admin/).

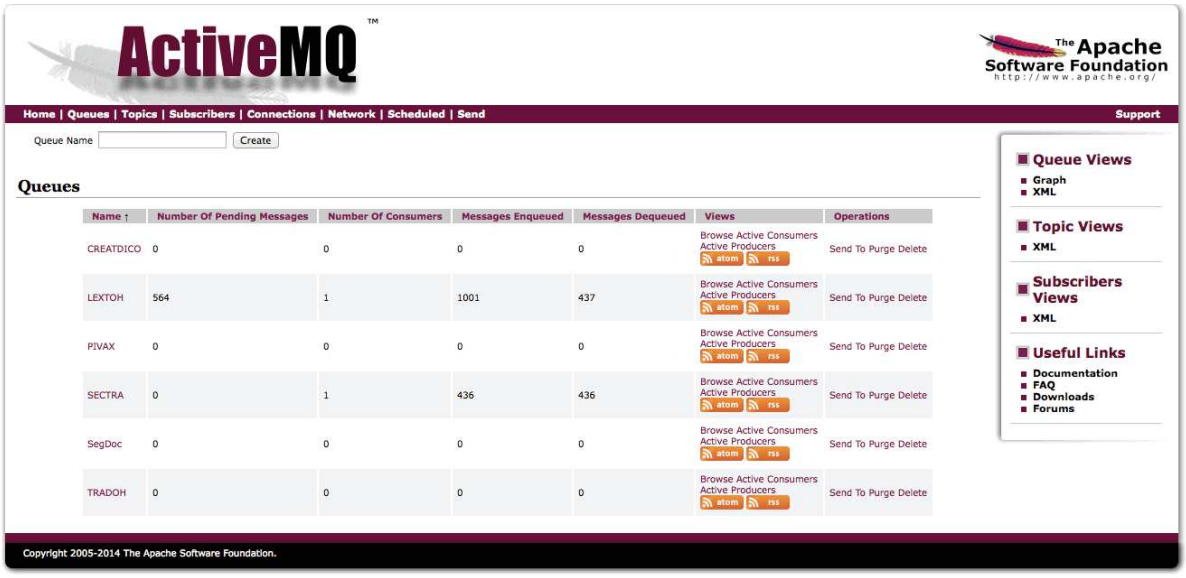


Figure 74 : Page d'accueil de l'interface de contrôle d'ACTIVEMQ

IV.1.3.3 **Expériences et validations**

Pour vérifier ce système, on a fait des expériences sur le problème du décalage des prétraductions par TA dans SECTRA, présenté au IV.1.1 ci-dessus.

La procédure principale est la suivante :

SECTRA→ACTIVEMQ→TRADOH→ACTIVEMQ→SECTRA

SECTRA envoie une suite de requêtes via l'émetteur de SECTRA vers la queue TRADOH, via le serveur ACTIVEMQ. Chaque requête contient une demande de prétraduction d'un segment avec ses paramètres. Ces requêtes sont enregistrées au niveau d'ACTIVEMQ quand elles arrivent. L'écouteur de TRADOH traite ces requêtes, et appelle l'API de TRADOH pour les traiter une par une. Ensuite l'émetteur de TRADOH envoie les réponses à la queue SECTRA sur le serveur ACTIVEMQ. Les prétraductions sont faites correctement et sans décalage.

La Table 14 montre quelques détails sur ces expériences. Les informations supplémentaires, le fichier des requêtes source et le fichier des prétraductions reçues dans l'expérience 1 sont donnés dans l'Annexe 9.

Table 14 : Expériences sur les appels de TRADOH à partir de SECTRA via ACTIVEMQ

Expérience	Nombre de requêtes	Langue source	Langue cible	Système de TA	Longueur moyenne de segment	Temps utilisé
1	100	zho	fra	Google	39.66	00:41
2	500	zho	fra	Google	40.688	03:20
3	1 000	zho	fra	Google	40.657	05:59
4	2 000	zho	fra	Google	40.657	08:38

Remarque :

- L'envoi des requêtes est fait très rapidement (moins d'une seconde dans les expériences 1 et 2, et moins de deux secondes dans les expériences 3 et 4).
- Temps utilisé = Début d'envoi de requête – Dernière prétraduction reçue par SECTra
- Longueur moyenne = longueur moyenne des segments source (chinois) en caractères

IV.2 LEXTOH

LEXTOH est un intergiciel de lemmatisation. Nous le présentons en 3 sous-sections : motivations, conception globale et expérimentation/validation. Pour éviter une longue description purement technique, les spécifications détaillées sont mises en annexe (Annexe 10).

IV.2.1 Motivations

IV.2.1.1 Support de systèmes de TA de type MOSES

MOSESLIG-FR-ZH est un système de traduction automatique français→chinois, réalisé par L. X. Wang dans le cadre de sa thèse [Wang, L. X., 2015]. Il a utilisé 80000 bisegments post-édités pour construire le modèle de traduction.

Les entrées (les ressources pour entraîner le modèle de traduction et également les textes à traduire) des systèmes de TA de type Moses n'acceptent que les textes avec des séparateurs de mots. Il le faut pour toutes les langues.

Par exemple, pour le français, *c'est* → *c* ▯ *est*.

Pour le chinois, c'est difficile à faire. On est obligé d'utiliser des segmenteurs spécifiques. Par exemple, *这周末有日内瓦车展。*(Il y a le salon de l'auto de Genève ce weekend.) → *这* (ce)▯*周末*(weekend)▯*有*(avoir)▯*日内瓦*(Genève)▯*车展*(le salon de l'auto).

Au début, L. X. Wang a utilisé l'outil XELDA[1] de XEROX avec la licence de L&M[2]. Après sa thèse, il ne pouvait plus l'utiliser, et il a utilisé des segmenteurs libres de droits comme celui de Stanford et celui de XMU (Université de Xiamen).

D'autre part, pour entraîner les systèmes MOSES à facteurs, en plus de la segmentation en mots, il faudra annoter les corpus par des résultats d'analyse morphologique et éventuellement de désambiguïsation lexicale (WSD).

IV.2.1.2 Consultation dictionnairique avancée

IV.2.1.2.1 État de l'art (de l'existant)

La recherche avec lemmatisation et segmentation en mots est une fonction assez répandue. Par exemple, dans le projet OMNIA [Rouquet, D. & Nguyen, H. T., 2009], les mots composés sont pris en compte. Cela veut dire que, pour le texte `white paper wall`, on a besoin de consulter les possibilités suivantes dans la BDLex : `white`, `paper`, `wall`, `white paper`, `paper wall` et `white paper wall`. On a utilisé NOOJ et DELAF pour ce projet (voir II.2.3.1).

Dans leurs thèses, H.-T. Nguyen [Nguyen, H.-T., 2009] et C.-P. Huynh [Huynh, C.-P., 2010], ont proposé d'intégrer la recherche dictionnairique avec l'analyse morphologique (ex. PIVAX-1) dans le système THAM (ex. SECTRA) en utilisant l'outil NOOJ[3]. Ils ont également discuté des possibilités d'utiliser PILAF[4] pour le français.

D'autre part, pour la plate-forme JIBIKI, on a besoin d'une fonction de ce type pour toutes les BDLex. M. Mangeot a utilisé l'analyseur morphologique du japonais MECAB[5] et l'analyseur du français TREETAGGER[6] pour réaliser la fonction "lecture active"[7] dans le cadre de son projet JIBIKI-JAPONAIS→FRANÇAIS à l'Université Hosei [Mangeot, M., 2016]. Cette fonction a également été reprise dans le sous-projet LEXINNOVA du projet INNOVALANGUES.

IV.2.1.2.2 Analyse

Le point important est que, jusqu'ici, dans le contexte de JIBIKI, ou des autres BDLex ayant la possibilité de lemmatiser ou de segmenter en mots, on doit écrire du code spécifique pour chaque langue de chaque BDLex et chaque lemmatiseur, et on ne peut appeler qu'un lemmatiseur à la fois. Notre objectif ici est d'avoir un outil générique permettant d'intégrer facilement tous ces analyseurs dans nos systèmes.

IV.2.1.3 Production de mini-dictionnaires de formats variés

La dernière motivation est de disposer d'un outil permettant de construire des mini-dictionnaires. Nous présenterons la production de mini-dictionnaires dans la section suivante (IV.3). Pour pratiquement tous, il faut commencer par une lemmatisation.

---

[1] XELDA : http://www.xrce.xerox.com/About-XRCE/History/Historical-projects/XeLDA

[2] L. X. Wang a fait une thèse CIFRE avec Lingua et Machina.

[3] On a mentionné NOOJ au I.3.1.3.1, pour plus détail, voir http://www.nooj-association.org.

[4] PILAF (Procédures Interactives Linguistiques Appliquées au Français) [Courtin, J. et al., 1992 ; Courtin, J. & Genthial, D., 1998] est un logiciel réalisant l'analyse et la génération morphologiques, la lemmatisation, et la traduction au niveau morphosyntaxique du français. Créé par J. Courtin en 1975-1976, il a été considérablement étendu par D. Genthial, qui lui a ajouté un composant d'analyse de dépendances.

[5] MECAB : http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

[6] TREETAGGER : http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[7] Lecture active : ajout de prononciation et traductions aux textes.

IV.2.2 Conception de LEXTOH

IV.2.2.1 Fonctionnalités désirées

Serveur. LEXTOH est un serveur paramétrable d'appel de lemmatiseurs, ou plus généralement de segmenteurs, de racineurs, ou d'analyseurs morphologiques complets. Nous l'appelons en bref "intergiciel de lemmatisation". Il fonctionne comme un service Web avec méthodes POST ou GET.

Interfaces. Les programmes peuvent accéder au service par l'API, et les utilisateurs humains peuvent accéder au service par l'API ou par un formulaire.

La requête d'URL par l'API contient les paramètres texte, langue, choix de lemmatiseur, format de sortie, formalisme de représentation etc., voir le paragraphe suivant (Entrées et sorties).

L'interface de type formulaire permet à des utilisateurs humains d'accéder facilement au service en utilisant des boutons et des zones de texte. On a repris le formulaire proposé par V. Bellynck, formulaire général auto-rempli[1] en PHP.

Outils à intégrer. Il y a plusieurs outils disponibles selon les langues comme JIEBA, XIP, XELDA, ARIANE-HELOISE, STANDFORD-CORENLP, STANDFORD-SEGMENTER, le DELAF, etc. On donnera une brève présentation de chaque outil au IV.2.2.2.2. LEXTOH est un système générique, c'est-à-dire que chaque outil y est intégré par un plugin.

Entrées et sorties. Les entrées de LEXTOH contiennent les paramètres suivants.

- `lang` : langue source, 3 caractères latins selon la norme iso 639-2/T.
- `lemmat` : choix du lemmatiseur.
- `output` : format de sortie (ex. TXT, JSON, XML).
- `formalism` : grammaire formelle (ex. `réduit`, `graphe-Q`), voir l'explication ci-dessous.
- `text` : texte à analyser, un texte directement extrait d'un document.
- `debug` : 1 ou 0, il permet d'afficher tous les paramètres et leurs valeurs de requête dans le composant de `debug`, sur la même interface que le résultat.
- `window` (pour le DELAF) : offset des résultats de tokenisation (ex. window=3, text=pomme de terre, le résultat de tokenisation est "pomme", "de", "terre", donc le système va vérifier les lemmatisations pour "pomme", "pomme de", "pomme de terre", "de", "de terre", "terre").
- `form` : 1 ou 0, si 1, affichage du formulaire ; si 0, affichage seulement des résultats (API-REST).

Pour les sorties, il y a les formalismes de sortie et les formats de sortie. Les formalismes définissent les éléments affichés dans une sortie et la forme de l'affichage.

Par exemple, le formalisme `graphe-Q(AC)`[2] est une combinaison de type

```
forme + (lemme ou UL) + POS + toutes les informations (genre, nombre,
        temps, etc.) + géométrie en graphe de chaînes.
```

Le formalisme `réduit` est une combinaison de type

```
forme + lemme + POS + géométrie en liste sans répétition.
```

[1] Auto-rempli : quand on envoie une requête à l'API, si on demande d'affichier le formulaire, le formulaire est rempli automatiquement avec les valeurs de la requête.

[2] AC = analyse complète

On peut afficher chaque formalisme dans n'importe quel format de sortie. Par exemple, pour le format JSON, on a ce qui suit.

- Formalisme `graphe-Q`

```
Entrée : ,
var JSONObject=[
{"system":"delaf", "langue":"fra", "output":"json", "formalism":"graphe-Q(AC)"},
{-0-$OCC($FORME(les),$LU($LEMMA(le),$CAT(DET),$AC(gender_masculine,number_plural)))-3-},
{-0-$OCC($FORME(les),$LU($LEMMA(la),$CAT(DET),$AC(gender_feminine,number_plural)))-3-},
{-0-$OCC($FORME(les),$LU($LEMMA(le),$CAT(PRON),$AC(person_3,gender_masculine,number_plural)))-3-},
{-0-$OCC($FORME(les),$LU($LEMMA(le),$CAT(PRON),$AC(person_3,gender_feminine,number_plural)))-3-},
{-3-$OCC($FORME(#U#20#),$LU($LEMMA(#U#20#),$CAT(UNK),$AC(none)))-4-},[1]
{-4-$OCC($FORME(pommes),$LU($LEMMA(pomme),$CAT(NOUN),$AC(gender_feminine,number_plural)))-10-},
{-4-$OCC($FORME(pommes),$LU($LEMMA(pommer),$CAT(VERB),$AC(tense_ind,person_2,number_singular)))-10-},
{-4-$OCC($FORME(pommes),$LU($LEMMA(pommer),$CAT(VERB),$AC(tense_subj,person_2,number_singular)))-10-}
]
```

- Formalisme `réduit`

```
Entrée : les pommes
var JSONObject=[
{"system":"delaf", "langue":"fra", "output":"json", "formalism":"réduit"},
{"form":"les", "lemma":"le", "pos":"DET"},
{"form":"les", "lemma":"la", "pos":"DET"},
{"form":"les", "lemma":"le", "pos":"PRON"},
{"form":"les", "lemma":"la", "pos":"PRON"},
{"sep":" "},[2]
{"form":"pommes", "lemma":"pomme", "pos":"NOUN"},
{"form":"pommes", "lemma":"pommer", "pos":"VERB"}
]
```

LEXTOH permet aussi d'afficher les sorties des différentes étapes (le flot de traitement), de la sortie native (le retour d'appel de lemmatiseur) à la sortie finale, voir l'Annexe 10.

Configurations et autres fonctionnalités. Les fichiers de configuration permettent de configurer les outils, les langues, les formalismes et les formats utilisables. On a prévu également d'offrir d'autres fonctions dans cette interface par formulaire, comme la fonction "trace ou debug" pour le débogage, l'interface d'explication, les exemples d'utilisation d'API et d'utilisation de formulaire, etc.

Robustesse. La fiabilité du système est nécessaire pour qu'on puisse lui envoyer un grand nombre de requêtes en même temps. Pour la gestion de files d'attente, on utilise ACTIVEMQ.

IV.2.2.2 Architecture globale de LEXTOH

IV.2.2.2.1 Schéma de l'architecture logicielle

LEXTOH est composé de trois couches principales : une couche de présentation, une couche de traitement et une couche de ressource/outil. Voir la Figure 75.

Couche de présentation. Elle est responsable des interactions avec les utilisateurs. Après avoir reçu une requête dans cette couche, le système envoie les paramètres à la couche de traitement.

Couche de traitement. Dans cette couche, il y a plusieurs modules. Le module "traitement de requête" est le module principal.

Le deuxième module est le module d'appel aux lemmatiseurs (les plugins, un plugin pour chaque outil). Chaque plugin contient deux fonctions. La première est le prétraitement et la méthode d'appel. La deuxième est le post-traitement (nettoyage, changement d'encodage, etc.).

---

[1] C'est le séparateur entre "les" et "pommes". Pour être complet, on doit indiquer les séparateurs dans ce graphe, par exemple : on pourrait remplacer `{-3-$OCC($FORME(#U#20#)-4-}` par `{-3-$SEP(%%__)-4-}`.

[2] C'est le séparateur entre "les" et "pommes".

Le troisième module est le module "traitement commun". Ce module comporte plusieurs étapes de traitement pour obtenir une sortie bien uniformisée.

Le quatrième module est le module de "sortie". Il s'agit de transformer la sortie uniformisée vers une sortie demandée par l'utilisateur, combinant un formalisme et un format.

Couche de ressource. Cette couche correspond aux lemmatiseurs externes et aux fichiers de configuration.

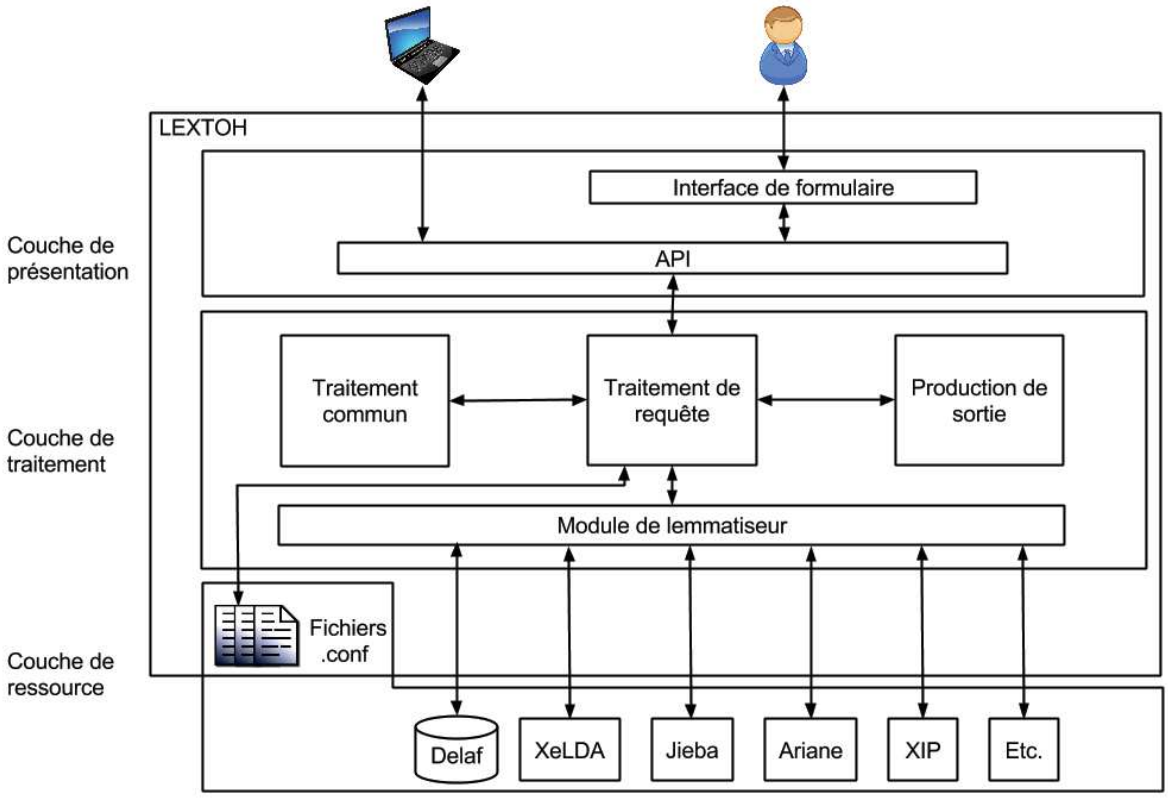


Figure 75 : Architecture de LEXTOH

IV.2.2.2.2 Outils appelés

JIEBA[1] est un segmenteur du chinois (avec un peu d'analyse syntaxique), libre de droits.

XIP[2] est un analyseur syntaxique de XEROX, créé par Claude Roux. Il contient pour chaque langue traitée, l'analyseur morphologique développé en XSLT et contenu dans le produit XELDA. Le GETALP en a une licence de recherche pour le français et l'anglais.

XELDA[3] est un produit de XEROX qui contient également les analyseurs morphologiques pour une vingtaine de langues. L&M en a une licence industrielle, et j'ai pu l'utiliser durant ma thèse.

ARIANE-HELOÏSE contient tous les analyseurs morphologiques écrits en ARIANE et développés au sein du GETA, puis du GETALP pour l'allemand. V. Berment a créé une API permettant d'appeler tous les analyseurs qui tournent sur son serveur HELOÏSE[4] [Berment, V. & Boitet, C., 2012]. Il y en a pour le russe, le français, l'anglais, le portugais, l'espagnol (depuis 2015), et l'allemand.

[1] JIEBA : https://github.com/fxsjy/jieba.
[2] XIP : https://open.xerox.com/Services/XIPParser/Pages/Using%20XIP.
[3] XELDA : http://www.xrce.xerox.com/About-XRCE/History/Historical-projects/XeLDA.
[4] ARIANE-HELOÏSE : http://laosoftware.com/Ariane/AnaFRDemo/Heloise.htm.

CoreNLP[1] contient des analyseurs morphologiques développés sur cette plate-forme par Alshawi et son équipe de l'université de Stanford, libres de droits.

Segmenter[2] est un autre outil de l'université de Stanford. Il contient les segmenteurs de l'arabe et du chinois, libres de droits.

Le **Delaf**[3] est un dictionnaire de formes (par opposition au Delas, dictionnaire de lemmes) développé par le LADL (M. Gross et son équipe) pour le français et pour l'anglais. La forme Xml de cette ressource est téléchargeable gratuitement. Pour l'utiliser, on peut le convertir en SQL par un script puis l'importer dans une base de données.

En plus des lemmatiseurs présentés ci-dessus, LexToH a prévu d'appeler trois outils externes.

SegDoc [Kalitvianski, R., 2013] est un outil pour segmenter un grand texte source.

Tracompl [Guillaume, P., 1989] est un des langages spécialisés d'Ariane-G5. Il permet de programmer très simplement la transformation d'une décoration d'un "jeu de décorations" `Jeu1` vers un "jeu de décorations" `Jeu2`. On peut l'utiliser pour normaliser les symboles utilisés pour noter les variables et leurs valeurs.

Par exemple, il existe des normes différentes pour les résultats d'analyse morphologique pour différentes langues et différents outils. Cela permettra de diminuer la programmation, de fusionner les résultats de plusieurs analyseurs morphologiques si nécessaire, et de filtrer pour ne sortir que les variables/valeurs utiles pour la recherche dans une BDLex.

ATEF [Chauché, J., 1975] développé en 1972 par J. Chauché, P. Guillaume et M. Quézel-Ambrunaz, est le langage utilisé en Ariane-G5 pour écrire les analyseurs morphologiques. On l'a déjà mentionné au III.4.2.3.

Pour les communications avec Tracompl et ATEF, nous avons spécifié et nous sommes en train de réaliser les implémentations, mais ce n'est pas encore mis en service. Pour la version actuelle publiée sur notre serveur, nous les avons réalisées de façon ad hoc.

IV.2.2.3 Utilisateurs et scénarios

IV.2.2.3.1 Utilisateurs

a. *Humain*

Rôle d'utilisateur humain	Droits
Administrateur	• Modification de la liste des outils appelables (leur "déclaration") et modification de la configuration. • Lancement du serveur.
Linguiste lexicographe	• Modification de certains paramètres (par exemple, les filtres), ou renommage d'une version. • Aussi et surtout : tests du fonctionnement de LexToH.
Linguiste-informaticien	• Exécution de tâches spécifiques (par exemple, ajout d'un nouvel outil en créant un plugin).

[1] CoreNLP : http://stanfordnlp.github.io/CoreNLP/.
[2] Segmenter : http://nlp.stanford.edu/software/segmenter.shtml.
[3] Delaf : http://infolingu.univ-mlv.fr/DonneesLinguistiques/Dictionnaires/telechargement.html. Le Delaf français contient 683 824 entrées simples pour 102 073 lemmes différents et 108 436 entrées composées pour 83 604 lemmes différents. Exemple : `pommes,pommer.V:P2s:S2s` et `pommes,pomme.N+z1:fp`. Le Delaf anglais contient 296 606 entrées simples pour 150 145 lemmes différents et 132 990 entrées composées pour 69 912 lemmes différents.

b. *Machine*

Rôle d'utilisateur machine	**Tâches**
CREATDICO (que l'on présentera au IV.3)	CREATDICO sera appelé par SECTRA/IMAG. Il appellera LEXTOH pour obtenir une sortie (Lemme + POS), puis appellera les services dictionnairiques, construira un mini-dictionnaire, et le renverra à l'appelant.
JIBIKI	Lemmatisation pour consultation d'un dictionnaire.
JIANDAN-EVAL[1]/MOSESLIG	Pour obtenir des "textes annotés" par l'analyseur morphologique, destinés à l'apprentissage de systèmes MOSES à facteurs. Segmentation des entrées multilingues (surtout pour le chinois).
Programme non encore spécifiés. On pense à OMNIA.	Annotation et extraction de contenu.

IV.2.2.3.2 Scénarios

Les utilisations de LEXTOH pourraient être nombreuses. On donne ici quatre scénarios complètement différents. Le premier est une utilisation par un programme, avec un grand nombre de requêtes. Le deuxième est une utilisation par un programme, avec une requête qui est activée par un humain. Le troisième est une utilisation directe sur l'interface de LEXTOH. Le quatrième concerne la configuration.

On montre les résultats de ces scénarios (avec des mesures et des interfaces) à la section suivante IV.2.3.2.

a. *Scénario 1 : Demande de JIANDAN-EVAL d'annoter un corpus en utilisant ACTIVEMQ*

1. JIANDAN-EVAL demande 1 000 requêtes à la fois (annotation de 1 000 segments postédités) par un bouton d'interface de JIANDAN-EVAL. Il s'agit de 310 segments anglais (155 avec l'outil XELDA, 155 avec l'outil XIP), 280 segments chinois avec l'outil JIEBA et 410 segments français (205 avec l'outil XELDA et 205 avec l'outil XIP) dans la mémoire de traductions.
2. Le *producteur* associé à JIANDAN-EVAL envoie ses requêtes à ACTIVEMQ avec deux types de paramètre : paramètres d'attributs de message (il s'agit des paramètres d'ACTIVEMQ, par exemple, `destination` = LEXTOH, `Priority` = 4) et paramètres de contenu de message (il s'agit de paramètres de LEXTOH, par exemple le segment, la langue, le choix du lemmatiseur et la sortie).
3. ACTIVEMQ enregistre ces requêtes dans le serveur.
4. Si LEXTOH est libre, le *consommateur* de LEXTOH reçoit une requête à la fois selon la priorité du message et selon le moment d'arrivée. Puis ACTIVEMQ considère que LEXTOH est occupé.
5. LEXTOH traite un message. Pour ces 1 000 requêtes, il y a deux cas différents :
    a. Si le traitement est fini et est réussi, le *consommateur* associé à LEXTOH est libre et envoie un symbole `ACKNOWLEDGE` à ACTIVEMQ. L'attribut `ReplyTo` de cette requête est JIANDAN-EVAL, le *consommateur* de LEXTOH demande au producteur de LEXTOH d'envoyer le résultat de cette requête à JIANDAN-EVAL (avec deux types de paramètres : paramètres d'attributs de message et paramètres de contenu de

[1] JIANDAN-EVAL est une plate-forme de construction, déploiement, évaluation et amélioration incrémentale continue de systèmes de TA de type MOSES, développée par L. X. Wang dans le cadre de sa thèse [Wang, L. X., 2015]. Ce système est déjà utilisé pour la construction de plusieurs versions du système de TA MOSESLIG-FR-ZH.

message, comme à l'étape 2). Ce programme est lancé en parallèle avec le consommateur de LEXTOH.

b. Si LEXTOH termine son traitement sans réussite, le consommateur de LEXTOH est libre et sans symbole `ACKNOWLEDGE`. La requête sera renvoyée par ACTIVEMQ.

6. Boucler sur les étapes 4 et 5 jusqu'à ce qu'il n'y ait plus de requête.

b. *Scénario 2 : Demande de lemmatisation par PIVAX-2*

On utilise ACTIVEMQ pour la gestion des travaux, comme dans le scénario 1. Comme une description avec ACTIVEMQ est longue et compliquée, on simplifie les scénarios suivants.

Par exemple, ce scénario 2 concerne une demande de lemmatisation par PIVAX-2. La procédure est PIVAX-2→ACTIVEMQ→LEXTOH→ACTIVEMQ→PIVAX-2.

On simplifie : PIVAX-2→LEXTOH→PIVAX-2.

Il y a une chose à remarquer ici, c'est que la requête est lancée par un humain, pas par un programme, donc nous fixons une priorité élevée : `Priority` = 7[1].

C'est une réalisation/spécification dans le *producteur* de PIVAX-2.

1. L'utilisateur arrive sur la page de consultation avancée de PIVAX-2.
2. Il entre les conditions de consultation avec les paramètres : `le mot-vedette est` "*allongerons*" et `le lemmatiseur` est "*xip*".
3. Il lance la requête par le bouton "➔".
4. La page est mise à jour et le résultat est affiché.

c. *Scénario 3 : Utilisation d'un exemple de formulaire par un linguiste*

Il s'agit d'un linguiste-informaticien ou d'un lexicographe qui fait une expérience dans la page de formulaire en utilisant l'exemple (le lien pré-rempli par un clic) en bas de la page.

1. L'utilisateur arrive à la page d'accueil de LEXTOH.
2. Il clique sur un lien d'exemple en bas de la page Web : http://46.105.41.94/Ci-Hai/Lextoh/index.php?text=bonjour, tu vas bien?&lemma=ariane-heloise&window=5&lang=fra(fra4)&output=txt&formalism=reduit&formule=1
3. La page est mise à jour avec des paramètres pré-remplis et le résultat est affiché.
4. L'utilisateur clique sur le bouton "mode avancé" pour suivre le flot de traitement.

d. *Scénario 4 : Configuration des outils appelables par l'administrateur*

Il s'agit d'un administrateur qui modifie les outils appelables et leurs décorations par les fichiers de configuration.

1. L'administrateur ouvre le fichier `conf_ressource.xml`.
2. Il désactive l'analyseur morphologique XELDA (il a mis la description de XELDA en commentaire). Voir la Figure 79.

IV.2.3 Expérimentation et validation

IV.2.3.1 Interface principale et disponibilité

La version interne est la version la plus récente, installée sur le serveur interne `danang.imag.fr` du GETALP. On a travaillé et testé sur cette version. Mais ce n'est pas une version stable, et elle ne permet pas l'accès depuis Internet.

La version publiée est installée sur http://46.105.41.94/Ci-Hai/Lextoh.

[1] Les valeurs de `Priority` sont entre 0 et 9. La valeur par défaut est 4 (faible si < 4, et haute si > 4).

L'interface principale est montrée dans la Figure 76.

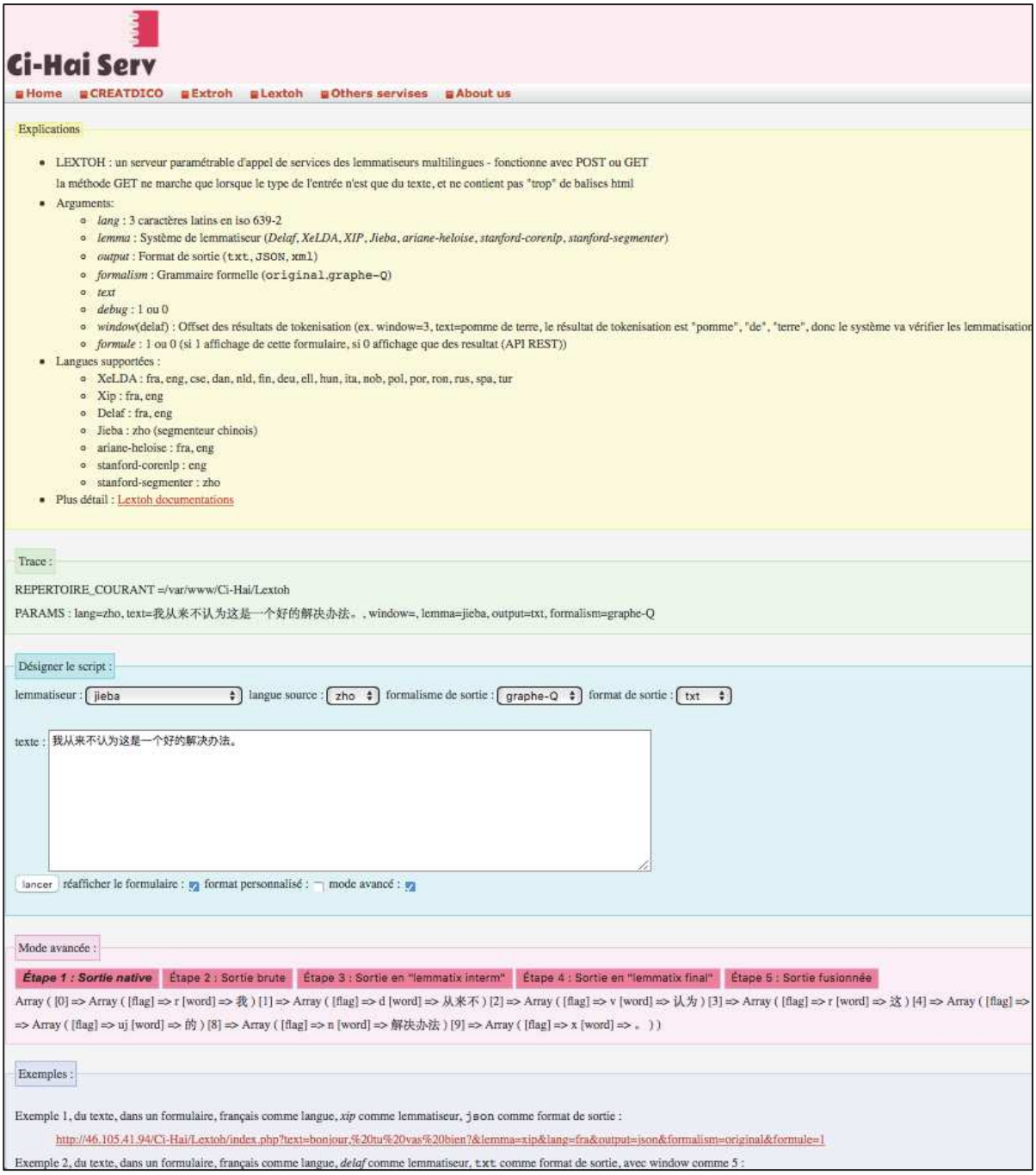


Figure 76 : Interface principale de LEXTOH

IV.2.3.2 **Test du système**

IV.2.3.2.1 Utilisation pour l'annotation de corpus (JIANDAN-EVAL)

Voici le résultat pour le scénario 1.

Nbr req	Langues	Lemmatiseurs	Temps total	Remarque
310	Anglais	XELDA ou XIP	28min20s	Annotation de corpus : s'il n'y avait que des segments français et anglais, ça serait beaucoup plus rapide. Mais avec le chinois, le lancement de JIEBA est lent.
280	Chinois	JIEBA		
410	Français	XELDA ou XIP		

IV.2.3.2.2 Utilisation pour la consultation de dictionnaires (PIVAX-2)

On a déjà présenté cette utilisation au II.3.3. Le résultat est affiché sur la Figure 30.

IV.2.3.2.3 Utilisation d'un exemple de formulaire par un linguiste (résultat du scénario 3)

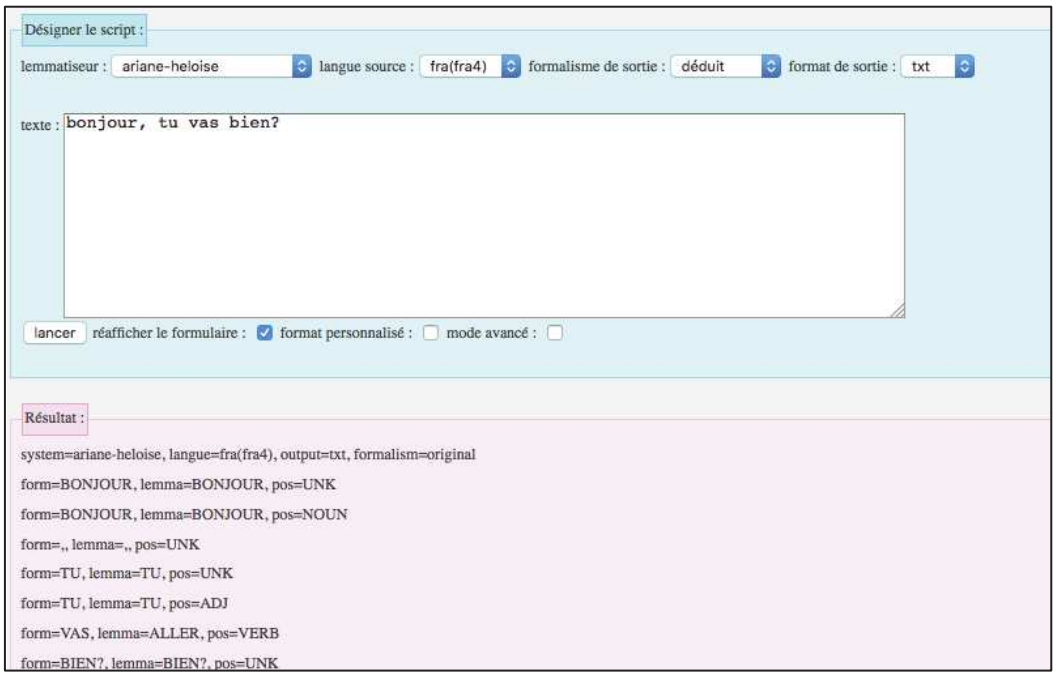


Figure 77 : Résultat de LEXTOH pour l'exemple utilisant ARIANE-HELOISE

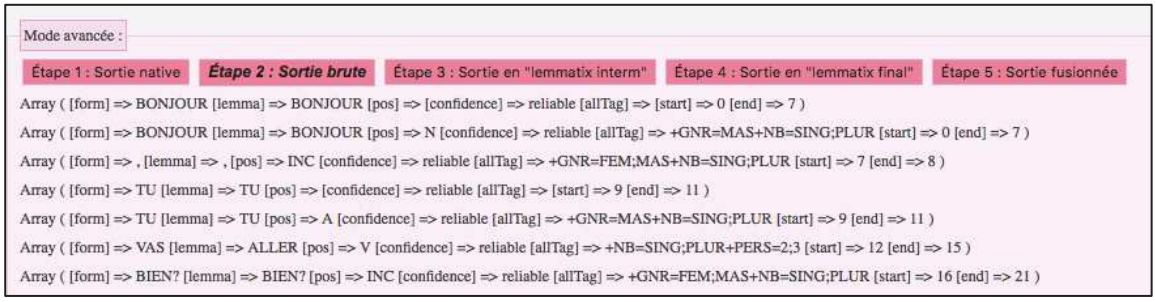


Figure 78 : Résultat en mode avancé de LEXTOH pour l'exemple utilisant ARIANE-HELOISE

IV.2.3.2.4 Configuration des outils appelables par l'administrateur (résultat du scénario 4)

Après la désactivation de XELDA, XELDA n'est plus contenu dans la liste de lemmatiseurs affichée sur l'interface. On ne peut plus l'utiliser. Si on demande les services sur l'API, le système nous envoie un message d'erreur (lemmatiseur indisponible).

```
<RESSOURCES dateCreation="20140221" dateDerniereModif="201501218" typeRessources="parOutil">

<Outils codeLM = "xip" codeEnt = "utf8" codeLg = "fra eng" sortiesPossibles = "arbre_A" url =
"https://open.xerox.com/Services/XIPParser/" requeteType = "API" requete="http://atoum.imag.fr
/getalp/Services/Web/CREATDICO/callXip.php?lang=$langSource&text=$fichEnt"/>

<!-- Modifié par Ying. Il n'y a plus de license à partir du 1 novembre 2015 -->
<!--Outils codeLM = "XeLDA" codeEnt = "utf8" codeLg = "fra eng cse dan nld fin deu ell hun ita
nob pol por ron rus spa tur" sortiesPossibles = "liste xml" url =
http://www.xrce.xerox.com/About-XRCE/History/Historical-projects/XeLDA/ requeteType = "EXEC"
requete = "xelda MorphoAnalysis plaintext $fichSort $fichEnt $langSource FST FSTPOSTag"/-->
……
</RESSOURCES>
```

Figure 79 : Fichier de configuration des outils appelables

Figure 80 : Affichage d'un message d'erreur de LEXTOH (lemmatiseur indisponible)

IV.3 CREATDICO

CREATDICO est un intergiciel de création de "mini-dictionnaires". Après l'analyse morphologique d'un texte ou d'un "mot-forme", on peut recevoir les lemmes grâce à LEXTOH. CREATDICO permet de fabriquer le dictionnaire pour chaque lemme de façon générique.

Cette section est divisée en 3 sous-sections : motivations, conception globale et expérimentation/validation. Comme pour LEXTOH, les spécifications détaillées ont été mises en annexe (voir l'Annexe 11).

IV.3.1 Motivations : des besoins variés

IV.3.1.1 Besoins des systèmes d'aide à la traduction

L'intégration des mini-dictionnaires dans les systèmes de THAM est un objectif déjà présenté et justifié dans la thèse de Huynh [Huynh, C.-P., 2010].

Pour le support des informations lexicales, on a besoin de mini-dictionnaires pour réaliser la fonction d'aide lexicale proactive. Il s'agit de "permettre à des contributeurs d'accéder pendant la post-édition à des ressources lexicales distantes de façon proactive".

La solution présentée dans [Huynh, C.-P., 2010] est le stockage d'un mini-dictionnaire associé à un segment dans la base de données, et les mini-dictionnaires sont préparés à l'avance et toujours disponibles pour la post-édition.

Définition 26. Un **mini-dictionnaire** contient les informations lexicales associés aux mots d'un fragment de texte, le plus souvent un segment. (Ces informations contiennent souvent les traductions dans une ou plusieurs langues.)

IV.3.1.1 Besoin humain de lecture active

On a déjà mentionné la fonction "lecture active" au IV.2.1.2.1. Cette fonction a déjà été implémentée dans plusieurs systèmes, comme FICUS [Lafourcade, M. & Chauché, J., 1998] par M. Lafourcade, et JIBIKI-JAPONAIS→FRANÇAIS par M. Mangeot (voir la Figure 82) [Mangeot, M., 2016], et aussi dans quelques outils comme Kindle.

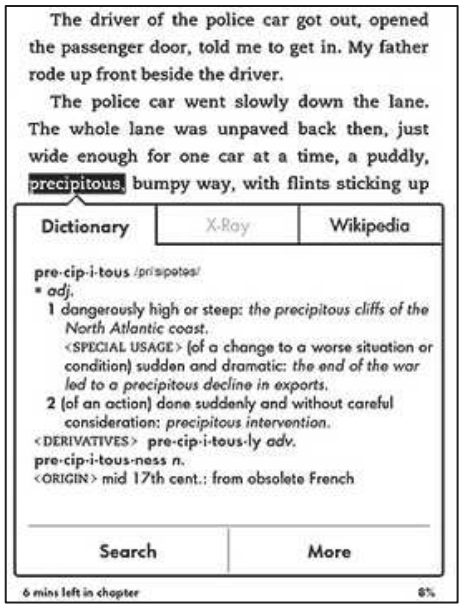


Figure 81 : Lecture active dans Kindle

Le logiciel IMAG (interactive Multilingual Access Gateway) a été développé par C.-P. Huynh en parallèle avec SECTRA (voir IV.1.1.1.1) comme une extension qui en est un "frontal". Il permet de naviguer sur des sites Web dans diverses langues, et d'améliorer les prétraductions dans le contexte de lecture. IMAG a également un besoin de lecture active. Il y a aussi ce besoin pour le projet INNOVALANG-LEXINNOVA (voir III.4.2), car c'est très utile pour progresser dans une langue étrangère.

Sans CREATDICO, ces réalisations seraient faites séparément, et on devrait les programmer de nouveau pour chaque système et chaque couple de langues.

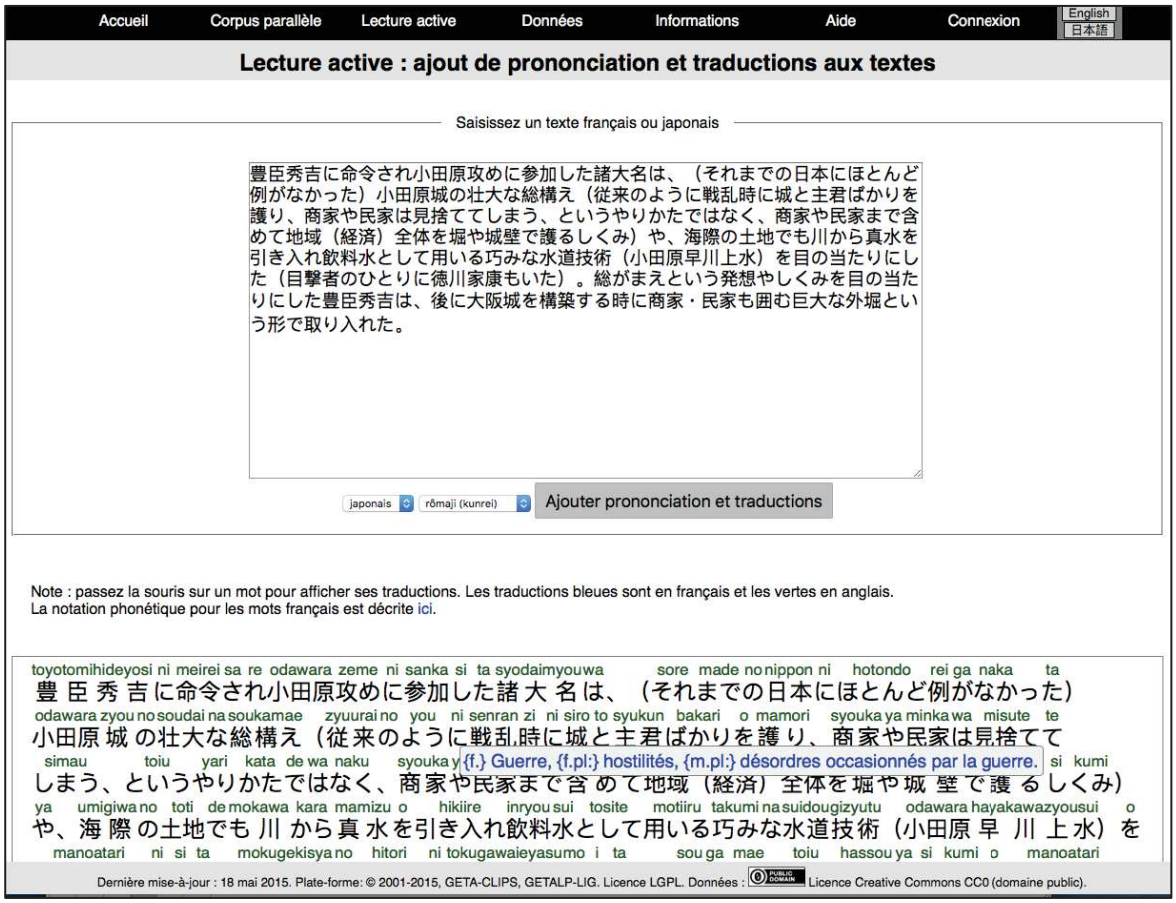


Figure 82 : Lecture active dans JIBIKI-JAPONAIS↔FRANÇAIS

IV.3.1.2 **Besoins pour des systèmes de TA**

Les systèmes de TA ont besoin de dictionnaires spécialisés ; voici deux exemples.

1. Dictionnaires utilisés dans un système de TA sous ARIANE-G5.

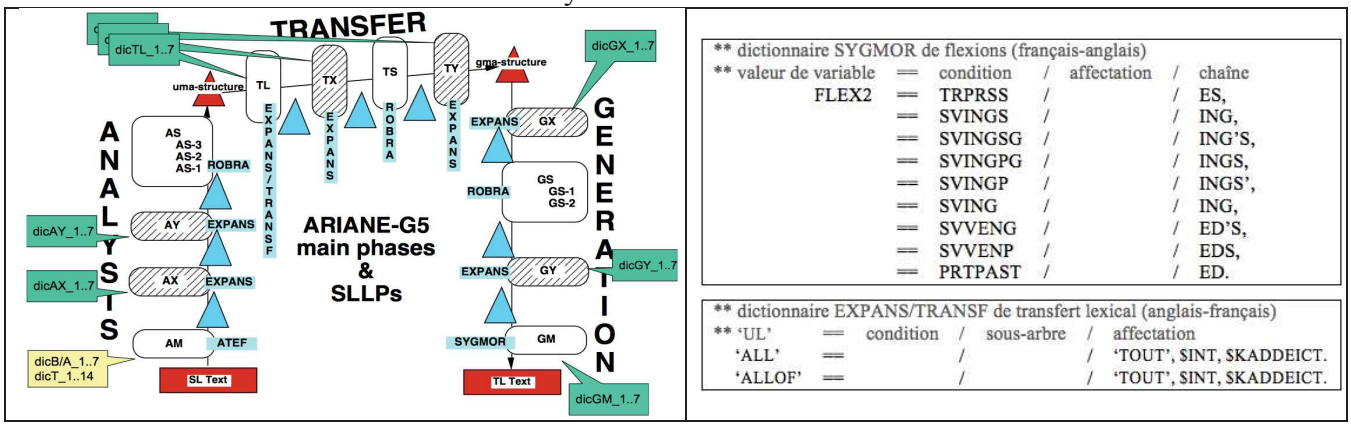
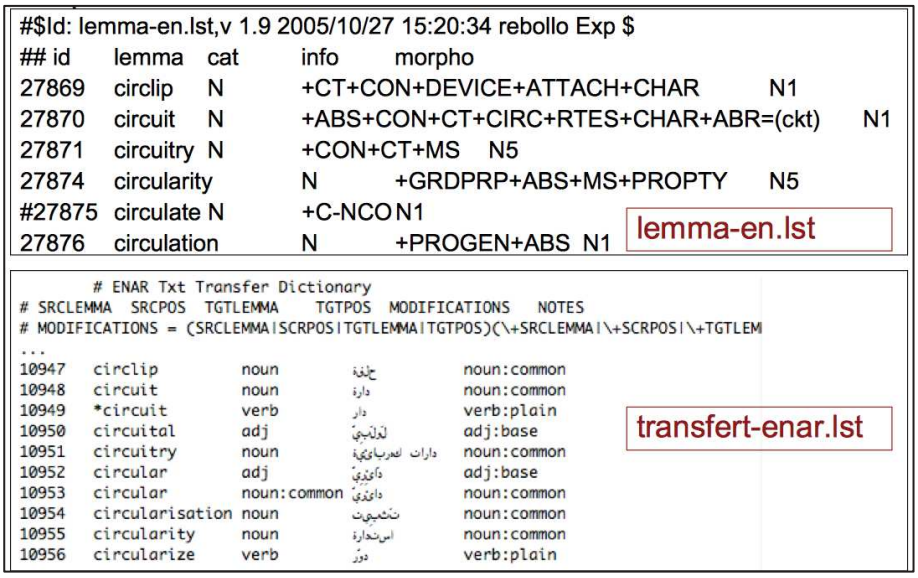


Figure 83 : Structure d'ARIANE-G5 et exemples d'entrées de dictionnaires

2. Dictionnaire pour SYSTRAN (lemma-en pour dictionnaire monolingue anglais, et transfert-enar pour dictionnaire bilingue anglais-arabe).



```
#$Id: lemma-en.lst,v 1.9 2005/10/27 15:20:34 rebollo Exp $
## id     lemma       cat    info          morpho
27869     circlip     N      +CT+CON+DEVICE+ATTACH+CHAR        N1
27870     circuit     N      +ABS+CON+CT+CIRC+RTES+CHAR+ABR=(ckt)    N1
27871     circuitry   N      +CON+CT+MS    N5
27874     circularity         N         +GRDPRP+ABS+MS+PROPTY     N5
#27875    circulate N         +C-NCO N1
27876     circulation         N         +PROGEN+ABS  N1
```

lemma-en.lst

```
        # ENAR Txt Transfer Dictionary
# SRCLEMMA   SRCPOS   TGTLEMMA     TGTPOS   MODIFICATIONS    NOTES
# MODIFICATIONS = (SRCLEMMA|SCRPOS|TGTLEMMA|TGTPOS)(\+SRCLEMMA|\+SCRPOS|\+TGTLEM
...
10947   circlip          noun          جليفة          noun:common
10948   circuit          noun          دارة           noun:common
10949   *circuit         verb          دار            verb:plain
10950   circuital        adj           لدوراني        adj:base
10951   circuitry        noun          دارات كهربائية  noun:common
10952   circular         adj           دائري          adj:base
10953   circular         noun:common   دائري          noun:common
10954   circularisation  noun          تدشيبات        noun:common
10955   circularity      noun          اس تدارة       noun:common
10956   circularize      verb          دوّر           verb:plain
```

transfert-enar.lst

Figure 84 : Exemples de dictionnaires SYSTRAN

IV.3.1.3 Besoins concernant des outils spécialisés

Comme dit plus haut au II.2.3.1, pour le projet OMNIA, on a eu besoin de mini-dictionnaires lemmes-UW sous forme de grammaires-Q. Pour l'outil SECTRA/IMAG, on en a aussi besoin, dans un format XML.

IV.3.2 Conception de CREATDICO

IV.3.2.1 Fonctionnalités désirées

Serveur. CREATDICO est un serveur paramétrable d'appel de dictionnaires ou de bases lexicales pour créer les "mini-dictionnaires" (un "intergiciel de création de mini-dictionnaires"). Comme LEXTOH, il fonctionne comme un service Web, avec les méthodes POST et GET.

Interfaces. CREATDICO a été développé juste après l'installation de la 1ère version de LEXTOH. L'interface de type formulaire a été développée à partir de celle de LEXTOH (voir IV.2.2.1).

Pour l'API, la requête d'URL contient les paramètres serveur, dictionnaire, langue source, langue cible, lemmatiseur, format de sortie, texte etc., voir le paragraphe suivant (Entrées et sorties).

Outils dictionnairiques. CREATDICO permet d'accéder à plusieurs bases lexicales et dictionnaires en ligne : PAPILLON-CDM, PIVAX-2, IATE, WIKTIONARY etc. À chaque service dictionnairique correspond un plugin.

Entrées et sorties. Les entrées de CREATDICO contiennent les paramètres suivants :

- `serv` : serveur de dictionnaire(s).
- `dico` : dictionnaire, pour le cas d'un serveur contenant plusieurs dictionnaires, comme PAPILLON-CDM[1]. S'il n'y a qu'un seul dictionnaire dans le serveur, ce paramètre n'est pas activé.
- `lang` : 3 caractères latins en ISO 639-2.
- `lc` : liste de langues cible, chacune notée par son code à 3 caractères (ISO 639-2).

[1] Il y a plusieurs dictionnaires dans le serveur du PAPILLON-CDM, ex. CEDICT, JMDICT, FEM, etc. Voir I.1.3.2.3.

- `lemmat` : outil d'analyse morphologique (choix parmi les outils disponibles sur LEXTOH).
- `output` : format de sortie.
- `text` : texte à analyser, sous forme d'une chaîne de caractères.
- `detail` : 1 ou 0, si 0 affichage des mots-vedette en traduction, et si 1 affichage de tous les détails. Dans certains types de sortie, ce paramètre n'est pas activé.
- `debug` : 1 ou 0, il permet d'afficher tous les paramètres et leurs valeurs de requête dans le composant de `debug`, sur la même interface que le résultat.
- `form` : 1 ou 0, si 1 affichage du formulaire, si 0 affichage des résultats seuls (API-REST).

On distingue les sorties générales et les sorties dédiées.

Les sorties générales sont des sorties par défaut. On présente ces formats dans l'Annexe 11. Les sorties dédiées (mini-dictionnaires) sont des sorties spécifiées et filtrées par les utilisateurs (ex. SECTRA, OMNIA etc.). Les formats dédiés sont réalisés par des plugins.

Il y a deux sous-types de mini-dictionnaire, un qui permet seulement la consultation, l'autre qui permet en plus de contribuer à des systèmes de type PIVAX (en cours d'implémentation, ce n'est pas encore mis en service), dans des limites précises. Pour permettre les contributions, on a prévu d'utiliser l'API de JIBIKI-2, réalisée par M. Mangeot.

Voici un exemple de mini-dictionnaire construit pour SECTRA (seulement la consultation).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SETRAMINIDICO "SETRAMINIDICO.dtd">
<minidico>
     <description>
          <format>minidico_sectra</format>
          <segment>Hello World</segment>
          <langueSource>eng</langueSource>
          <langueCible>zho</langueCible>
          <lemmatiseur>xip</lemmatiseur>
          <serv_dico>wikitionary</serv_dico>
          <id>demo2_doc18_seg1059</id>
          <date>20150705-10:59:54</date>
     </description>
     <dictionnaires>
          <dictionnaire>
               <lemme>hello</lemme>
               <traductions>
                    <traduction>（用于问候、接电话或引起注意）你好，喂，（用于询问所在的地方是否有人）请问
                         有人在吗</traduction>
                    <traduction>（表示惊讶）嘿</traduction>
               </traductions>
          </dictionnaire>
          <dictionnaire>
               <lemme>world</lemme>
               <traductions>
                    <traduction>世界</traduction>
                    <traduction>地球和包括其所有的棲息生物，及位於其上的所有東西(环境) </traduction>
               </traductions>
          </dictionnaire>
     </dictionnaires>
</minidico>
```

Configurations et autres fonctionnalités. Les fichiers de configuration permettent de configurer les outils utilisables.

Ils permettent également de configurer les préférences de l'utilisateur, par exemple, le lemmatiseur préféré, le service dictionnairique préféré, etc. Avec ces fichiers de configuration, les utilisateurs n'ont pas besoin d'envoyer tous les paramètres à chaque fois. Ils les envoient uniquement s'ils sont différents de ceux de la configuration courante.

Comme pour LEXTOH, on a réalisé les fonctions "trace ou debug" pour le débogage, l'interface d'explication, les exemples d'utilisation d'API et d'utilisation de formulaire, etc.

Robustesse. Comme pour LEXTOH, on utilise ACTIVEMQ pour la gestion des tâches et des files d'attente.

IV.3.2.2 Architecture globale de CREATDICO

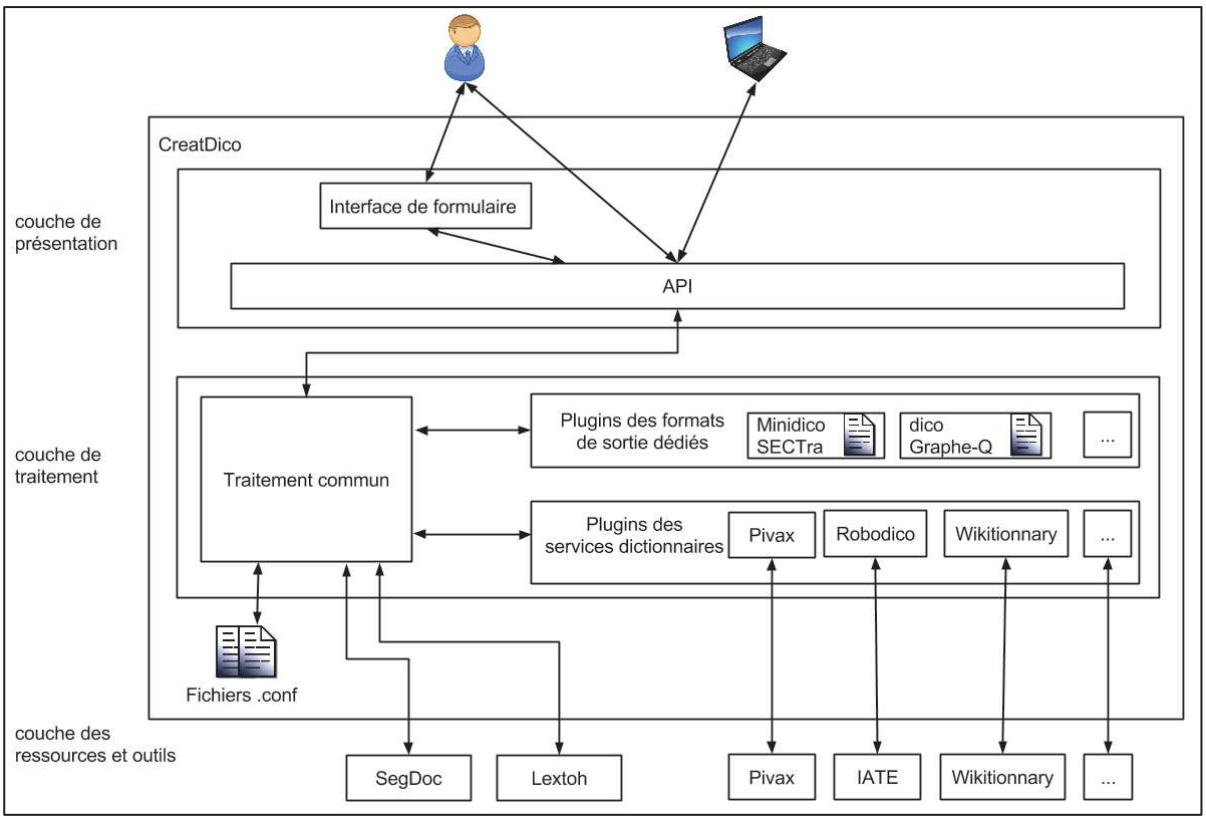


Figure 85 : Architecture de CREATDICO

Couche de présentation. Elle permet l'interaction avec les utilisateurs humains (par formulaire ou par API) et les programmes (par API via le serveur d'ACTIVEMQ).

Couche de traitement. Il y a un module "traitement commun" et deux plugins (l'un pour la consultation des dictionnaires, l'autre pour la production des mini-dictionnaires) dans cette couche.

Le module "traitement commun" contient les programmes principaux, qui réalisent plusieurs étapes de traitement pour traiter les requêtes, unifier le résultat natif des différents outils, communiquer avec les plugins, etc. Il est également responsable de la communication avec SEGDOC et LEXTOH.

Si l'entrée est un grand texte (plus d'une page standard[1]), ce module envoie le document à SEGDOC pour qu'il le segmente. Si l'entrée est un terme, un segment ou un petit texte, le

[1] Environ 1400 à 1500 caractères en français ou en anglais (250 mots), et environ 400 caractères en chinois ou en japonais.

système l'envoie à LEXTOH directement. Après avoir reçu les lemmes de LEXTOH, le module "traitement commun" les envoie à un plugin de service dictionnaire en fonction de la demande de l'utilisateur.

Chaque plugin de service dictionnaire correspond à un outil dictionnairique. Il spécifie le prétraitement, la méthode d'appel et le post-traitement.

Le dernier module est le module des plugins de sortie dédiés (les mini-dictionnaires). Chaque plugin réalise un mini-dictionnaire (transformation de la sortie uniformisée vers une sortie spécifiée). Il y a un fichier de configuration pour chaque plugin. Ces fichiers mémorisent les préférences des utilisateurs, par exemple quel lemmatiseur est préféré pour quelle langue, quel dictionnaire est préféré pour quel couple de langues.

Couche des ressources et outils. Cette couche correspond aux dictionnaires externes et aux fichiers de configuration de CREATDICO (ex. les services dictionnairiques et les lemmatiseurs appelables).

IV.3.2.3 Utilisateurs et scénarios

IV.3.2.3.1 Utilisateurs

a. *Humain*

Rôle d'utilisateur humain	**Droits**
Administrateur	• Activation ou désactivation de la liste des outils appelables et de la liste de sortie dédiée par les fichiers de configuration. • Lancement du serveur.
Linguiste lexicographe	• Négociation (ex : demande d'ajouter un autre outil de service dictionnaire, demande d'une sortie particulière etc.). • Tests du fonctionnement de CREATDICO.
Informaticien	• Exécution de tâches spécifiques (ex : ajout d'une nouvelle sortie dédiée).
Utilisateur indirect	• Par exemple, contribution ou évaluation sur les mini-dictionnaires via SECTRA/IMAG (pas encore mis en service).

b. *Machine*

Rôle d'utilisateur machine	**Tâches**
SECTRA	Demande de mini-dictionnaires.
IMAG	Demande de dictionnaires pour "lecture active".
OMNIA	Demande de mini-dictionnaires en grammaires-Q.
ARIANE	Demande de dictionnaires pour des systèmes de TA.
Autres	Programmes non encore spécifiés. On pense à l'évaluation comparative de différents services dictionnairiques.

IV.3.2.3.2 Scénarios

Comme pour LEXTOH, les utilisations de CREATDICO pourraient être nombreuses. On donne ici trois scénarios différents. Le premier est une utilisation complète, qui concerne LEXTOH, CREATDICO, SECTRA, PIVAX-2, et également ACTIVEMQ, avec un grand nombre de requêtes. Le deuxième est une utilisation directe sur l'interface de CREATDICO. Le troisième concerne la configuration.

On montre les résultats de ces scénarios à la section suivante 0.

a. *Scénario 1 : Un exemple complet de demande de mini-dictionnaire par SECTRA*

C'est une utilisation assez complète (globale) avec plusieurs systèmes : LEXTOH, CREATDICO, SECTRA et PIVAX-2. Les appels entre les systèmes sont réalisés par l'outil ACTIVEMQ. Pour faciliter l'explication, on cache l'utilisation d'ACTIVEMQ.

1. SECTRA appelle CREATDICO pour demander des mini-dictionnaires pour 50 segments français vers toutes les autres langues possibles.
2. Pour chaque segment, CREATDICO appelle LEXTOH avec les paramètres adéquats.
3. LEXTOH appelle XIP (dans la fichier de configuration, XIP est l'analyseur préféré pour la combinaison de SECTRA+français) pour réaliser l'analyse morphologique de ce segment.
4. XIP renvoie le résultat à LEXTOH.
5. LEXTOH produit le format souhaité et renvoie le résultat à CREATDICO.
6. CREATDICO envoie les lemmes à PIVAX-2 pour consulter la base lexicale.
7. PIVAX-2 envoie les résultats sous forme de dictionnaire en HTML à CREATDICO.
8. CREATDICO regroupe les dictionnaires associés aux différents lemmes du segment, crée la sortie en format dédié "mini-dictionnaire de SECTRA", et le renvoie à SECTRA.
   On boucle sur les étapes de 2 à 8, jusqu'à ce qu'il n'y ait plus de segment.

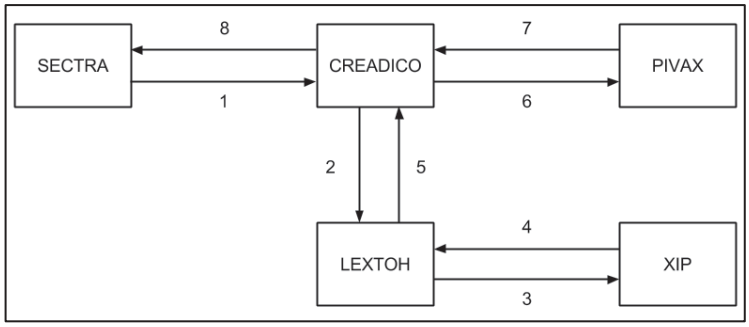


Figure 86 : Procédure de demande de mini-dictionnaires par SECTRA

b. *Scénario 2 : Utilisation de formulaire par un linguiste*

Il s'agit ici d'un linguiste-informaticien ou d'un lexicographe qui fait trois expériences en utilisant le formulaire.

Expérience 1. Consultation d'un segment avec une langue cible produisant une sortie simple.

1. L'utilisateur arrive à la page d'accueil de CREATDICO.
2. Il remplit le formulaire avec les paramètres suivants.
   - `Serveur = papillon`
   - `Dictionnaire = Cedict`
   - `langue source = zho`
   - `langue cible = eng`
   - `lemmatiseur = jieba`

- `type de sortie = txt` (une sortie générale)
- `text = 我喜欢去电影院。` (j'aime bien aller au cinéma.)
- `réafficher le formulaire = oui`

3. Il clique sur le bouton "`lancer`".
4. La page est mise à jour et le résultat est affiché.

Expérience 2. Consultation d'un segment avec demande des équivalents dans plusieurs langues cible produisant une sortie simple.

1. L'utilisateur modifie le formulaire avec les paramètres suivants.
   - `Serveur = iate`
   - `Dictionnaire = iate` (comme le serveur IATE n'a pas plusieurs dictionnaires, ce champ est rempli automatiquement après le choix du serveur.)
   - `langue source = eng`
   - `langue cible = toutes`
   - `lemmatiseur = xip`
   - `type de sortie = txt` (une sortie générale)
   - `text = Hello world.`
   - `réafficher le formulaire = oui`
2. Il clique sur le bouton "`lancer`".
3. La page est mise à jour et le résultat est affiché.

Expérience 3. Consultation d'un segment avec une langue cible produisant une sortie détaillée.

1. L'utilisateur modifie le formulaire avec les paramètres suivants.
   - `Serveur = papillon`
   - `Dictionnaire = Cedict`
   - `langue source = eng`
   - `langue cible = zho`
   - `lemmatiseur = delaf`
   - `type de sortie = txt` (une sortie générale)
   - `afficher les détaillés = oui` (une sortie générale détaillés)
   - `text = Hello world.`
   - `réafficher le formulaire = oui`
2. Il clique sur le bouton "`lancer`".
3. La page est mise à jour et le résultat est affiché.

c. *Scénario 3 : Configuration des outils appelables disponibles et modification des préférences d'une sortie dédiée par l'administrateur*

Il s'agit de deux niveaux de configuration.

(1) Configuration de CREATDICO : l'administrateur modifie les outils disponibles par les fichiers de configuration.
(2) Configuration d'un plugin de sortie spécifique : l'administrateur modifie le lemmatiseur préféré pour les mini-dictionnaires de SECTRA.

1.1. L'administrateur ouvre le fichier `conf_ressource.xml`.
1.2. Il désactive le service IATE (en mettant la description d'IATE en commentaire).
2.1. Il ouvre le fichier `conf_minidico_Sectra.xml`.
2.2. Il modifie le serveur préféré `serveur_pref = Pivax` pour la demande de mini-dictionnaire de fra→eng.

IV.3.3 Expérimentation et validation

IV.3.3.1 Interface principale et disponibilité

Comme LEXTOH, la version interne est installée sur le serveur `danang.imag.fr` du GETALP pour tests et développements. Elle ne permet pas l'accès depuis Internet.

La version publiée est installée sur http://46.105.41.94/Ci-Hai/CREATDICO. Le formulaire est similaire à celui de LEXTOH. Il y a 5 composants : Explications, Trace, Créer un script, Résultat et Exemples. On affiche seulement le composant "Créer un script" ici. Pour le formulaire complet, voir la Figure 76.

Figure 87 : Composant "Créer un script" de l'interface CREATDICO

IV.3.3.2 **Tests fonctionnels**

IV.3.3.2.1 Demande de mini-dictionnaires par SECTRA (résultat du scénario 1)

Le temps de réponse est acceptable. Le scénario 1 (50 segments français vers toutes les autres langues, voir IV.3.2.3.2a) a pris environ 4,5 minutes sans erreur. L'interface de SECTRA est montrée dans la Figure 88. Ici, le mini-dictionnaire est grand, c'est pourquoi L. X. Wang a inclus un ascenseur dans le composant d'affichage.

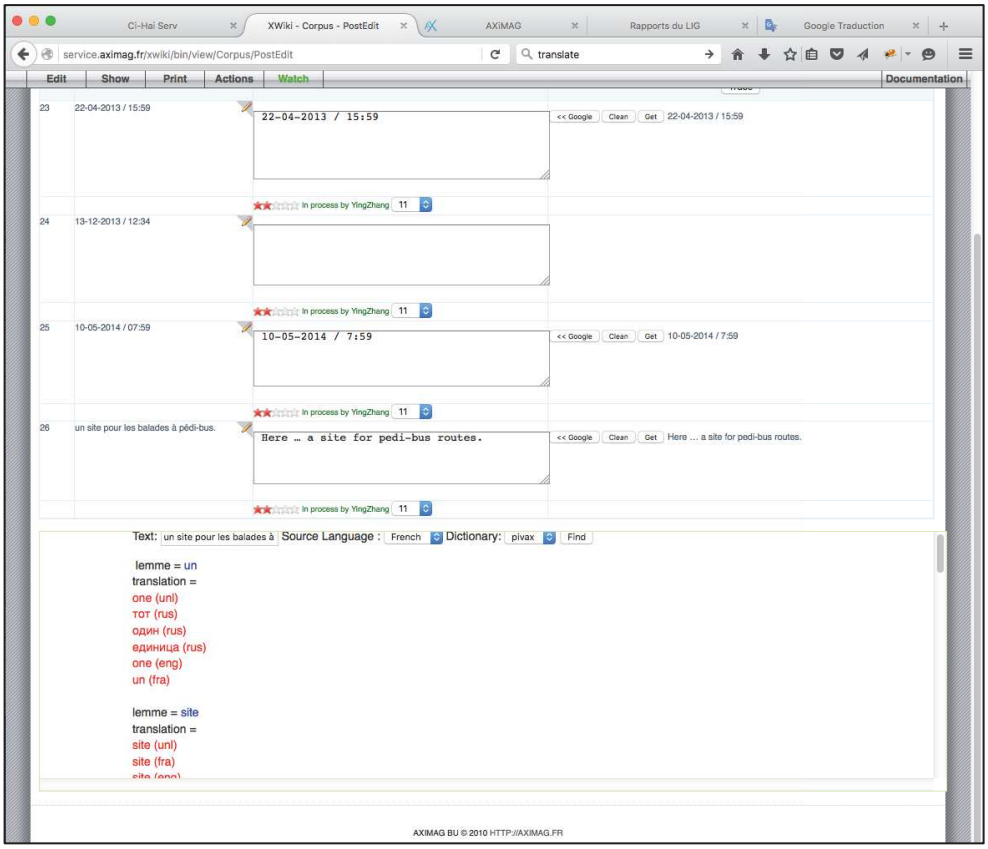


Figure 88 : Mini-dictionnaire (multicible) intégré dans SECTRA pour le segment "un site pour les balades à pédi-bus."

IV.3.3.2.2 Utilisation du formulaire par un linguiste (résultat de scénario 2)

a. *Résultat de l'expérience 1*

Nous avons déjà montré le résultat de l'expérience 1 (consultation d'un segment avec une langue cible produisant une sortie simple) dans la Figure 87.

b. *Résultat de l'expérience 2*

Le résultat de l'expérience 2 est présenté dans la Figure 89. Nous avons demandé toutes les langues cibles, soit 25 dans IATE, car la langue source est aussi une langue cible si on les demande toutes.

L'exemple montre qu'effectivement IATE donne deux réponses différentes pour "Hello world" en anglais, et des réponses dans 9 autres langues. Il n'en donne donc aucune dans 15 langues (pour cet exemple).

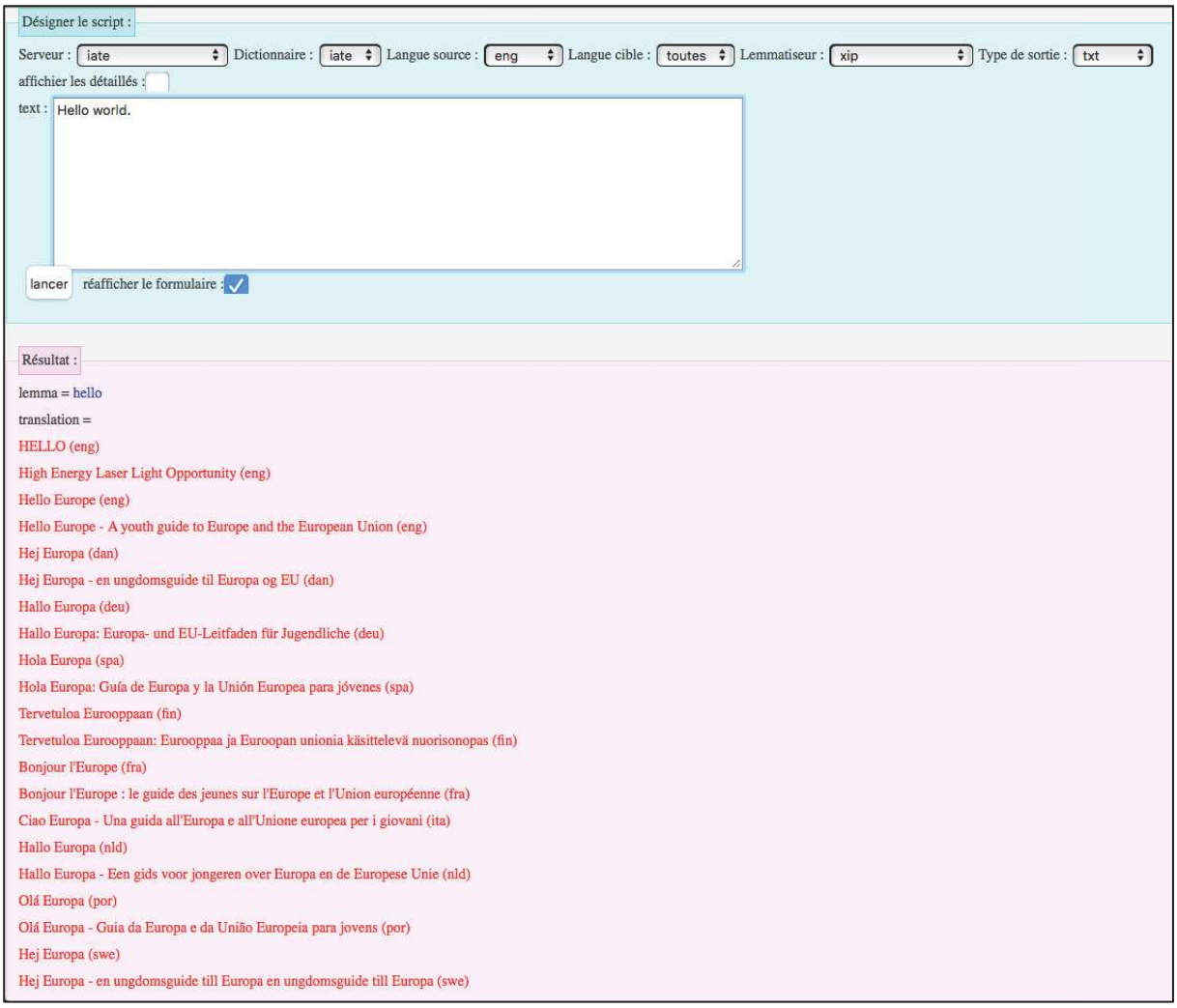


Figure 89 : Consultation d'un segment en "multicible", produisant une sortie simple

c. *Résultat de l'expérience 3*

Le résultat de l'expérience 3 est montré dans la Figure 90. Pour chaque traduction, la sortie "détaillée" en XML permet de décrire toutes les informations qu'on peut trouver, par exemple, la date de création, l'auteur, l'origine, les exemples, les définitions, les catégories grammaticales, les prononciations, etc.

C'est une sortie en XML, et uniformisée. Toutes les sorties dédiées sont produites à partir de cette sortie détaillée par des programmes simples (plugins). Il s'agit principalement du filtrage, du renommage des balises pour obtenir un nouveau mini-dictionnaire au format XML, de la transformation de XML vers d'autres formats (ex. JSON, TSV, ou une nouvelle structure XML, etc.).

Pour un segment, cette sortie "détaillée" pourrait être très longue. Notre interface permet d'ouvrir ou de fermer les nœuds XML de n'importe quel type en utilisant les boutons "+" ou "-" devant des balises.

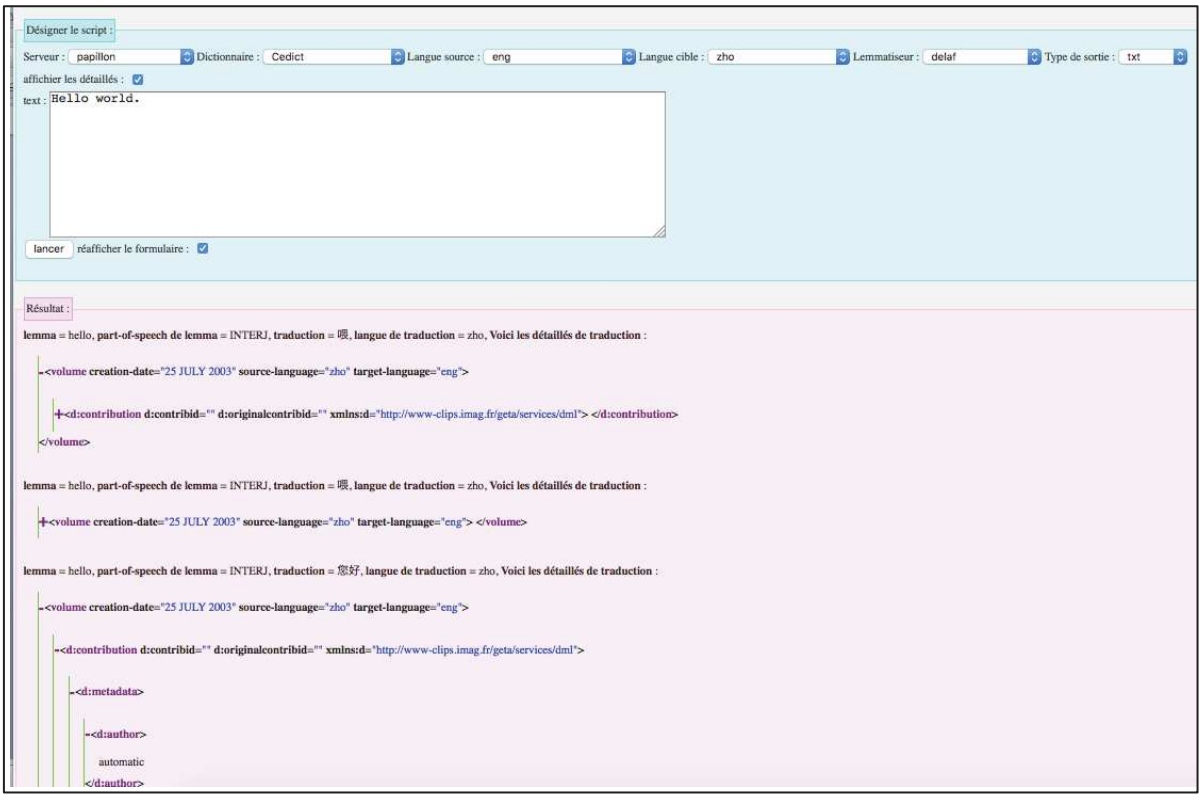


Figure 90 : Consultation d'un segment avec une seule langue cible, et sortie détaillée

IV.3.3.2.3 Configuration de CREATDICO et de plugin (résultat de scénario 3)

a. *Résultat d'expérience 1*

Après la désactivation de IATE, IATE n'est plus contenu dans la liste des serveurs de dictionnaires affichée sur l'interface. On ne peut plus l'utiliser. Si on demande les services sur l'API, le système envoie un message d'erreur (serveur/dictionnaire indisponible).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESSOURCES dateCreation = "20140222" dateDerniereModif = "20151207" Auteur = "Ying"
Accessible = "Public">
<!-- Services dictionnairiques-->
<SERVEURS>
<SERV codeServ = "Papillon" codeEnt = "utf8" callType="API" url = "http://www.papillon-
dictionary.org/papillon/api/$dico/$lang/cdm-headword/$headword/cdm-translation?strategy=EQUAL"
/>
<SERV codeServ = "Pivax" codeEnt = "utf8" callType="API" url =
"http://getalp.imag.fr/pivax/api/$dico/$lang/link/manger?strategy=EQUAL" />
<!-- Désactivation d'IATE, Test de configuration -- >
<!-- SERV codeServ = "IATE" codeEnt = "utf8" callType="EXEC" url = " http://iate.europa.eu"
requete="php -f Robodico.PHP LS=$lang LC=$lc lemme=$text"  /-->
……
</SERVEURS>
</RESSOURCES>
```

Figure 91 : Fichier de configuration des dictionnaires appelables

b. *Résultat d'expérience 2*

Après la modification du dictionnaire préféré pour les mini-dictionnaires de SECTRA, si aucun dictionnaire n'est spécifié dans la requête, CREATDICO prend le nouveau dictionnaire préféré.

```
<!-- configuration de Mini-dictionnaire de Sectra -- >
<MINIDICO name="sectra" dateCreation = "20140225" dateDerniereModif = "20160205" Auteur =
"Ying" Accessible = "Public">
<!-- lemmatiseur préféré pour chaque langue -->
<Lemmatisation ls = "eng" lemmat = "xip"/>
<Lemmatisation ls = "zho" lemmat = "jieba"/>
<Lemmatisation ls = "fra" lemmat = "delaf"/>
<Lemmatisation ls = "rus" lemmat = "ariane-heloise"/>
……
<!-- Service dictionnairique préféré par couple de langue -->
<Dico ls = "eng" lc = "zho" serv = "pivax" dico = "" />
<Dico ls = "zho" lc = "eng" serv = "papillon" dico = "Cedict" />
<Dico ls = "eng" lc = "fra" serv = "pivax" dico = "" />
<Dico ls = "fra" lc = "eng" serv = "pivax" dico = "" />
……
</MINIDICO>
```

Figure 92 : Fichier de configuration de plugin de mini-dictionnaires de SECTRA

Conclusions et perspectives

Conclusions

Dans le cadre de cette thèse, effectuée dans le cadre d'une bourse CIFRE, et prolongeant un des aspects du projet ANR TRAOUIERO, notre recherche a concerné la lexicographie computationnelle, non seulement pour les services lexicaux destinés à des humains, mais aussi pour les supports informatiques des programmes.

Nous avons d'abord étudié l'évolution des idées en lexicographie computationnelle de 1980 à 2012, depuis l'informatisation des dictionnaires classiques jusqu'aux plates-formes de construction de bases lexicales.

Pour nos travaux, nous sommes partie de PIVAX-1, développé par H.-T. Nguyen dans le cadre de sa thèse [Nguyen, H.-T., 2009]. Ce système a été construit en utilisant la plate-forme générale JIBIKI-1. Nous lui avons apporté des améliorations fonctionnelles et techniques.

D'une part, nous avons aidé M. Mangeot à améliorer la plate-forme sous-jacente JIBIKI pour supporter plusieurs fonctions de façon générique, par exemple, la gestion de "liens riches", le choix de(s) volume(s) cible(s), la recherche avec lemmatisation préalable, et la présentation par une liste non bornée.

D'autre part, nous avons réimplémenté PIVAX-1 sur la nouvelle version de JIBIKI-2. Nous avons transformé l'algorithme de calcul des liens, qui était dans la partie "code spécifique" de PIVAX-1, et l'avons inclues dans la partie "code générique" de PIVAX-2. Après ces améliorations, PIVAX-2 est devenu beaucoup plus rapide et plus stable que la première version. PIVAX-2 a permis de mettre réellement à disposition de la communauté les ressources lexicales rassemblées par le projet ANR TRAOUIERO : 270K entrées de COMMONUNLDICT de V. Dikonov, 9M entrées d'UWPEDIA de D. Rouquet, et 17K entrées d'ITOLDU de V. Bellynck.

La troisième partie de notre recherche a été motivée par un besoin industriel. Il s'agissait initialement de gestion des acronymes et d'autres types d'abréviations. Nous nous sommes beaucoup inspirée de la thèse de M. Tran [Tran, M. & Maurel, D., 2006] et des quatre variations du diasystème de Coseriu.

Nous avons traité non seulement les acronymes et les noms propres, mais aussi tous les types d'unités lexicales, plus ou moins "situés". Nous avons utilisé le terme "situement", proposé par Ch. Boitet, pour décrire les différents degrés "situés". Nous avons considéré quatre degrés de situement : général, situé, très situé, et terminologique. Avec la conception d'une base lexicale "métier", on peut organiser et gérer les unités lexicales de tous ces types dans une même base lexicale.

En ce qui concerne l'implémentation informatique, nous avons étendu la macrostructure de PIVAX en y intégrant des volumes de "prolexèmes" et un volume de "proaxies". Nous avons utilisé une étiquette libre dans les "liens riches" pour stocker les informations supplémentaires portées par les liens. Nous avons illustré le fonctionnement de notre prototype avec l'exemple *organisation des nations unies* en quatre langues. Nous avons prouvé les possibilités d'intégration pour l'utilisation dans un domaine linguistique spécifique et pour l'utilisation des quatre dimensions du diasystème. Pour présenter les résultats, nous avons introduit trois niveaux de précision d'une traduction, en théorie et en affichage.

Dans le dernier chapitre, nous avons présenté des services lexicaux généraux. Nous avons d'abord décrit la solution retenue, qui utilise ACTIVEMQ pour réaliser la gestion des tâches et des requêtes entre plusieurs serveurs. Ensuite, nous avons présenté le premier intergiciel de

lemmatisation, LEXTOH, qui permet d'appeler plusieurs services de lemmatisation, puis d'unifier et de filtrer leurs résultats. Enfin, nous avons créé un intergiciel de création de "mini-dictionnaires", CREATDICO, et l'avons validé en introduisant (en coopération avec L. X. Wang) dans SECTRA/IMAG une première version opérationnelle de la fonction d'aide lexicale proactive.

Perspectives

Les perspectives de cette recherche sont multiples. Nous les distinguons en court terme et en long terme.

Pour le court terme, il s'agit d'apporter des améliorations à LEXTOH et à CREATDICO. D'une part, l'ajout du pinyin pour le chinois dans LEXTOH est une demande formulée par les enseignants de chinois de l'UGA dans le cadre du projet INNOVALANGUES. Nous prévoyons d'intégrer l'analyseur syntaxique du chinois HANLP [HanLP, 2016] qui permet de transformer les phrases chinoises en pinyin. D'autre part, nous n'avons pas eu assez de temps pour implémenter tous les outils disponibles dans nos intergiciels. Par exemple, pour LEXTOH, on pourrait intégrer MORPHALOU, LEFFF, TREETAGGER, MECAB etc. Pour CREATDICO, on pourrait également appeler plusieurs serveurs lexicaux, comme GDEF, MOTÀMOT, etc.

Pour le long terme, nous avons trois perspectives.

D'abord, il y a le passage à l'échelle de PIVAX-3. Dans la base de notre prototype, il y a uniquement des exemples issus de PROLEXBASE et une toute petite partie des données d'acronymes (protégées) de Lingua et Machina. Nous prévoyons d'importer plusieurs ressources : la ressource complète de PROLEXBASE, la grosse base CJK, les listes d'abréviations en multilingue de WIKPEDIA[1] (il y a plusieurs listes, par exemple : abréviations en informatique, en médecine, abréviations militaires, etc.), et les ressources de DBNARY.

Ensuite, au milieu de cette thèse, nous avions prévu de développer une plate-forme, nommée CI-HAI SERV (Ci-Hai : océan de mot, c'est le nom d'un dictionnaire chinois très connu), pour fournir des services lexicaux à des projets. Nous avions prévu d'y intégrer LEXTOH, CREATDICO, et EXTROH. EXTROH est défini comme un intergiciel d'extraction et d'import de termes (en monolingue ou multilingue). Mais le temps nous a manqué pour finir cet intergiciel, et pour réaliser la gestion des projets, et la gestion des utilisateurs dans CI-HAI SERV.

La dernière perspective a été mentionnée au II.4.3.1. Il s'agit de remédier au manque de programmabilité dans PIVAX, car c'est un vrai problème pour construire, corriger et enrichir nos ressources lexicales. Il serait vraiment intéressant d'implémenter un outil pour manipuler les ressources lexicales en utilisant les idées de "langage narratif".

[1] https://fr.wikipedia.org/wiki/Abréviation

Bibliographie

[ISO 1087-1] Terminology work – Vocabulary – Part 1: Theory and application. International Organization for Standardization.

[Archer, V., 2009] Archer, V. (2009). Graphes linguistiques multiniveau pour l'extraction de connaissances : l'exemple des collocations. *Thèse de doctorat en informatique*, Université Joseph Fourier : Grenoble I, 274 p.

[Audibert, L., 2003] Audibert, L. (2003). Outils d'exploration de corpus et désambiguïsation lexicale automatique. *Thèse de doctorat en informatique*, Université d'Aix-Marseille 1 - Université de Provence, 360 p.

[Aymerich, J. & Camelo, H., 2009] Aymerich, J. and Camelo, H. (2009). The Machine Translation Maturity Model at PAHO. Proc. *IAMT*, Ontario, Canada., 7 p.

[Bachut, D. & Verastegui, N., 1984] Bachut, D. and Verastegui, N. (1984). Software tools for the environment of a computer-aided translation system. Proc. *COLING-84, ACL*, Stanford, pp. 330-340.

[Ball, S., 2003] Ball, S. (2003). Joined-up Terminology - The IATE system enters production. Proc. *the 25th International Conference on Translating and the Computer*, London, UK, 5 p.

[Bellynck, V., 2001] Bellynck, V. (2001). Un langage « narratif » pour Cabri-géomètre. *Hypermédias et Apprentissages 5*. Grenoble**:** pp. 344-346.

[Bellynck, V. et al., 2005] Bellynck, V., Boitet, C. and Kenwright, J. (2005). ITOLDU, a Web Service to Pool Technical Lexical Terms in a Learning Environment and Contribute to Multilingual Lexical Databases. Proc. *CICLing 2005*, Mexico City, pp. 324-332.

[Berment, V. & Boitet, C., 2012] Berment, V. and Boitet, C. (2012). Héloïse - An Ariane-G5 compatible environment for developing expert MT systems online. Proc. *COLING 2012*, Bombay, 7 p.

[Blanc, E., 1999] Blanc, E. (1999). PARAX-UNL : A large scale hypertextual multilingual lexical database. Proc. *5th Natural Language Processing Pacific Rim Symposium 1999*, Beijing, pp. 507-510.

[Boitet, C., 1990] Boitet, C. (1990). Software and lingware engineering in recent (1980—90) classical MT : Ariane-G5 and BV/aero/F-E. Proc. *ROCLing-III (tutorials)*, Taipei, 21 p.

[Boitet, C., 1996] Boitet, C. (1996). Synergies possibles entre outils pour traducteurs (THAM, mémoires de traduction) et systèmes de traduction automatique, avec les nouvelles possibilités introduites par Internet. Proc. *TAL+AI/NLP/IA*, Moncton, 12 p.

[Boitet, C., 2005] Boitet, C. (2005). What will it take for Papillon to be concretely useful not only for humans, but for machines? Proc. *SNLP-2005*, Bangkok, Thailand, 10 p.

[Boitet, C. et al., 2009] Boitet, C., Blanchon, H., Seligman, M. and Bellynck, V. (2009). Evolution of MT with the Web. Proc. *Machine Translation 25 Years On*, Cranfield, England, 13 p.

[Boitet, C. et al., 2007] Boitet, C., Boguslavskij, I. and Cardeñosa, I. (2007). An Evaluation of UNL Usability for High Quality Multilingualization and Projections for a Future UNL++ Language. Proc. *CICLING-2007*, Mexico City, pp. 361-373.

[Boitet, C. et al., 2002] Boitet, C., Mangeot, M. and Sérasset, G. (2002). The PAPILLON project : cooperatively building a multilingual lexical data-base to derive open source

dictionaries & lexicons. Proc. *the 2nd workshop on NLP and XML, Post-COLING 2002 workshop*, Taipei, 3 p.

[Boitet, C. & Nedobejkine, N., 1986] Boitet, C. and Nedobejkine, N. (1986). Towards integrated dictionaries for M(a)T : motivations and linguistic organization. Proc. *COLING 1986*, Bonn, pp. 423-428.

[Brace, C., 1994] Brace, C. (1994). Bonjour, Eurolang Optimizer. *Language Industry Monitor*, (Issue March-April), 3 p.

[Breen, J., 2004] Breen, J. (2004). JMDict : a Japanese multilingual dictionary. Proc. *the Workshop on Multilingual Linguistic Resources, COLING 2004*, Geneva, pp. 71-79.

[Brosseau-Villeneuve, B., 2010] Brosseau-Villeneuve, B. (2010). Désambiguïsation de Sens par modèles de contextes et son application à la Recherche d'Information. *Mémoire présenté à la Faculté des arts et des sciences en vue de l'obtention du grade de Maître ès sciences en informatique*, 101 p.

[Brown de Colstoun, F. et al., 2011] Brown de Colstoun, F., Delpech, E. and Monneret, E. (2011). Libellex : une plateforme multiservices pour la gestion des contenus multilingues. Proc. *TALN 2011, Démonstrations*, Montpellier, 1 p.

[Brown, J. S. et al., 1989] Brown, J. S., Collins, A. and Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher*, Vol. 18, pp. 32-42.

[Chakrabarti, D. et al., 2002] Chakrabarti, D., Narayan, D. K., Pandey, P. and Bhattacharyya, P. (2002). An Experience in Building the Indo WordNet - a WordNet for Hindi. Proc. *International Conference on Global WordNet (GWC 02)*, Mysore, India, 8 p.

[Chauché, J., 1975] Chauché, J. (1975). Les langages ATEF et CETA. *AJCL*, microfiche 17, pp. 21-39.

[Coseriu, E., 1992] Coseriu, E. (1992). Einführung in die allgemeine Sprachwissenschaft. Tübingen : Francke, pp. 262-264.

[Coseriu, E., 1998] Coseriu, E. (1998). Le double problème des unités dia-s. *Les Cahiers dia. Etudes sur la diachronie et la variation linguistique*, Vol. 1, pp. 9-16.

[Coseriu, E., 2001] Coseriu, E. (2001). L'homme et son langage. Louvain - Paris, Peeters, 492 p.

[Courtin, J. et al., 1992] Courtin, J., Dujardin, D. and Kowarski, I. (1992). PILAF: Software Tools for Lexicography and Text Research. Proc. *COMPLEX'92*, Budapest, Hungary, pp. 113-121.

[Courtin, J. & Genthial, D., 1998] Courtin, J. and Genthial, D. (1998). Parsing with Dependency Relations and Robust Parsing. Proc. *COLING'98 Workshop on Processing of Dependency-Based Grammars*, Montréal, Canada, pp. 95-101.

[Daoud, M., 2010] Daoud, M. (2010). Utilisation de ressources non conventionnelles et de méthodes contributives pour combler le fossé terminologique entre les langues en développant des "préterminologies" multilingues. *Thèse de doctorat en informatique*, Université de Grenoble, 192 p.

[Delpech, E., 2013] Delpech, E. (2013). Traduction assistée par ordinateur et corpus comparables : contributions à la traduction compositionnelle. *Thèse de doctorat en informatique*, Université de Nantes, 266 p.

[Doan-Nguyen, H., 1998a] Doan-Nguyen, H. (1998a). Accumulation of Lexical Sets: Acquisition of Dictionary Resources and Production of New Lexical Sets. Proc. *COLING-ACL 1998*, Canada, 5 p.

[Doan-Nguyen, H., 1998b] Doan-Nguyen, H. (1998b). Techniques génériques d'accumulation d'ensembles lexicaux structurés à partir de ressources dictionnairiques informatisées multilingues hétérogènes. *Thèse de doctorat en informatique*, Institut National Polytechnique de Grenoble, 168 p.

[Dolan, W. et al., 1993] Dolan, W., Vanderwende, L. and Richardson, S. D. (1993). Automatically deriving structured knowledge bases from on-line dictionaries. Proc. *First Conference of the Pacific Association for Computational Linguistics*, Vancouver, Canada, 10 p.

[Dolan, W. B. & Richardson, S. D., 1996] Dolan, W. B. and Richardson, S. D. (1996). Interactive Lexical Priming for Disambiguation. Proc. *MIDDIM'96, Post-COLING seminar on Interactive Disambiguation*, Le Col de Porte, pp. 54-56.

[Dong, D. Z., 1990] Dong, D. Z. (1990). TRANSTAR: a commercial English-Chinese MT system. . Proc. *ACL'90*, Pittsburgh, Pennsylvania, USA, pp. 339-341.

[Dong, D. Z., 2007] Dong, D. Z. (2007) China's MT in the 1970s and the 1980s - from revival to prosperity. 7 p.

[Dong, Z. et al., 2010] Dong, Z., Dong, Q. and Hao, C. (2010). Hownet and its computation of meaning. Proc. *COLING 2010 - Demonstrations*, Beijing, pp. 53-56.

[Fillmore, C. J. & Atkins, B. T. S., 1998] Fillmore, C. J. and Atkins, B. T. S. (1998). FrameNet and lexicographic relevance. Proc. *the First International Conference on Language Resources and Evaluation*, Granada, Spain, pp. 28-30.

[Forcada, M. L. et al., 2011] Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G. and Tyers, F. M. (2011). Apertium : a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2), pp. 127-144.

[Gadet, F., 2003] Gadet, F. (2003). La variation sociale en français. Paris, Ophrys, 135 pp.

[Gan, K.-W. et al., 2002] Gan, K.-W., Wang, C. Y. and Mak, B. (2002). Knowledge-based sense pruning using the HowNet : an alternative to word sense disambiguation. Proc. *the International Symposium of Chinese Spoken Language Processing*, Taiwan, pp. 189-192.

[Gaschler, J. & Lafourcade, M., 1994] Gaschler, J. and Lafourcade, M. (1994). Manipulating human-oriented dictionaries with very simple tools. Proc. *COLING 1994*, Kyoto, pp. 283-286.

[Grass, T., 2000] Grass, T. (2000). Typologie et traductibilité des noms propres de l'allemand vers le français. Proc. *TAL*, pp. 643–670.

[Grass, T. et al., 2004] Grass, T., Maurel, D. and Tran, M. (2004). Prolexbase : une ontologie pour le traitement multilingue des noms propres. *Linguistica Antverpiensia*, (3), pp. 293-309.

[Greeno, J. G. & Moore, J. L., 1993] Greeno, J. G. and Moore, J. L. (1993). Situativity and Symbols: Response to Vera and Simon. *Cognitive Science*, Vol. 17, pp. 49-59.

[Guillaume, P., 1989] Guillaume, P. (1989). Ariane-G5 - Les langages spécialisés TRACOMPL et EXPANS. Document interne GETA (Ariane-G5 version 3). 93 p.

[Gut, Y. et al., 1996] Gut, Y., Ramli, P. R. M., Yusoff, Z., Kim, C. C., Samat, S. A., Boitet, C., Nédobejkine, N., Lafourcade, M., Gaschler, J. and Levenbach, D. (1996). Kamus

Perancis-Melayu Dewan, dictionnaire français-malais. Kuala Lumpur, Dewan Bahasa Dan Pustaka, 667 p.

[Halpern, J., 2002] Halpern, J. (2002). Lexicon-based Orthographic Disambiguation in CJK Intelligent Information Retrieval. Proc. *3rd workshop on Asian language resources and international standardization, COLING 2002*, Taipei, 7 p.

[Halpern, J., 2006] Halpern, J. (2006). The Role of Lexical Resources in CJK Natural Language Processing. Proc. *the Workshop on Multilingual Language Resources and Interoperability*, Sydney, pp. 9-16.

[Hsu, Y.-L. U. & Su, K.-Y., 1997] Hsu, Y.-L. U. and Su, K.-Y. (1997). The Current Status and Prospects of BehaviorTran English-to-Chinese Machine Translation System. Proc. *the First R.O.C. International Conference on Translation*, Taipei, pp. 65-79.

[Huynh, C.-P., 2010] Huynh, C.-P. (2010). Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia. *Thèse de doctorat en informatique*, Université de Grenoble, Université de Danang pp. 229.

[Jensen, K., 1986] Jensen, K. (1986). PEG 1986: a Broad-coverage Computational Syntax of English. IBM research report. T.J. Watson research center, Yorktown Heights, N.Y.

[Jonasson, K., 1994] Jonasson, K. (1994). Le nom propre. Constructions et interprétations. Paris, Duculot, 255 p.

[Kalitvianski, R., 2013] Kalitvianski, R. (2013). De la multilinguïsation contributive de supports pédagogiques pour les OFI à la « segnormalisation » multiple et récursive de documents. *Rapport de stage de M2R*, 48 p.

[Lafourcade, M., 1996] Lafourcade, M. (1996). Serveurs de dictionnaires - Etude de cas avec l'outil Alex et le projet de dictionnaire Français-Anglais-Malais. Proc. *Séminaire Lexique - Représentation et Outils pour les Bases Lexicaless - Morphologie Robuste*, Grenoble, pp. 185-192.

[Lafourcade, M., 1997] Lafourcade, M. (1997). Multilingual dictionary construction and services : case study with the Fe projects. Proc. *PACLING 1997*, Tokyo, pp. 171-181.

[Lafourcade, M. & Chauché, J., 1998] Lafourcade, M. and Chauché, J. (1998). Ficus - un agent dictionnaire coopératif et extensible. Proc. *NLP+IA'98*, Moncton, New-Brunswick, Canada, 8 p.

[Lepage, Y., 2000] Lepage, Y. (2000). Languages of analogical strings. Proc. *COLING-2000*, Saarbrücken, pp. 488–494.

[Luong, N. Q., 2014] Luong, N. Q. (2014). Word Confidence Estimation and its Applications in Statistical Machine Translation. *Thèse de doctorat en informatique*, Université de Grenoble, 184 p.

[Mangeot, M., 2001] Mangeot, M. (2001). Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue. *Thèse de doctorat en informatique*, Université Joseph Fourier, 296 p.

[Mangeot, M., 2002] Mangeot, M. (2002). Projet Papillon : intégration de dictionnaires existants et gestion des contributions. Proc. *JST'02 Journées Science et Technologie*, Tokyo, pp. 64-65.

[Mangeot, M., 2006] Mangeot, M. (2006). Dictionary building with the Jibiki platform : software demonstration. Proc. *EURALEX 2006*, Torino, pp. 121-126.

[Mangeot, M., 2016] Mangeot, M. (2016). Collaborative construction of a good quality broad coverage and copyright free Japanese-French dictionary. *Hosei University International Found Foreign Scholar Fellowship Report*, Hosei University, Tokyo, Japan, Vol. XVI 2013-2014, pp. 175-208.

[Mangeot, M. et al., 2003] Mangeot, M., Bilac, S. and Thevenin, D. (2003). Construction collaborative d'un dictionnaire multilingue : le projet Papillon. Proc. *JSF'2003 Journées Scientifiques Francophones*, Tozeur, 3 p.

[Mangeot, M. & Chalvin, A., 2006] Mangeot, M. and Chalvin, A. (2006). Dictionary building with the Jibiki platform : the GDEF case. Proc. *LREC 2006*, Genoa (Gênes), pp. 1666-1669.

[Mangeot, M. & Nguyen, H. T., 2009] Mangeot, M. and Nguyen, H. T. (2009). Projet MotÀMot : élaboration d'un système lexical multilingue par le biais de dictionnaires bilingues. Proc. *journées scientifiques LTT*, Lisbonne, 12 p.

[Mangeot, M. & Sérasset, G., 2001] Mangeot, M. and Sérasset, G. (2001). Projet Papillon : architecture du serveur Web et structure des articles. Proc. *JST 2001 Journées Science et Technologie*, Tokyo, pp. 149-150.

[Mangeot, M. & Touch, S., 2010] Mangeot, M. and Touch, S. (2010). MotÀMot project : building a multilingual lexical system via bilingual dictionaries. Proc. *Second International Workshop on Spoken Languages Technologies for Under-Resourced Languages*, Penang, 6 p.

[McCord, M. C., 1989] McCord, M. C. (1989). Design of LMT : a Prolog-based Machine Translation System. *Computational Linguistics*, Vol. 15, Number 1, pp. 33-52.

[Mel'čuk, I. & Polguère, A., 2006] Mel'čuk, I. and Polguère, A. (2006). Dérivations sémantiques et collocations dans le DiCo/LAF. *Langue française, numéro spécial sur la collocation « Collocations, corpus, dictionnaires », sous la direction de P. Blumenthal et F. J. Hausmann*, pp. 66-83.

[Mel'čuk, I., 1992] Mel'čuk, I. (1992). DEC : Dictionnaire Explicatif et Combinatoire du français contemporain, recherches lexico-sémantiques III. Montréal, Canada, Université de Montréal, 332 p.

[Melby, A., 1989] Melby, A. (1989). New tools for terminology management on microcomputers. Proc. *Terminology Diachronique*, Paris, pp. 257-269.

[Miller, G. A. et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. J. (1990). Introduction to WordNet : An on-line lexical database. *International journal of lexicography*, 3(4), pp. 235-244.

[Mitamura, T. & Nyberg, E., 1992] Mitamura, T. and Nyberg, E. (1992). The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains. Proc. *Coling-92*, Nantes, pp. 1069-1073.

[Monteleone, M., 2003] Monteleone, M. (2003). Lexicographie et dictionnaires électroniques. Des usages linguistiques aux bases de données lexicales. *Thèse*, Université de Marne-la-Vallée, 198 p.

[Morardo, M. & Villemonte de La Clergerie, É., 2013] Morardo, M. and Villemonte de La Clergerie, É. (2013). Vers un environnement de production et de validation de ressources lexicales sémantiques. Proc. *TALN*, Les Sables d'Olonne, 14 p.

[Murata, T. et al., 2003] Murata, T., Kitamura, M., Fukui, T. and Sukehiro, T. (2003). Implementation of Collaborative Translation Environment 'Yakushite Net'. Proc. *MT Summit IX*, New Orleans, pp. 479-482.

[Navigli, R. & Ponzetto, S. P., 2010] Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a Very Large Multilingual Semantic Network. Proc. *Artificial Intelligence*, Elsevier, pp. 217-250.

[Nguyen, H.-T., 2009] Nguyen, H.-T. (2009). Des systèmes de TA homogènes aux systèmes de TAO hétérogènes. *Thèse de doctorat en informatique*, Université Joseph-Fourier - Grenoble I, 236 p.

[Nguyen, H.-T. et al., 2007] Nguyen, H.-T., Boitet, C. and Sérasset, G. (2007). PIVAX, an online contributive lexical data base for heterogeneous MT systems using a lexical pivot. Proc. *SNLP-2007*, Bangkok, 6 p.

[Nguyen, H. T. & Boitet, C., 2009] Nguyen, H. T. and Boitet, C. (2009). Lexical synergy between MT & Translator Aids : PIVAX, a generic online contributive lexical database platform. Proc. *International Conference "Machine Translation 25 Years On"*, Cranfield, 8 p.

[Nirenburg, S. & Defrise, C., 1990] Nirenburg, S. and Defrise, C. (1990). Lexical and Conceptual Structure for Knowledge-Based Machine Translation. Proc. *ROCLING III*, Taipeh, pp. 105-130.

[Noy, N. F. & McGuinness, D. L., 2003] Noy, N. F. and McGuinness, D. L. (2003). Ontologie pour le Web sémantique. *Web sémantique, rapport final de l'Action spécifique 32 CNRS/STIC*.

[Perscheid, M. M., 1985] Perscheid, M. M. (1985). Computer-Aided Translation at WCC. *CALIC Journal (The Computer Assisted Language Instruction Consortium)*, Vol 3, No. 1, pp. 22-24.

[Planas, E., 2005] Planas, E. (2005). SIMILIS second-generation translation memory software. Translating and the Computer 27. Londres.

[Ramisch, C., 2012] Ramisch, C. (2012). A generic and open framework for multiword expressions treatment: from acquisition to applications. *Thèse de doctorat en informatique*, Universit´e de Grenoble; Universidade Federal do Rio Grande do Sul, 234 p.

[Richardson, S. D. et al., 1998] Richardson, S. D., Dolan, W. B. and Vanderwende, L. (1998). MindNet : acquiring and structuring semantic information from text. Proc. *ACL 1998*, Stroudsburg, pp. 1098-1102.

[Roche, C., 2007] Roche, C. (2007). Le terme et le concept : fondements d'une ontoterminologie. Proc. *TOTh*, Annecy, 13 p.

[Rouquet, D. et al., 2013] Rouquet, D., Falaise, A., Boitet, C. and Bellynck, V. (2013). OMNIA : extracteur de contenu semantique multilingue (Rapport interne dans le cadre du projet Traouïero), 31 p.

[Rouquet, D. & Nguyen, H. T., 2009] Rouquet, D. and Nguyen, H. T. (2009). Interlingual annotation of texts in the OMNIA project. Proc. *LTC 2009*, Poznań, Poland, 5 p.

[Schwab, D., 2005] Schwab, D. (2005). Approche hybride-lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte. *Thèse de doctorat en informatique*, Université Montpellier II, 390 p.

[Schwab, D., 2013] Schwab, D. (2013). Tableau_blanc-iMAG, agents, tâches en boucles infinies. Projet Traouiéro, document L3.2, 10 p.

[Sérasset, G., 1994a] Sérasset, G. (1994a). Interlingual lexical organisation for multilingual lexical databases in NADIA. Proc. *COLING 1994*, Kyoto, pp. 278-282.

[Sérasset, G., 1994b] Sérasset, G. (1994b). SUBLIM : un système universel de bases lexicales multilingues et NADIA : sa spécialisation aux bases lexicales interlingues par acceptions. *Thèse de doctorat en informatique*, Université Joseph-Fourier-Grenoble I, 206 p.

[Sérasset, G., 1996] Sérasset, G. (1996). Un éditeur pour le dictionnaire explicatif et combinatoire du français contemporain. Proc. *Journées lexique du PRC-CHM*, Grenoble, pp. 131-138.

[Sérasset, G., 2004] Sérasset, G. (2004). A generic collaborative platform for multilingual lexical database development. Proc. *the Workshop on Multilingual Linguistic Resources, COLING 2004*, Geneva, pp. 79-86.

[Sérasset, G., 2008] Sérasset, G. (2008). La plate-forme" Jibiki" dans le projet LexALP. Proc. *Normazione, armonizzazione e pianificazione linguistica; Normierung, Harmonisierung und Sprachplanung; Normalisation, harmonisation et planification linguistique*, Bolzano, pp. 31-47.

[Sérasset, G., 2012] Sérasset, G. (2012). Dbnary : Wiktionary as a LMF-based Multilingual RDF network. Proc. *LREC 2012*, Istanbul, pp. 2466-2472.

[Sérasset, G. et al., 2006] Sérasset, G., Brunet-Manquat, F. and Chiocchetti, E. (2006). Multilingual legal terminology on the Jibiki platform : the Lexalp project. Proc. *COLING/ACL 2006*, pp. 937-944.

[Sérasset, G. & Tchechmedjiev, A., 2014] Sérasset, G. and Tchechmedjiev, A. (2014). Dbnary : Wiktionary as Linked Data for 12 Language Editions with Enhanced Translation Relations. Proc. *3rd Workshop on Linked Data in Linguistics, LREC 2014*, Reykjavik, 4 p.

[Seretan, V., 2008] Seretan, V. (2008). Collocation extraction based on syntactic parsing. *Thèse de doctorat en informatique*, University of Geneva, 249 p.

[Shimohata, S. et al., 1999] Shimohata, S., Murata, T., Ikeno, A., Fukui, T. and Yamamoto, H. (1999). Machine Translation System PENSÉE : System Design and Implementation. Proc. *MT Summit VII*, Singapore, pp. 380-384.

[Takebayashi, Y., 1993] Takebayashi, Y. (1993). EDR Electronic Dictionary. Proc. *MT Summit IV*, Kobe, Japan, pp. 117-126.

[Teeraparbseree, A., 2005] Teeraparbseree, A. (2005). Méthodes et outils pour la création automatique et l'évaluation de structures de bases lexicales multilingues (symétriques) à lexies et axies. *Thèse de doctorat en informatique*, Université Joseph Fourier : Grenoble I, 166 p.

[Tomokiyo, M. et al., 2000] Tomokiyo, M., Mangeot, M. and Planas, E. (2000). Papillon : a Project of Lexical Database for English, French and Japanese, using Interlingual Links. Proc. *Journées Science et Technologie (JST-2000)*, Tokyo, 3 p.

[Tran, M., 2006] Tran, M. (2006). Prolexbase : un dictionnaire relationnel multilingue de noms propres : conception, implémentation et gestion en ligne. *Thèse de doctorat en informatique*, Université de Tours, 171 p.

[Tran, M. & Maurel, D., 2006] Tran, M. and Maurel, D. (2006). Prolexbase : un dictionnaire relationnel multilingue de noms propres. *Traitement automatique des langues*, 47(3), pp. 115-139.

[Vasconcellos, M. & León, M., 1985] Vasconcellos, M. and León, M. (1985). SPANAM and ENGSPAN: Machine translation at the Pan American Health Organization. *Computational Linguistics*, Vol. 11 No. 2-3, pp. 123-136.

[Villemonte de la Clergerie, É. et al., 2009] Villemonte de la Clergerie, É., Sagot, B., Nicolas, L. and Guénot, M.-L. (2009). FRMG: évolutions d'un analyseur syntaxique TAG du français. Proc. *Journée de l'ATALA sur : Quels analyseurs syntaxiques pour le français ?*, Paris, 10 p.

[Vo-Trung, H., 2004] Vo-Trung, H. (2004). Méthodes et outils pour utilisateurs, développeurs et traducteurs de logiciels en contexte multilingue. *Thèse de doctorat en informatique*, INPG, GETA, Grenoble, 224 p.

[Vossen, P., 1998] Vossen, P. (1998). EuroWordNet : a multilingual database with lexical semantic networks. Dordrecht, Kluwer Academic Publishers, 3 p.

[Wang, L. X., 2015] Wang, L. X. (2015). Outils et environnements pour l'amélioration incrémentale, la post-édition contributive et l'évaluation continue de systèmes de TA. Application à la TA français-chinois. *Thèse de doctorat en informatique*, Université Grenoble Alpes, 223 p.

[Wehrli, E., 2007] Wehrli, E. (2007). FIPS, a Deep Linguistic Multilingual Parser. Proc. *ACL 2007 Workshop on Deep Linguistic Parsing*, Athènes, pp. 90-94.

[Zhang, Y. & Mangeot, M., 2013] Zhang, Y. and Mangeot, M. (2013). Bases lexicales multilingues : traitement des acronymes. Proc. *JCLTT 2013*, Bruxelles, 16 p.

[Zhang, Y. et al., 2014] Zhang, Y., Mangeot, M., Bellynck, V. and Boitet, C. (2014). Jibiki-LINKS : a Tool between Traditional Dictionaries and Lexical Networks for Modelling Lexical Resources. Proc. *Workshop on Cognitive Aspects of the Lexicon (CogALex), COLING 2014*, Dublin, pp. 87-98.

Netographie

[ActiveMQ, 2015] ActiveMQ. Intergiciel à messages utilisable pour le développement d'une architecture MOM http://activemq.apache.org

[Apertium, 2015] Apertium. A free/open-source machine translation platform. https://www.apertium.org

[CJK, 2015] CJK. The CJK Dictionary. http://www.cjk.org/cjk/index.htm

[FeM, 2015] FeM. Dictionnaire général : français-anglais-malais (versions statiques). http://www.lirmm.fr/~lafourcade/HTMFEM_HTMFEMCS/HTMFEMCS/FEMCS.HTM

[HanLP, 2016] HanLP. Analyseur du chinois, visité en 2016. https://github.com/hankcs/HanLP

[IATE, 2015] IATE. InterActive Terminology for Europe. http://iate.europa.eu

[ISOcat, 2008] ISOcat. ISOcat - a Data Category Registry. 2015, http://www.isocat.org/rest/dcs/119

[Jieba, 2015] Jieba. Chinese text segmentation. https://github.com/fxsjy/jieba

[JobCenter, 2015] JobCenter. A client-server application and framework for job management and distributed job execution. 2015, https://github.com/yeastrc/jobcenter

[Lingua&Machina, 2015] Lingua&Machina. L'écrit multilingue dans l'entreprise. http://www.lingua-et-machina.com

[Metricc, 2015] Metricc. Projet Metricc. http://www.metricc.com

[OKI, 2015] OKI. Oki Electric Industry. http://www.oki.com

[OneLook, 2016] OneLook. Liste de dictioinnaires consultables, visité en 2016. http://www.onelook.com/?d=all_&v=&sort=&langdf=all

[PAHOMTS, 2015] PAHOMTS. Machine translation at the Pan American Health Organization. http://www1.paho.org/ENGLISH/AM/GSP/TR/MACHINE_trans.htm

[Papillon, 2015] Papillon. Le projet Papillon. http://www.papillon-dictionary.org/papillon/Home.po

[ProMT, 2015] ProMT. Logiciels de traduction et dictionnaires PROMT en ligne. http://www.promt.com

[Reverso, 2015] Reverso. Traduction et dictionnaire en ligne gratuit. http://www.reverso.net

[TM/2, 1992] TM/2. Language Industry Monitor. http://www.mt-archive.info/LIM-1992-11-5.pdf

[UIMA, 2015] UIMA. Apache UIMA project. https://uima.apache.org

[wAlex, 2015] wAlex. Serveur du dictionnaire français-malais (version dynamique). http://www.lirmm.fr/~lafourcade/index2.html

[XeLDA, 2015] XeLDA. Xerox Linguistic Development Architecture. http://www.xrce.xerox.com/About-XRCE/History/Historical-projects/XeLDA

Table des définitions

Définition 1. Les **informations lexicales formelles** sont définies de manière "exacte" (ex. respecter certaines règles syntaxiques), et sont compréhensibles/lisibles par les machines.

Définition 2. Au contraire des informations lexicales formelles, les **informations lexicales naturelles** sont définies de manière usuelle, en langue naturelle, et sont orientées vers les humains.

Définition 3. L'**indexage** d'une entrée source vers des entrées cible consiste à associer aux traductions possibles des conditions sur l'occurrence de l'entrée dans le contexte syntaxique ou sémantique permettant de les choisir.

Définition 4. La **terminologie** [ISO 1087-1] est définie comme "l'étude scientifique des notions et des termes en usage dans les langues de spécialité".

Définition 5. Une **ontologie** (informatique) est un ensemble structuré de termes et de concepts représentant les sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou par les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts.

Définition 6. Une **ontoterminologie** [Roche, C., 2007] est une terminologie dont le système notionnel est une ontologie formelle.

Définition 7. Un **vocable** est la forme de citation d'un lexème, accompagnée de sa classe morphosyntaxique. La **nomenclature** d'un dictionnaire est la liste de ses vocables. On appelle (micro-)structure du dictionnaire la structure générique de ses articles.

Définition 7. Un **dictionnaire de traduction** est composé d'un ou plusieurs *volumes,* chacun relatif à une des langues considérées.

Définition 8. Un **volume** d'un dictionnaire est un ensemble d'articles, accédés par des mots-vedettes (lemmes en général), décrivant des mots d'une même langue. Un article d'un dictionnaire de traduction contient les traductions du mot identifié dans la ou les langues cibles considérées.

Définition 9. Un **article de dictionnaire** comporte au moins le mot-vedette, et le plus souvent d'autres informations (prononciation, classe grammaticale, définition, gloses identifiant des sens, ou *lexies,* exemples, etc.), ainsi que des traductions s'il s'agit d'un dictionnaire de traduction.

Définition 10. Une **lexie** est un sens de mot dans un dictionnaire.

Définition 11. Une **axie** est une classe d'équivalence de lexies synonymes.

Définition 12. Une **acception** est un ensemble de lexies synonymes. On appelle **axème** une acception monolingue, et **axie** une acception interlingue (pouvant regrouper des lexies de différentes langues).

Définition 13. On appelle "**espace lexical**" d'une langue l'ensemble structuré de ses unités, à des niveaux de plus en plus abstraits ou génériques : formes, lemmes, racines (dans certaines langues) familles dérivationnelles plus ou moins productives, prolexèmes (réunissant par exemple *US*, *USA* et *Etats-Unis*), et "acception" ou "sens de mots"... sans oublier qu'il y a à tous ces niveaux des unités simples et des unités complexes (mots composés, lexies ou acceptions complexes).

Définition 14. La **macrostructure** d'une BDLex est la description de son architecture générale, c'est à dire des types de ses volumes et de leurs relations.

Définition 15. La **microstructure** d'un dictionnaire est la structure de ses articles, c'est à dire l'organisation de ses entrées.

Définition 16. **CDM** (Common Dictionary Markup) est une DTD (extensible) définissant un ensemble de balises XML correspondant aux types d'information des dictionnaires en source ouvert existants. Un **pointeur** CDM associé à une balise CDM `<bbb>` et à un dictionnaire `Dic` est un chemin XPATH permettant d'accéder à l'information correspondant à `<bbb>` dans `Dic`.

Définition 17. Dans notre modèle (PROLEXBASE), le **prolexème** correspond à une projection du nom propre conceptuel dans une langue donnée.

Définition 18. Nous appellerons "**situement**" la qualité d'être situé, et parlerons du "degré de situement" d'un lexème.

Définition 19. Chaque **type de vocable** correspond à un degré de situement : général, terminologique, situé, ou très situé.

Définition 20. Une "**BDLex métier**" est une BDLex qui couvre tous les types de vocable dans une même base pour un certain domaine.

Définition 21. **Nom propre** [Jonasson, K., 1994] : Toute expression associée dans la mémoire à long terme à une entité particulière en vertu d'un lien dénominatif conventionnel stable.

Définition 22. ***Prolexème.*** Dans une BDLex PIVAX-3, il y a un seul volume de prolexèmes pour chaque langue. Dans ce volume, les prolexèmes regroupent les lexies qui représentent le même sens mais dont la réalisation syntaxique est différente (forme de surface, classe grammaticale, etc.).

Définition 23. ***Proaxie.*** Il y a un seul volume de proaxies dans une instance de PIVAX-3. Les proaxies regroupent les prolexèmes de langues différentes partageant un même sens.

Définition 25. Un **mini-dictionnaire** contient les informations lexicales associés aux mots d'un fragment de texte, le plus souvent un segment. (Ces informations contiennent souvent les traductions dans une ou plusieurs langues.)

Table des "idées-guides"

Idée guide 1 Établir et maintenir des correspondances étroites entre informations lexicales formelles (codées) et "naturelles" (exprimées en termes usuels).

Idée guide 2 Pour les dictionnaires, la plus grande difficulté est l'incohérence entre les dictionnaires pour traducteurs, les dictionnaires pour humains, et les dictionnaires pour la TA.

Idée guide 3. Sans bonne liaison, ou mieux intégration, entre informations lexicales formelles et naturelles, elles ne peuvent pas rester cohérentes, et les unes ou les autres deviennent de moins en moins utiles dans les applications impliquant une synergie homme-machine.

Idée guide 4. L'unification des informations lexicales générales et des terminologies est un besoin réel pour les systèmes de TA et THAM, et a été réalisée dès 1992 par le système opérationnel KANT/CATALYST basé sur KBMT-89.

Idée guide 5. La non-symétrie des dictionnaires de traduction classiques est une difficulté majeure.

Idée guide 6. Une base lexicale contributive ne peut bien fonctionner que si (1) on peut motiver les contributeurs, et (2) un ou des animateurs (non-informaticiens) peuvent définir de nouvelles tâches et organiser puis contrôler eux-mêmes le fonctionnement de l'outil.

Annexes

Annexe 1 Liste d'une partie des ressources lexicales

Voici une liste des ressources lexicales informatisées, téléchargeables et gratuites, trouvées sur Internet.

Nom	Brève description	Nbr d'entrées	Langues	URL
APERTIUM	218 dictionnaires monolingues en 93 langues différentes, 236 dictionnaires bilingues en 235 couples de langues différents	Plus de 3,8M pour dictionnaires monolingues. Plus de 2,5M pour dictionnaires bilingues		http://wiki.apertium.org/wiki/List_of_dictionaries https://sourceforge.net/projects/apertium/files/
CC-CEDICT	Un dictionnaire de zho→eng.	114K	zho→eng	https://www.mdbg.net/chindict/chindict.php?page=cedict
CFDICT	Un dictionnaire de zho↔fra	6240	zho↔fra	http://www.chine-informations.com/mandarin/open/CFDICT/
DELAF	Lexique des formes fléchies du français et de l'anglais.	Dictionnaire du français : 683K entrées simples pour 102K lemmes différents et 108K entrées composées pour 83K lemmes différents. Dictionnaire de l'anglais : 296K entrées simples pour 150K lemmes différents et 133K entrées composées pour 70K lemmes différents	eng, fra	http://infolingu.univ-mlv.fr/DonneesLinguistiques/Dictionnaires/telechargement.html
DILAF	Base lexicale contributive conllaborative pour les dictionnaires de langues africaines vers le français.	bam : 10 800 hau : 7238 kan : 6000 thm : 6100 dje : 6914	bam→fra, hau→fra, kan→fra, thm→fra, dje→fra	http://www.dilaf.org/dilaf/Home.po
GLÀFF	Un lexique du français à large couverture extrait du Wiktionnaire, le dictionnaire collaboratif en ligne	2M	fra	http://redac.univ-tlse2.fr/lexiques/glaff.html
IATE	Une grande base terminologique.	8M	25 langues européennes	http://iate.europa.eu/tbxPageDownload.do
JIBIKI.FR	Base lexicale contributive collaborative pour dictionnaire fra-jpn. Trois ressources disponibles : Cesselin, JMdict, Kanjdic	153 897	jpn, fra, eng	http://jibiki.fr/data/
JMDICT	Dictionnaire japonais-anglais et autres langues par Jim Breen	173K	jpn→ eng,fra,esp, por,deu,rus,	http://www.edrdg.org/jmdict/j_jmdict.html

			nld	
KAIFANGCIDIAN	Quatre dictionnaires : zho→nor, nor→zho, zho→epo, epo→zho	zho→nor : 2544 nor→zho : 3469 zho→epo : 6313 epo→zho : 5492	zho↔nor, zho↔epo	http://kaifangcidian.com/xiazai/nuo/ http://kaifangcidian.com/xiazai/shi/
LEFFF	Lexique des formes fléchies du français.	Nombre de formes : 404 634 Nombre de lemmes : 105 595	fra	http://www.labri.fr/perso/clement/lefff/
LITTRE	Dictionnaire monolingue français	136K	fra	http://www.littre.org/faq
MORPHALOU	Lexique des formes fléchies du français.	Nombre de formes : 524 725 Nombre de lemmes : 95 810	fra	http://www.cnrtl.fr/lexiques/morphalou/
MOTÀMOT	Base lexicale contributive conllaborative pour un dictionnaire fra↔khm.	13 249 articles français avec traductions en khmer. 23 766 articles khmer avec traductions en français. 32 402 liens bilingues français-khmer.	fra↔khm	http://jibiki.univ-savoie.fr/motamot/Download.po;jsessionid=ivcrZwUlAQHW-L9LomhO0jJN
PROLMF	Ressource de Prolexbase. Lexique relationnel multilingue de noms propres	Nombre de pivots : 73 365 Nombre de relations : 72 604 (3 908 accessibilités, 67 417 méronymies et 1 279 synonymies) Nombre d'entrées lexicales en français : 96 800 (75 501 noms propres et 21 299 dérivés, soit 147 518 formes) Nombre d'entrées lexicales en anglais : 19 449 (19 355 noms propres et 94 dérivés, soit 19 449 formes) Nombre d'entrées lexicales en polonais : 39 020 (35 937 noms propres et 3 083 dérivés, soit 165 309 formes)	10 langues (principalement fra, eng, pol)	http://www.cnrtl.fr/lexiques/prolex/
WADOKU	Dictionnaire japonais-allemand par Ulrich Apel	280K	jpn→deu	http://www.wadoku.de/wiki/display/WAD/Downloads+und+Links
WEBSTER'S	Dictionnaire monolingue anglais Webster's de 1913	182 700	eng	http://www.gutenberg.org/ebooks/29765
WIKTIONARY	Le dictionnaire multilingue contributif le plus connu au monde.	4 683 963	Plus de 100 langues	http://dumps.wikimedia.org/backup-index.html
WORDNET	Grande base de données lexicale pour l'anglais.	147 278	eng	https://wordnet.princeton.edu/wordnet/download/current-version/

Annexe 2 Exemple de diversité des UW dans le projet UNL

Exemple du mot "*access*" dans différents dictionnaires d'UW.

U++C

access(icl>recover>do, agt>thing, obj>thing);
Cat : VERB;
Déf . : obtain or retrieve from a storage device; as of information on a computer;

access(icl>reach>do, agt>thing, obj>thing);
Cat. : VERB;
Déf. : reach or gain access to;
Ex. :
"How does one access the attic in this house?";
"I cannot get to the T.V. antenna, even if I climb on the roof";

UNLKB

right(icl>abstract thing)
access(icl>right)
way(icl>abstract thing)
abandon(icl>way)
access(icl>way)
opportunity(icl>time)
access(icl>opportunity)
market(icl>opportunity)
enter(agt>person,obj>thing)
access(icl>enter(agt>person,obj>thing))
barge in(agt>person,obj>thing)
open(agt>thing,obj>thing)
access(icl>open(agt>thing,obj>thing))
reach(agt>thing,obj>thing)
access(icl>reach(agt>thing,obj>thing)
use(agt>thing,obj>thing)
access(icl>use(agt>thing,obj>thing))
build(icl>use(agt>thing,obj>thing))
drive(icl>use(agt>thing,obj>thing))

UNLDeco/UNL-FR

[accéder] {CAT(CATV),AUX(AVOIR),VAL1(GN),GP1(A)} **"access(agt>thing,obj>thing)"**;

EOLSS/UNL-FR

access(icl>use(agt>thing,obj>thing))

Annexe 3 Historique du projet PAPILLON

(Ce qui suit est un résumé par Ch. Boitet de diverses présentations).

Le projet PAPILLON [Tomokiyo, M. et al., 2000] a été lancé en juillet 2000 par Emmanuel Planas, François Brown de Colstoun et Mutsuko Tomokiyo[1], avec le concours de F. Andrès, U. Apel, V. Bellynck, Ch. Boitet, J. Breen, M. Mangeot, G. Sérasset, M. Zock, K. Kageura, M. Lafourcade, etc. La motivation principale du projet était qu'il y a très peu de ressources dictionnairiques français↔japonais, et que ces ressources ne sont pas informatisées. Les dictionnaires disponibles sous forme papier sont chers et difficiles à utiliser par les apprenants, à cause du manque de transcription phonétique et d'autres informations cruciales pour les étrangers, comme les classificateurs et quantificateurs. Le but du projet était de créer un environnement de développement coopératif à travers Internet et de créer une grande base lexicale multilingue. Il s'agissait initialement du français, du japonais et de l'anglais.

Les premières années du projet, de 2000 à 2006, il y eut un séminaire par an. Le séminaire de lancement du projet Papillon eut lieu en août 2000 au NII, à Tokyo. Il fut décidé de développer non pas une, mais deux bases lexicales multilingues, PAPILLON-CDM et PAPILLON-NADIA, décrites au I.1.3.2.3. Le second séminaire (juillet 2001, Grenoble) fut surtout consacré à des décisions techniques et organisationnelles, au vu du fonctionnement durant la première année. Le séminaire de 2002 fut organisé en juillet à Tokyo. Il y fut décidé que toutes les ressources de ce projet seraient mises en "source ouvert" (*open source),* sous licence Creative Commons. On discuta aussi des ressources financières et des façons de construire la base. En juillet 2003, le séminaire fut accolé à ACL-2003, à Sapporo. On y discuta de la plate-forme logicielle utilisée pour la construction de la base lexicale. Le séminaire de 2004 fut organisé en août à Grenoble sous la forme d'un post-workshop (Multilingual Linguistic Ressources, MLR2004) de COLING-2004, qui avait lieu à Genève. En 2005, le séminaire fut organisé en décembre à Chiang Rai, Thailande. Le dernier eut lieu en juillet 2006 à Bangkok, en liaison avec celui du STIC-Asie-NLP, juste avant COLING-2006.

Depuis 2006, la base lexicale est maintenue en service par M. Mangeot et est utilisée pour de la consultation, et de temps en temps pour de la contribution. Contrairement aux attentes initiales, il n'y a pas de contributions "isolées" par des contributeurs opportunistes. Par contre, quand quelqu'un ou un groupe a produit un dictionnaire intéressant, il lui est facile de le faire intégrer la base.

Il faut en particulier noter le projet GDEF (Grand Dictionnaire Estonien Français) [Mangeot, M. & Chalvin, A., 2006], supporté depuis plus de 10 ans par une instance particulière de PAPILLON. Un petit groupe de Français et d'Estoniens y travaille tous les jours.

[1] Alors respectivement chercheur postdoctoral au centre de recherche de NTT à Keihanna, attaché scientifique à l'ambassade de France à Tokyo, et doctorante en linguistique computationnelle à Paris VI mais travaillant en fait au GETA-CLIPS à Grenoble.

Annexe 4 Questions et réponses sur HOWNET[1]

Ce qui suit est tiré d'échanges passionnants avec le Pr. Dong Zhen Dong (en chinois), qui a créé et continue à augmenter HOWNET avec quelques collaborateurs, depuis 1988.

Aupararant, il avait participé au projet KY-1, un système TA zh↔en, développé par l'académie des sciences militaires entre 1982 et 1987. En 1988, ce système a été renommé TRANSTAR, le premier système commercial de TA en Chine [Dong, D. Z., 1990 ; Dong, D. Z., 2007].

1. Is HowNet a WordNet-like lexical database?

No. HowNet is not a WordNet-like lexical database. HowNet is completely different from WordNet both in its philosophy and implementation. WordNet is an online lexical database, while HowNet is a common-sense knowledge system with concepts as its basis. HowNet is purely computer-oriented.

2. What is HowNet most prominent feature?

HowNet most prominent feature is its unique way to define the concepts denoted by the lexical units in its basic database.

For example,

In HowNet:	*cf. WordNet:*
buy 1. DEF={buy|买} 2. DEF={GiveAsGift|赠 :manner={guilty|有罪 },purpose={entice|勾引}}	**buy** 1. (594) buy, purchase — (obtain by purchase; acquire by means of a financial transaction;) 2. (7) bribe, corrupt, buy, grease one's palms — (make illegal payments to in exchange for favors or influence;)
Buyer 1. {human|人:domain={commerce|商业 },{buy|买:agent={~}}}	**Buyer** 1. buyer, purchaser, emptor, vendee — (a person who buys)

HowNet concepts or senses are defined in a structured language called Knowledge Data Mark-up Language (KDML), which is composed of two basic elements and a set of symbols as its operators.

```
sememes: buy|买, GiveAsGift|赠, guilty|有罪, entice|勾引, human|人,
commerce|商业 ...
semantic relations: manner, purpose, domain, agent...
operators: {}, =, :, ~ ...
```

3. What is the current size of HowNet basic database?

Current general statistics are as follows (25/01/2016) :

Chinese characters	*20 892*
Chinese words & expressions	*125 022*
English words & expressions	*116 641*
Chinese meanings	*142 485*
English meanings	*137 566*
Definition	*33 268*
Record	*226 860*

[1] https://groups.google.com/forum/?hl=en#!forum/hownet, visité en 2015.

4. Then what is the current size in terms of syntactic and semantic categories?

Syntax		
PartOfSpeech	*English*	*Chinese*
adj	*12 599*	*13 721*
adv	*3 122*	*2 433*
aux	*101*	*103*
char(acter)	*0*	*14 405*
classifier	*0*	*447*
conj	*85*	*135*
coor	*7*	*14*
det	*123*	*58*
echo	*7*	*137*
expr(ession)	*1 000*	*850*
infs	*7*	*0*
letter	*57*	*57*
noun	*66 823*	*61 712*
num(eral)	*572*	*557*
pp	*1 372*	*0*
prefix	*28*	*7*
prep	*320*	*257*
pron	*109*	*176*
pun	*50*	*45*
root	*3 233*	*0*
stru	*0*	*82*
suffix	*7*	*0*
verb	*27 415*	*32 620*
wh	*87*	*48*

Semantics		
PartOfSpeech	*English*	*Chinese*
Entity	*4*	*3*
Event	*14 799*	*15 558*
Attribute	*5 201*	*4 665*
AttributeValue	*10 967*	*10 860*
Thing	*94 579*	*94 593*
Time	*2 872*	*2 872*
Space	*1 427*	*1 427*
Component	*9 208*	*9 209*

Categories:

木，林，树，梨，桃，李，森，枝，茶，权，栋，梁，檩

医院，医生，医药，医师，医士，医患，医德，医闹，医疗，医术

5. What does a RECORD look like in HowNet basic database?

As described in Q3, HowNet currently includes over 220,000 records in its database. In HowNet each word or expression is written in the form of a record. A record is composed of the following items:

`NO.=`	*serial number*
`W_C=`	*Chinese word or expression*
`G_C=`	*Chinese grammatical information*
`S_C=`	*Sentiment information for Chinese*
`E_C=`	*Examples for Chinese*
`W_E=`	*English word or expression*
`G_E=`	*English grammatical information*
`S_E=`	*Sentiment information for English*
`E_E=`	*Examples for English*
`DEF=`	*Definition of the concept*
`RMK=`	*Remarks*

Some records are shown below.

```
NO.=047395
W_C=动脉硬化症
G_C=noun [dong4 mai4 ying4 hua4 zheng4]
S_C=MinusEntity|负面实体
E_C=
W_E=atherosclerosis
G_E=noun [2 atherosclerosisnoun-0static,uncount]
S_E=MinusEntity|负面实体
E_E=
DEF={disease|疾病:scope={part|部件:PartPosition={nerve|络},whole={part|部件
:PartPosition={viscera|脏},whole={AnimalHuman|动物},{circulate|循环
:instrument={~}}}}}
RMK=Atherosclerosis comes from the Greek words athero (meaning gruel or
paste) and sclerosis (hardness).
```

```
NO.=162300
W_C=问
G_C=verb [wen4]
S_C=
E_C=
W_E=ask
G_E=verb [2 askverb-0vt,dobj,sobj,whobj,ofnpa22    ]
S_E=
E_E=
DEF={ask|问}
RMK=
```

```
NO.=128941
W_C=请求
G_C=verb [qing3 qiu2]
S_C=
E_C=
W_E=ask
G_E=verb [3 askverb-0vt,sobj,tocomp,thatobj,ofnpa23    ]
S_E=
E_E=
DEF={request|要求}
RMK=
 (Note the difference between two "ask" in their grammatical information)
```

```
NO.=142804
W_C=适销对路
G_C=adj [shi4 xiao1 dui4 lu4]
S_C=PlusSentiment|正面评价
E_C=
W_E=salable
G_E=adj [3 salableadj-0endmore,prepo1     ]
S_E=PlusSentiment|正面评价
E_E=
DEF={suitable|适宜:domain={commerce|商业},scope={sell|卖:possession={$}}}
RMK=
```

6. If I search with some semantic feature(s), can I extract all the words with the specified feature(s) from HowNet? If I can, how should I do?

Yes, you can. Look at HowNet Browser and you will find the following dialogue boxes which function with the search: KeyWord, Language, and search Mode at the Browser bottom line, the selection result may be shown as follows: exact, first, fuzzy, etc.

Take the English word "patient" for example.

Input the word "patient" into KeyWord and choose "exact" for search mode.

KeyWord: patient Language: Automatic exact

Then choose the first sense:

```
DEF={human|人 :domain={medical|医 },{SufferFrom|罹 患 :experiencer={~}},
{doctor|医治:patient={~}}}
```

Then choose the whole DEF as a semantic feature and input it, and select "first" as search mode.

KeyWord: "{human|人:domain={medical|医},{SufferFrom|罹患 :experiencer={~}}, {doctor|医治:patient={~}}}" *Language: feature first*

Then the Browser will show the statistical results at the right-hand corner: 8 DEF(s) found and 52 record(s) found.

Now take {SufferFrom|罹患:experiencer={~}} *as value of the KeyWord, and fuzzy as search mode.*

KeyWord: "{SufferFrom|罹患:experiencer={~}}" *Language: feature fuzzy*

Then the Browser will also show the statistical results at the right-hand corner: 23 DEF(s) found and 80 record(s) found.

If we want to extract and export all the records of the search results, we place the cursor at the result box and right-click. We then see a menu, choose "Export Search Result(full)", and "Export Search Result(Chinese words)" or "Export Search Result(English Words)".

Some of the 80 records of the search results are exported and shown below.

```
NO.=140765
W_C=失语症患者
G_C=noun [shi1 yu3 zheng4 huan4 zhe3]
S_C=MinusEntity|负面实体
E_C=
W_E=aphasic
G_E=noun [2 aphasicnoun-0static 个     ]
S_E=MinusEntity|负面实体
E_E=
DEF={human|人:{SufferFrom|罹患:content={disease|疾病:scope={speak|说
}},experiencer={~}}}
RMK=
```

```
NO.=120276
W_C=陪床
G_C=verb [pei2 chuang2]
S_C=
E_C=
W_E=stay in a hospital ward to look after a patient
G_E=verb [51stayverb-0vi      ]
S_E=
E_E=
DEF={stay|停留:location={room|房间:domain={medical|医},{doctor|医治
```

:location={~}}},purpose={TakeCare|照料:patient={human|人
:domain={medical|医},{SufferFrom|罹患:experiencer={~}},{doctor|医治
:patient={~}}}}}
RMK=

NO.=201564
W_C=主诉
G_C=noun [zhu3 su4]
S_C=
E_C=
W_E=chief complaint
G_E=noun [52complaintnoun-0]
S_E=
E_E=
DEF={text|语文:RelateTo={human|人:domain={medical|医},
{SufferFrom|罹患:experiencer={~}},
{doctor|医治:patient={~}}},
{ExpressDissatisfaction|示不满:instrument={~}}}
RMK=

NO.=181525
W_C=医嘱
G_C=noun [yi1 zhu3]
S_C=
E_C=
W_E=doctor's advice
G_E=noun [52advicenoun-0]
S_E=
E_E=
DEF={text|语文:domain={medical|医},
{propose|提出:agent={human|人:{doctor|医治:agent={~}}},content={~},
target={human|人:{SufferFrom|罹患:experiencer={~}}}}}
RMK=

NO.=040849
W_C=担架
G_C=noun [dan1 jia4]
S_C=
E_C=
W_E=stretcher
G_E=noun [2 stretchernoun-0static 付]
S_E=
E_E=
DEF={tool|用具:domain={medical|医}, {transport|运送:instrument={~},
patient={human|人:domain={medical|医},{SufferFrom|罹患:
experiencer={~}},{doctor|医治:patient={~}}} {human|人:{wounded|受伤
:experiencer={~}}}}}
RMK=

NO.=151072

```
W_C=探病
G_C=verb [tan4 bing4]
S_C=
E_C=
W_E=visit a patient
G_E=noun [51visitverb-0vi      ]
S_E=
E_E=
DEF={visit|看望:content={human|人:domain={medical|医},{SufferFrom|罹患
:experiencer={~}},{doctor|医治:patient={~}}}}
RMK=
```

7. Are you developing any HowNet-based machine translation systems?

Yes, we have recently built a HowNet-based English-to-Chinese MT system, named HowNet MT. The system is a rule-based and semantic-enhanced system. As the system employs much common-sense knowledge and semantics, its translation quality is incredibly good, much better than the free MT systems on the web. The system is composed of three parts: HowNet Knowledge-base, English syntactic and semantic analysis subsystem, and Chinese synthesis subsystem. The English analysis subsystem can be used on its own as an English parser to meet other NLP demands.

8. What is the HowNet-based tool called Sense_Colony_Tester and what tasks can it be applied to?

Each kind of bacteria has its own colony based on the same morphology. Similarly, each concept has its own colony. HowNet-based Sense_Colony_Tester (SCT) can represent and thus display the relevancy of a concept, or sense, to the other concepts in a given context.

SCT is bilingual, English or Chinese. Let's look at the following demo. Suppose your text to be tested is:

"I usually have bread and butter and homemade jam for breakfast."

After you input your text into the "Source Text" Box, and select English or Chinese and then click "submit".

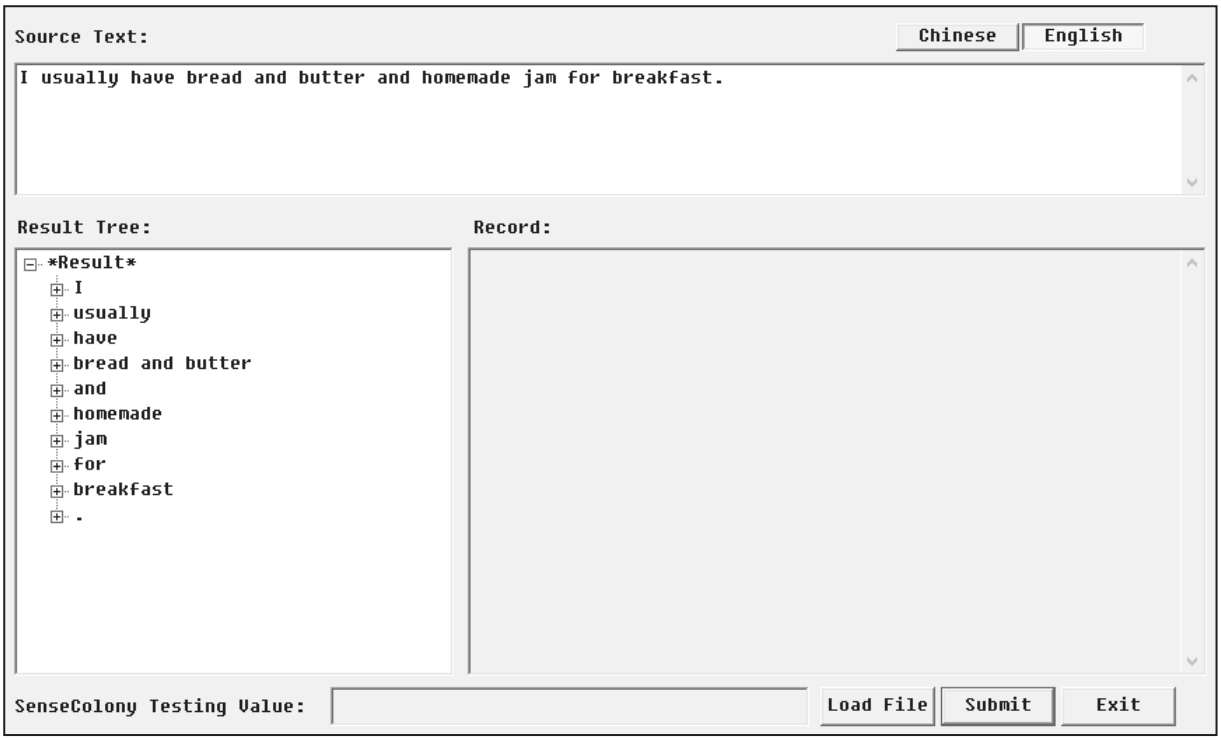


Figure 93 : HowNet Interface 1

In the "Result Tree" box, all the concepts or senses of the given text will be shown. We call this processing a text-CT. If only words rather than concepts or senses can be displayed, it is called text-X-ray as shown by Figure 93.

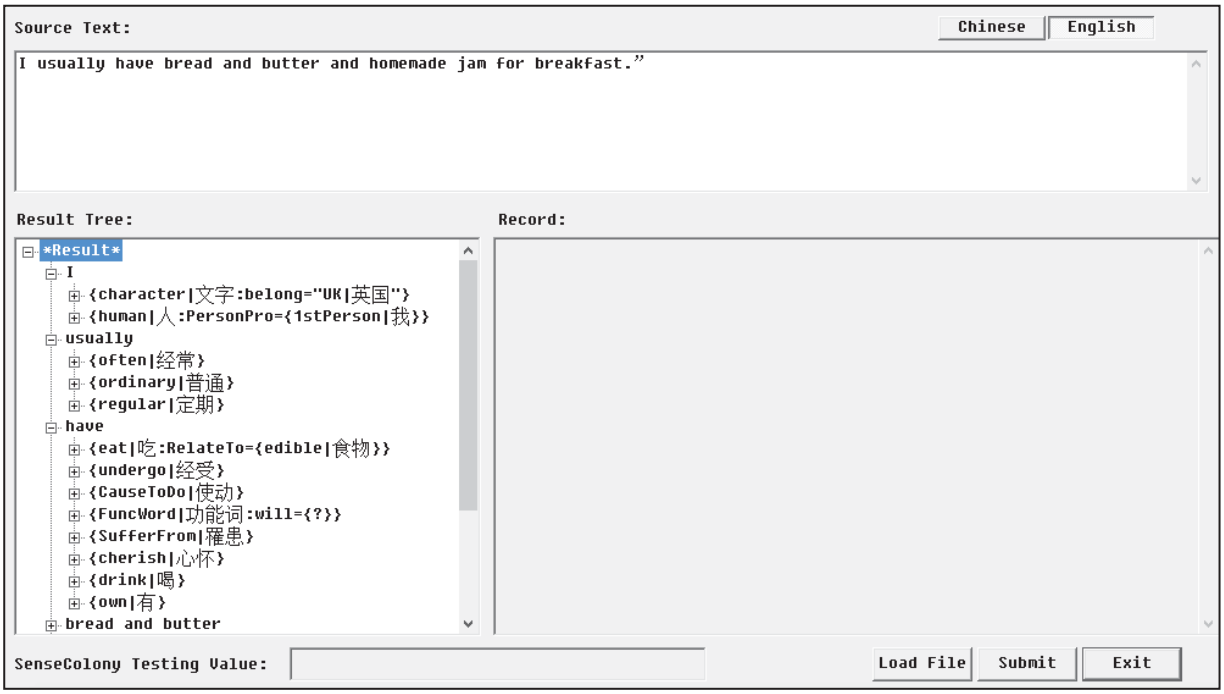


Figure 94 : HowNet Interface 2

SCT critical function is to give two statistical results: (1) Relevancy Count for every concept of the words in the tested text; (2) Distance Value of every concept from its immediate relevant neighbour.

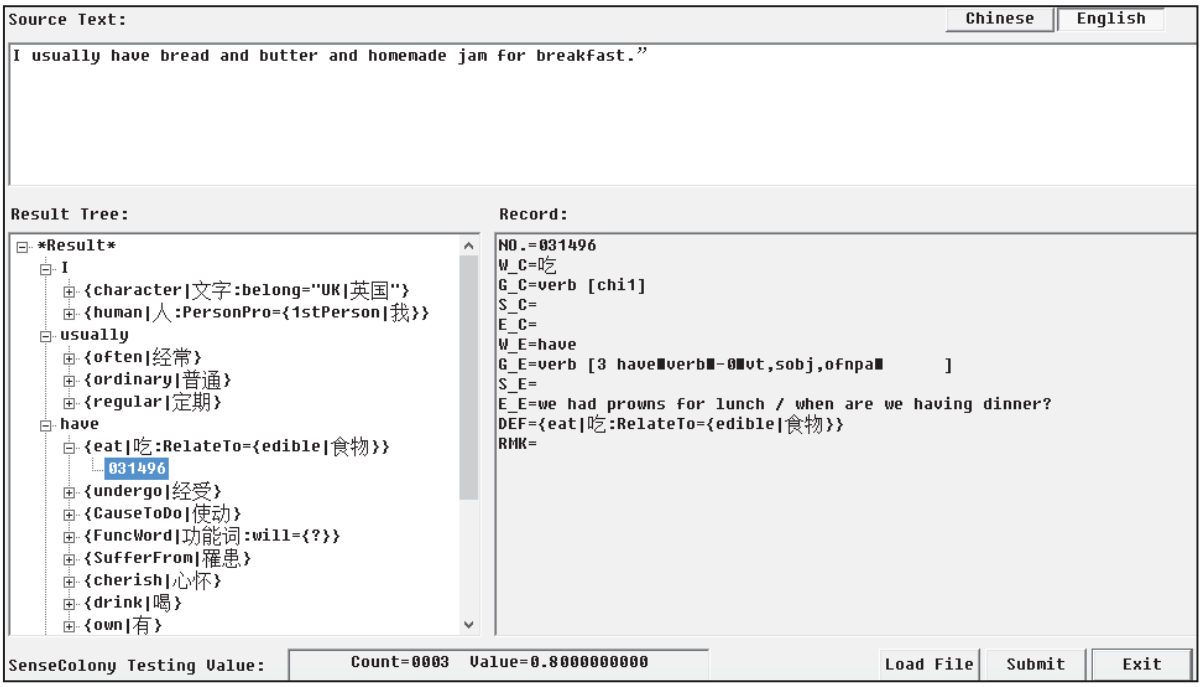


Figure 95 : HowNet Interface 3

Therefore, SCT can be applied to WSD, especially for those types of discourse ambiguities. SCT is employed in the HowNet-based machine translation system. Figure 95 *shows SCT will choose the sense of "eat" for the word "have", cf.* Figure 96 *And* Figure 97 *shows SCT will choose the sense of "food" for the word "jam", cf.* Figure 98. *The relevancy count of the sense of food in the word "jam" is 0004, which is the highest of its three senses. We suppose you may know what senses are the voters for this: "eat", "drink" of the word "have", and "food" of "bread and butter" and "eat" in the word "breakfast".*

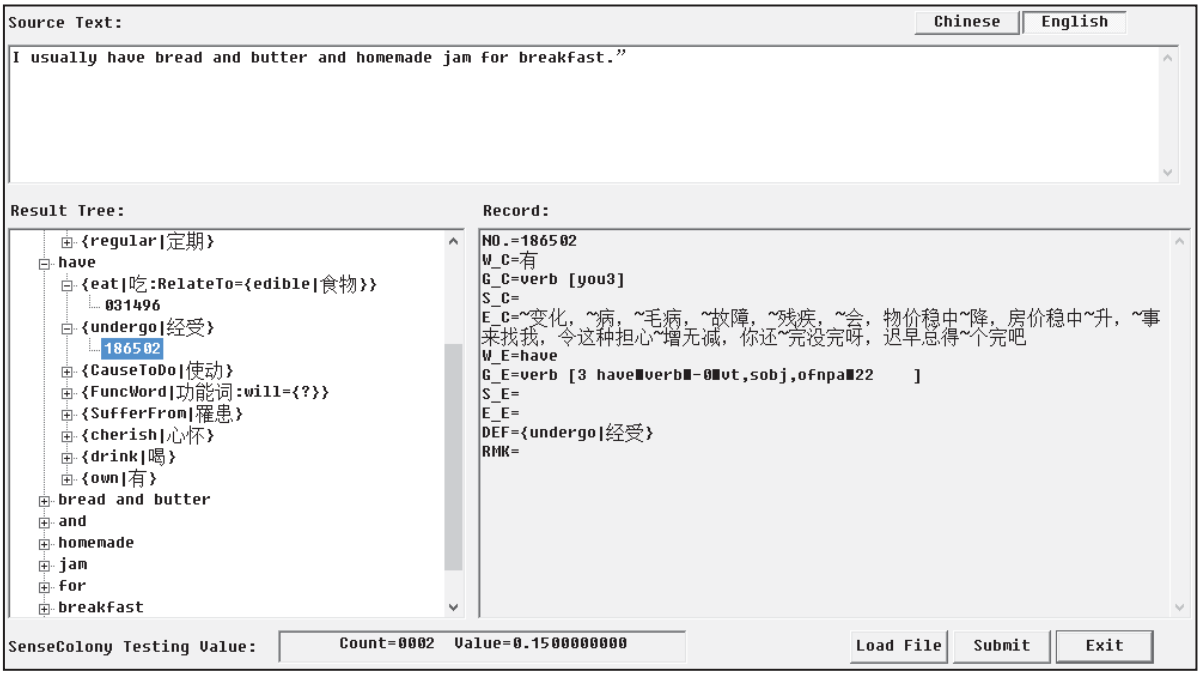


Figure 96 : HowNet Interface 4

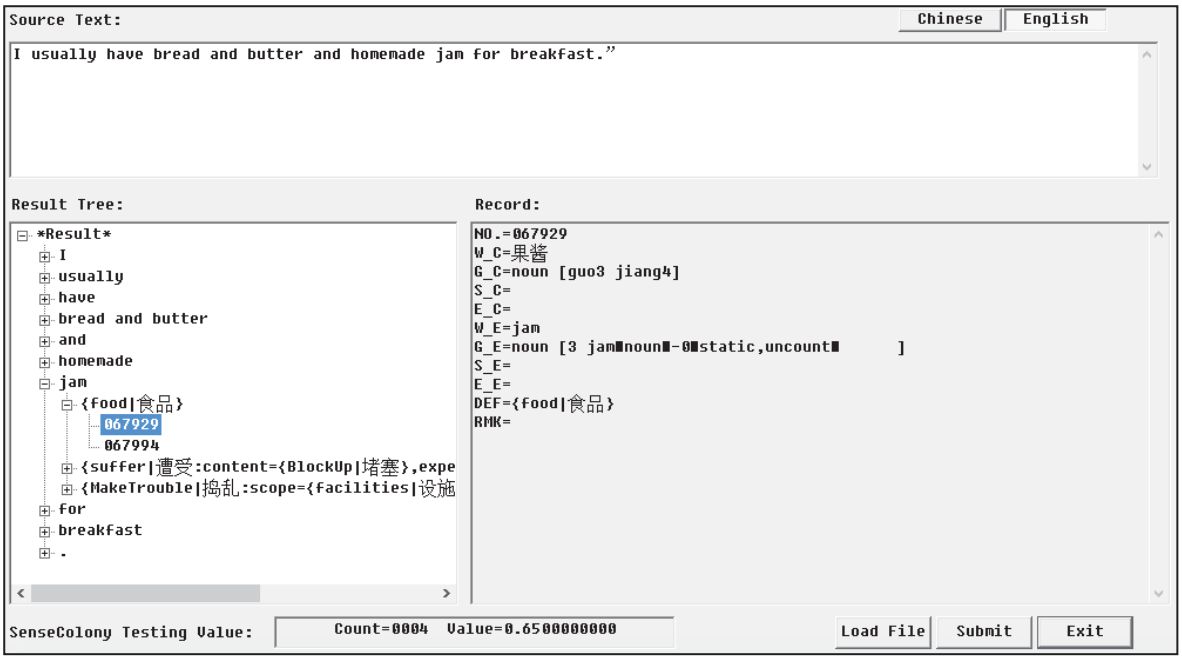


Figure 97 : HowNet Interface 5

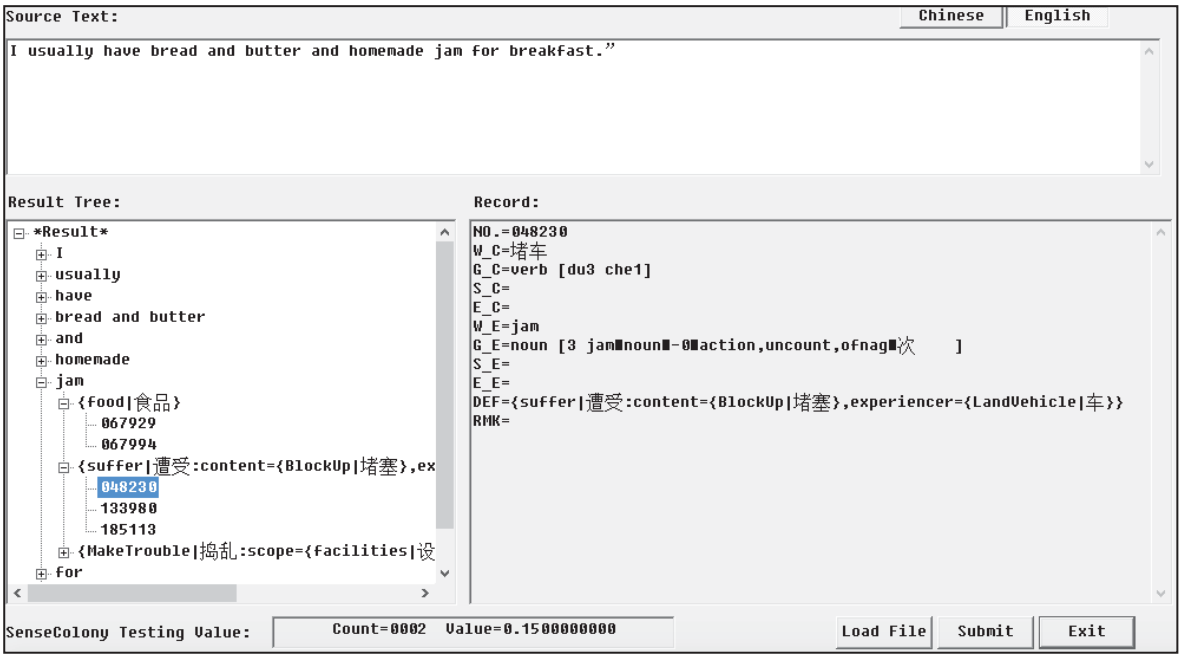


Figure 98 : HowNet Interface 6

SCT can also serve as an effective tool for tasks based on meaning computation, such as text summarization, text topic identification, text categorization, etc. Figure 99, Figure 100 *and* Figure 101 *show that the topic of the given text may be some kind of medicine and its relatedness to a symptom and infection, the statistical results of the concepts, the following concepts enjoy higher priority in Relevancy Count (more than 0010).*

Medicine:	*0024 (0025)*	*Treat:*	*0017*
Symptom:	*0024*	*Infection:*	*0018*
Cell:	*0024*	*Fungus:*	*0012*
Skin:	*0022*	*Cell membrane:*	*0012*

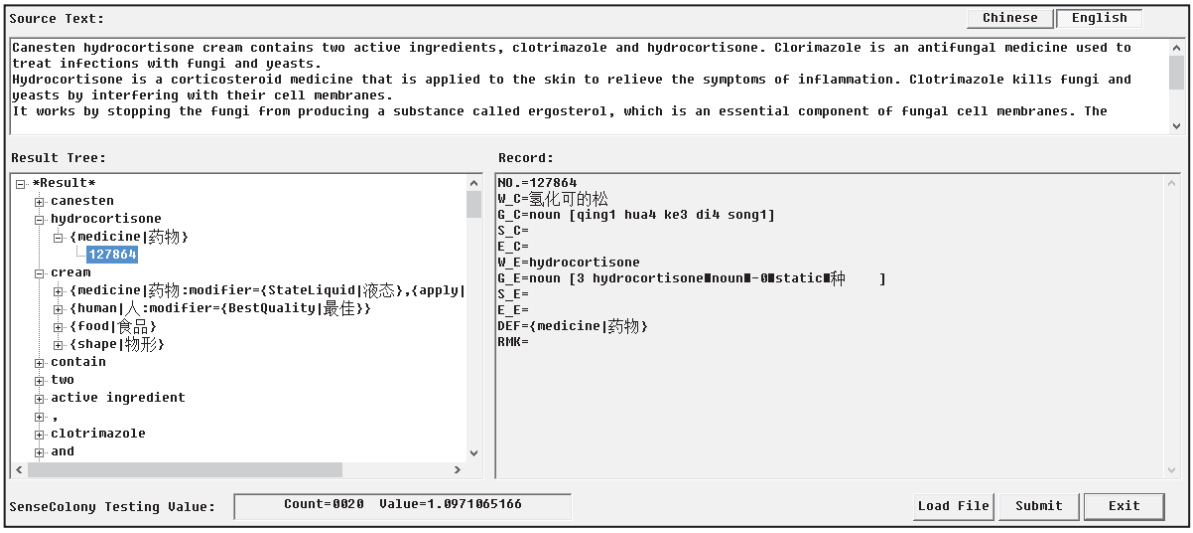


Figure 99 : HowNet Interface 7

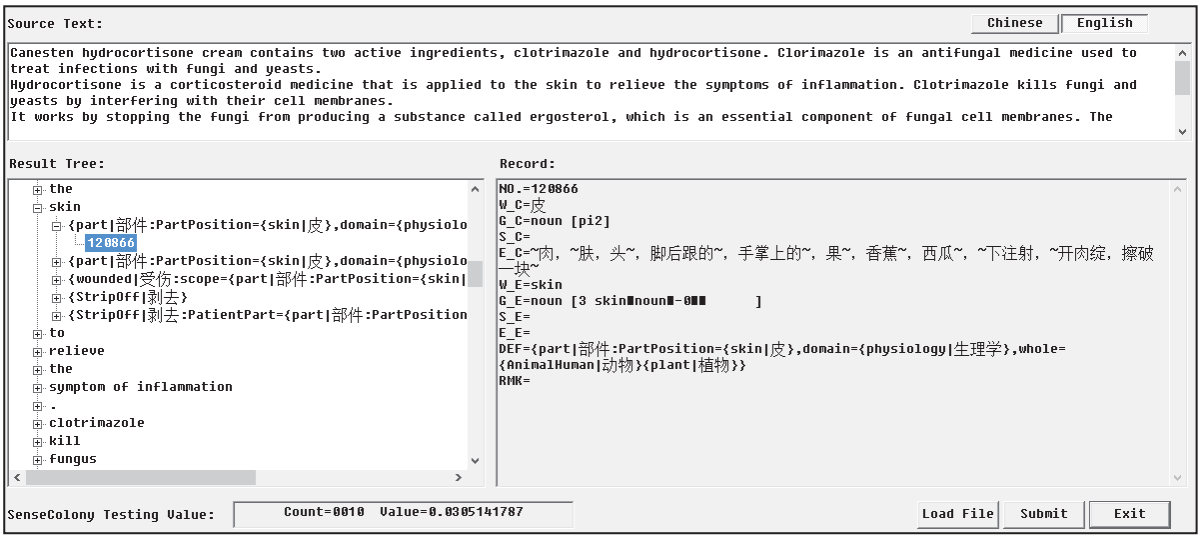


Figure 100 : HowNet Interface 8

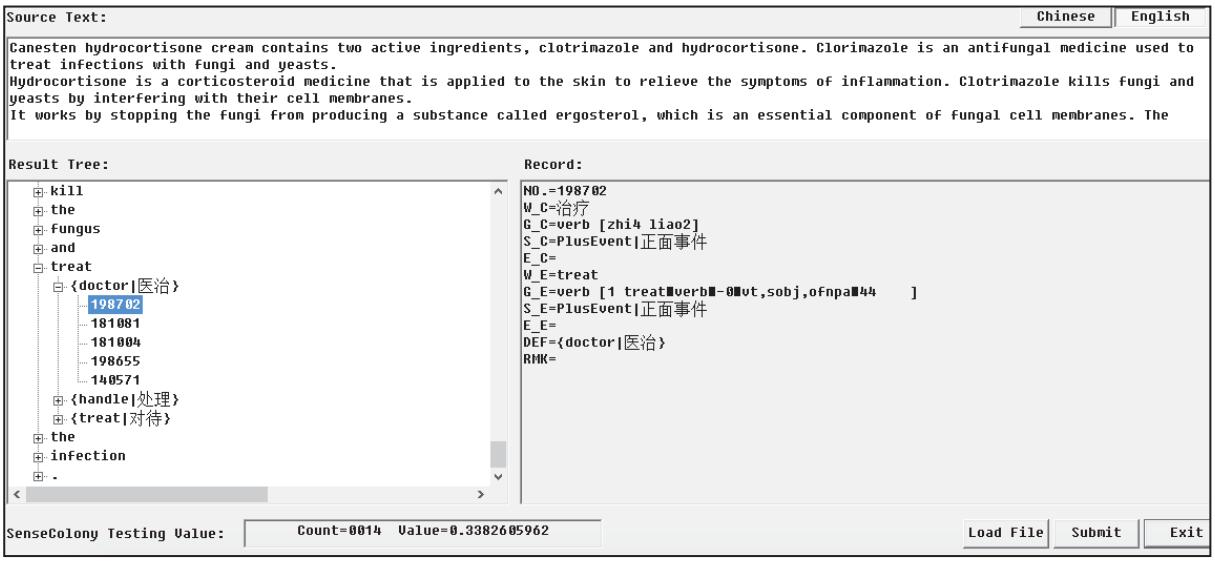


Figure 101 : HowNet Interface 9

9. I have a question on English; I think a native speaker will answer it. My questions are:

What are the correct or accurate English questions for the following statements:

(1) I kicked him in/on the leg. – "Where did you kick him?" It seems inaccurate.

(2) He signed his name at the right lower corner. "Where did he sign his name?" It also seems inaccurate.

In Chinese there are two different kinds of questions which we can distinguish:

你在哪踢了他？(Where did you kick him?) – in the playground

你把他哪踢了？(You kicked him where?) – in the leg

他在哪签了他的名字？(Where did he sign his name?) – in the office

他把名字签在哪了？(He signed his name where?) – at the right lower corner

Or in English the questions of the kind are ambiguous themselves.

Our friend Thomas Yale answered my questions as follows.

To answer your question, this is another way English is more ambiguous than Chinese.

(1) For the question "Where did you kick him?" the answer "I kicked him in/on the leg" seems the most reasonable in one's mind in English, but it could be answered with a particular place: "I kicked him in my apartment."

(2) The same case for "Where did he sign his name?" The most reasonable contextual answer in English is "He signed his name at the right lower corner (of the document)" but could also be answered "He signed his name at the law office" or even "He signed his name in Quebec", although we would regard both the latter geographical places as one being within another, and the anatomical place, which is a body part, can move and be anywhere.

Consider this significance, however. It appears that the verb in the question implies the kind of place that is to be understood. Compare "Where did you kick him?" which implies which body part, with "Where did you take him?" which implies a geographical place.

Because, you see, the definitions (DEFs) in HowNet that "kick" is defined with the single sememe "kick", and "take" (the definition that applies with the above sentence, one translation I think is dai4 ru4) is defined as "guide". Significantly, the two words are in different places in the Event taxonomy: "kick" is listed under "AlterForm" / "AlterAppearance" / "AlterAttribute" and "guide" is listed under "CauseToDo" / "MakeAct".

So perhaps this is the key to solving this problem of translation: the fact that they do occupy different places in the taxonomy is the basis to deduce what kind of place (anatomical, geographical, or maybe other kinds of places) expected from the answer is most reasonable, or is anticipated in the answer.

From your examples it appears that Chinese avoids this ambiguity in its syntax, by placing interrogative pronouns (like "where", perhaps others?) in a different place among the words in the sentence:

- *你在哪踢了他？("Where you kick+past him?") as opposed to "You kicked him where?"*
- *The same with 他在哪签了他的名字？("Where he sign+past his name?") as opposed "He signed his name where?"*

10. What are the features specified for English words in HowNet Dictionary? [features total 135]

Apart from POS, the following features are specified and allotted in the five domains of each English word or MWE in the square brackets. Here they are.

1. 1st domain – word structural information

The numbers represent: [7]

[1+(space)+single word token ("1" – most high-frequency sense)
[2+(space)+single word token ("2" – more high-frequency sense
[3+(space)+single word token ("3"– general high-frequency sense)
[7+(space)+single word token ("7" – never-chosen for E-C MT
[4+(1~10/11/12...)+head of MWE ("4" – most high-frequency sense)
[5+(1~10/11/12...)+head of MWE ("5" – general frequency sense
[6+(1~10/11/12...)+head of MWE ("6" – never-chosen for E-C MT
(1~10/11/12...) represents the place of the head in the MWE

2. 2nd domain – POS information [17]

noun
pron
verb
adj
adv
num
det – determiner (the)
aux – auxiliary verb (be, have, do, can, may…)
wh – wh-words (who, when, as, though)
conj – coordinate conjunctions
prep – prepositions (in, for, according to, on behalf of…)
pp – preposition phrase (on bad terms…)
infs – infinitive sign (to, in order to, so as to)
expr – an expression, syntactically independent
echo – a sound
prefix
suffix

3. 3rd domain – Word-ending information: [11]

```
-0   //normal word
-s (children)  // special plural
-0s (police)  // no plural or invariable plural
-v (wrote)  // special past form
-ed (bought) //special  -ed (past)
-en (written) //special -en (past participle form)
-ing (feeling, meeting)
-ings
-aaa (put)  //strong verb with 3 same forms: present – past – past
participle
-aba(come-came-come) // present = past participle
-aab (beat-beat-beaten) //present = past
```

4. 4th domain – syntactic information [4]

neg – negative (no, none…)
present – present time
past – past time
future – future time

4.1 Word/MWE form information: [6]

fup – first letter in upper-case
aup – all letters in upper-case
acro – acronym
endthat – MWE ending in "that" (it is said that,

endprep – MWE ending in a preposition (be confident in, rely on…)
endto – MWE ending in an infinitive-to (be determined to)

4.2 for nouns: [9]

action – action noun
ofnpa – in "action-noun + X": X normally as its patient
ofnag – in "action-noun + X": X normally as its agent
static – static noun
uncount – uncountable noun
unit – a unit (meter, foot, gram…)
proper – a proper noun (Canada, Tokyo, Obama, IBM…)
nomodinoun – almost never used to modify a noun (means)
moneysign – a moneysign ($,￥..)

4.3 for numerals [19]

ordi – ordinal (first, 2nd, tenth…)
card – cardinal (1, two, million…)
predet – can come before determiners (all, both, either, none of)
singular – singular (1)
plural – plural (2,10,100…)
indef – indefinite amount (some)
demo – demonstrative (this, those)
date – indicates "date" (1- 31)
hour – indicates "hour" (1- 24)
minute – indicates "minute" (1 - 60)
bage – indicates "baby's age" (1 - 3)
tage – indicates "troddler's age" (4 - 5)
cage – indicates "child's age" (6 - 9)
tnage – indicates "teenager's age" (10 - 15)
yage – indicates "youth" (16 - 41)
mage – indicates "middle-age" (42 - 69)
oage–indicates "old age" (70 -112)
1digit – indicates one-digit number (1, 2, 3… 9)
2digit – indicates two-digit number (10, 11, 12… 99)
3digit – indicates three-digit number (100, 101, 110… 999)
4digit – indicates four-digit number (1000, 1001, 1110… 1999)

4.4 for pronouns [5]

accusat – accusative case (him)
nominat – nominative case (we)
reflex – reflexive (oneself)
singular – singular
plural – plural

4.5 for verbs:[15]

vt – transitive
vi – intransitive
copula – copula
linkall – copula which can take n/adj/ved/ven as its complement (be)
linkadj – copula which can take adj/ved/ven as its complement (get)
sobj – can take a single obj (destroy)
dobj – can take a double obj (give)
ingobj – can take a Ving obj (start, keep, stop…)
thatobj – can take a that-clause as its obj (say)
whobj – can take a wh-clause as its obj (ask)
toobj – can take an to-infinitive-verb as its obj (want)
nouncomp – can take a noun as its complement (call)
tocomp – can take an to-infinitive-verb as its complement (ask)
adjcomp – can take an adj as its complement (paint)
ingcomp – can take a ving as its complement (keep)

4.6 for adjectives: [5]

ender – for adj/adv takes "-er" ending when in comparative
endmore – for adj/adv takes more" when in comparative
prepo – usually comes before the element it modifies
postpo – usually comes after the element it modifies
infmodi – often followed by an infinitive phrase (difficult, eager…)

4.7 for adverbs: [7]

ender – for adj/adv takes "-er" ending when in comparative
endmore – for adj/adv takes more" when in comparative
minf – can be used to modify infinitives (merely, just…)
mnum – can be used to modify numerals (only, about, just…)
madjonly – can only be used to modify adjectives
sentent – usually used as a sentential adverb (it be said that)
nomverb – never used to modify verbs to its left (more than)

4.8 for conjunctions: [3]

conjand – conjuction "and"
conjbut – conjunction "but"
conjor – conjunction "or"/'nor'

4.9 for infinitive sign: [TM/2]

infw – single word (to)
infp – MWE (in order to, ready to, so as to…)

4.10 for auxiliaries: [6]

auxdo auxiliary "do"
auxheve auxiliary "have"
auxbe auxiliary "be"
modal modal auxiliaries (can, may, should, would rather…)
tense tense auxiliaries (will, would, shall, should)
quasiaux quasi-auxiliaries (may just as well, be more likely to)

4.11 for wh-words: [7]

question – can be used as a question-word (who, which,why…)
objclaus – can introduce an object clause (which, what, if…)
whinf – can introduce an infinitive-phrase (how, which, what…)
person – a person (who)
advsubj – can be the subject of the clause (who, which, that…)
anonsubj – can not be the subject of the clause (why,)
adverbial – can introduce an adverbial clause (though, if…)

4.12 for prepositions or pp: [3]

advb – mostly used to be an adverbial (during, by…)
attr – mostly used to be an attributive (like, regarding…)
ingpred – when a ving after it, ving mostly is a quasipredicate (by…)

4.13 for punctuations: [9]

symbolic – a symbolic (&, #...)
fullstop – fullstop (“.”. “…”)
quesmark – question mark (?)
bracket – brackets (“”, (,)
hyphen – hyphen (-)
comma – comma (,)
colon – colon (:)
title – (#)
semicolon – (:)

5. 5th domain – Chinese-transfer information
In fact, the information is for Chinese equivalents
For example,
specific classifiers for nouns (本， 只...);
Chinese function words for verbs in passive (被， 加以...) with "地" ending for adverbs.

11. What are the logico-semantic relations used in HowNet Translation System?

They are as follows:

00	*title*	*题目、标题、题外话*
02	*ExtraSentence*	*句外成分 (sentential adverbs, like: Frankly speaking, in a word.)*
03	*sentential*	*句副词（如：一般情况下）*
04	*comment*	*评论性短语（如：众所周知）*
05-1	*SincePeriod*	*起自时段（近年来，近十年来，四年来，这些年来..）*
05-2	*SincePoint*	*起自时点 （建国以来，从小，自从他出国以来..)*
06	*DurationBeforeEvent*	*前耗时段 （不到三年就成名了）*
07-1	*TimeAfter*	*之后*
07-2	*TimeBefore*	*之前*
08	*concession*	*让步（如：尽管、虽然）*
09	*condition*	*条件（如：如果、要是）*
10-1	*attrcl*	*定语从句*
10-2	*advcl*	*状语从句*
20-1	*agent*	*施事*
20-2	*experiencer*	*经验者*
20-3	*relevant*	*关系主体*
20-4	*possessor*	*领有者*
20-5	*coagent*	*合作施事*
22	*byagent*	*带被-施事*
24	*time*	*时间*
26	*location*	*处所*
28	*negfre*	*否定频率（如：从不）*
30	*demo*	*指示（如：这、那种）*
32	*quantity*	*数量/次第*
34	*also*	*也（如：她也来了）*
36	*emphasis*	*强调（绝对）*
38	*will*	*意志（如：他也会来）*
40	*neg*	*否定*
42-1	*must*	*务必*
42-2	*seem*	*似乎*
42-3	*able*	*能够*
44	*frequency*	*频率*
48	*cause*	*缘由、原因*

50	*source*	*来源（从书店买书，问她买了一本书）*
52-1	*LocationThru*	*经过处所*
52-2	*EventProcess*	*事件经过*
54	*scope*	*范围*
56-1	*AccordingTo*	*根据*
56-2	*concerning*	*有关*
58	*material*	*材料*
60-1	*means*	*手段*
60-2	*method*	*方法*
62	*instrument*	*工具*
64	*bapatient*	*带把(pinyin: bǎ)–受事 (patient – in the sentence contains 把)*
66-1	*direction*	*方向*
66-2	*aim*	*目标*
68	*beneficiary*	*受益者*
70	*contrast*	*参照体*
72	*partner*	*相伴体*
74	*accompaniment*	*伴随*
76	*attributive*	*定语*
80	*cost*	*代价*
82	*degree*	*程度*
84	*modifier*	*修饰（通常是形容词）*
86	*restrictive*	*限定（通常是名词，如计算机软件）*
88	*manner*	*方式*
92-1	*LocationIni*	*原处所（如：从北京）*
92-2	*StateIni*	*原状态（如：从英语）*
100	*Head (Pivot)*	*中心（中心事件、中心实体）*
110	*equiv*	*同位成分*
117	*target*	*对象*
118	*distance*	*距离*
120	*possession*	*占有物*
122-1	*patient*	*受事*
122-2	*PatientProduct*	*受事成品*
122-4	*ContentProduct*	*内容成品*
122-5	*descriptive*	*描写体*
122-6	*isa*	*类指*
122-7	*existent*	*存现体*
123	*content*	*内容*
132	*PartOfTouch*	*触及部件*
134-1	*ResultIsa*	*结果类指*
134-2	*ResultEvent*	*结果事件*
134-3	*ResultWhole*	*结果整体*
136-1	*PatientValue*	*受事属性值/属性*

140-1 LocationFin	*终处所*
140-2 StateFin	*终状态*
144 QuantityCompare	*比较量（如：这条路比那条长 3 公里）*
146-1 spefreq	*特定频率*
146-2 times	*动量*
150 result	*结果*
160 purpose	*目的*
162-1 duration	*时段*
162-2 DurationAfterEvent	*后延时段*
162-3 TimeFin	*终止时间*
170 succeeding	*后续*
180 remark	*注脚*

Annexe 5 Exemples des microstructures de LEXALP

```
<entry id="fra.espèces_protégées.1038.e" lang="fra" legalSystem="AC" status="UNKNOWN">
    <term>espèce protégée</term>
    <grammar>n.f.</grammar>
    <phraseological-unit>false</phraseological-unit>
    <domain>1.4</domain>
    <domain>4.4</domain>
    <usage frequency="common" geographical-code="AC" technical="false"/>
    <definition>
        <text>
            Espèce menacée ou vulnérable qui est protégée de l'extinction par des mesures
            préventives.
        </text>
        <source date="2006/08/31" sourceType="Url">
            http://glossary.eea.europa.eu/EEAGlossary/P/protected_species
        </source>
        <source date="2006/09/15" sourceType="Author">Randier (trad)</source>
    </definition>
    <context>
        <text>
             [L'] [é]tat des inventaires des espèces animales sauvages [comprend l'élaboration
             de] [l]istes des espèces protégées [.]
        </text>
        <source documentId="3491828" >Prot. PNEP, ann. I, art. 1.1.2.</source>
    </context>
    <note>...</note>
    <harmonisation-note>...</harmonisation-note>
</entry>
```

Figure 102 : Microstructure XML du terme "espèce protégée"

```
<axie id="axi..1432634.e">
    <termref idref="ita.specie_protette_per_legge.1037.e" lang="ita"/>
    <termref idref="fra.espèces_protégées.1038.e" lang="fra"/>
    <termref idref="deu.rechtlich_geschützte_Arten.1039.e" lang="deu" />
    <termref idref="slv.pravno_zavarovana_vrsta.1040.e" lang="slv"/>
    <termref idref="ita.specie_protetta.1200075.e" lang="ita"/>
    <termref idref="slv.zavarovana_vrsta.1256879.e" lang="slv"/>
    …
    <axieref idref="axi..2184588.e"/>
    <axieref idref="axi..2184615.e"/>
    …
</axie>
```

Figure 103 : Microstructure XML d'une axie

Annexe 6 Implémentation de l'affichage en colonnes dans PIVAX-2

Nous avons présenté l'interface en colonnes dans Pivax-2 au II.4.3.2. Voir la Figure 32.

Pour l'implémentation, on a repris la méthode de H.-T. Nguyen. Il a créé un modèle d'affichage en colonnes (un fichier en XHTML) dans PIVAX-1. Voir la Figure 104.

Pour le résultat affiché sur l'interface, on copie le modèle comme un nœud, puis on remplace les contenus par les identificateurs.

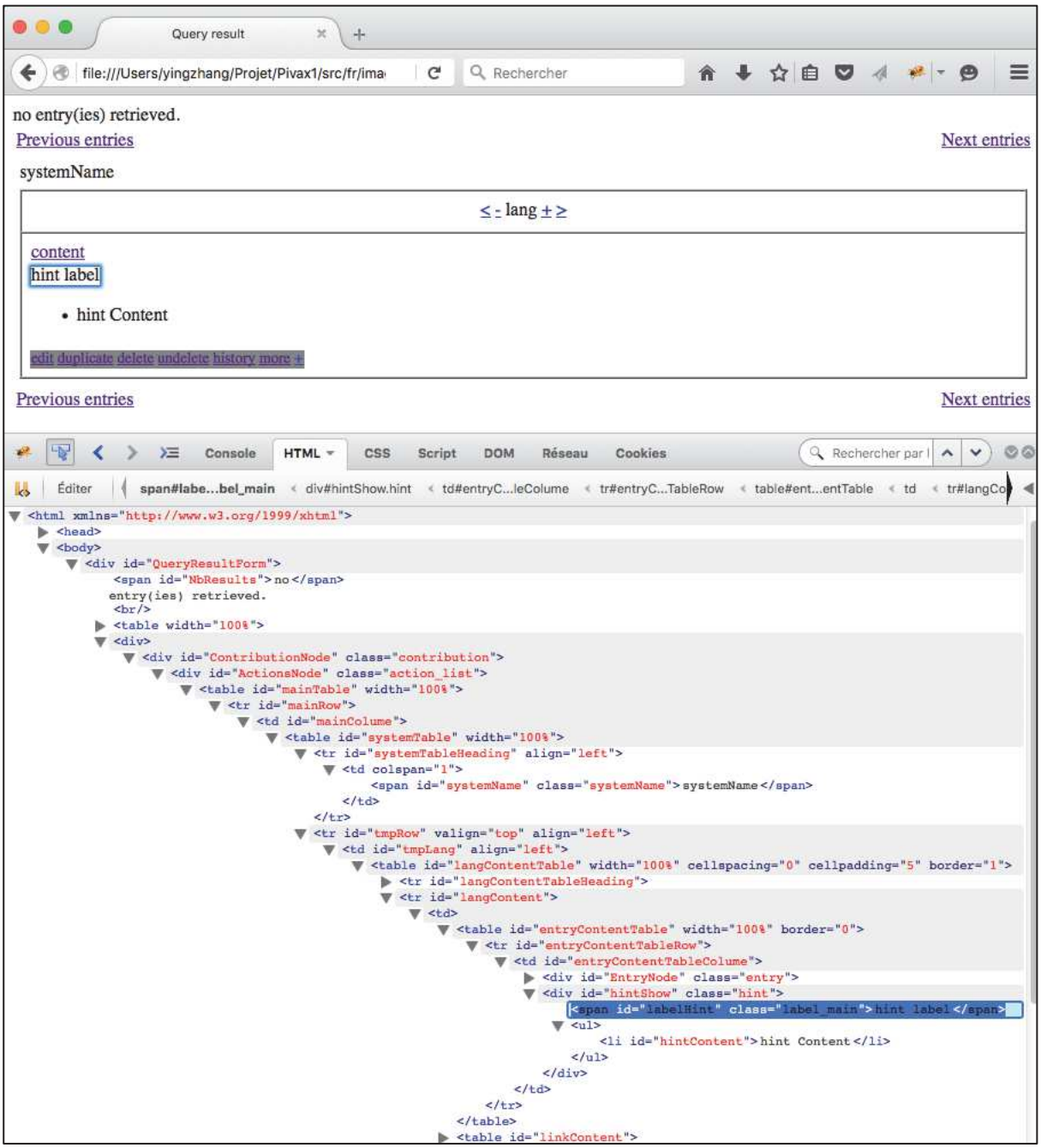
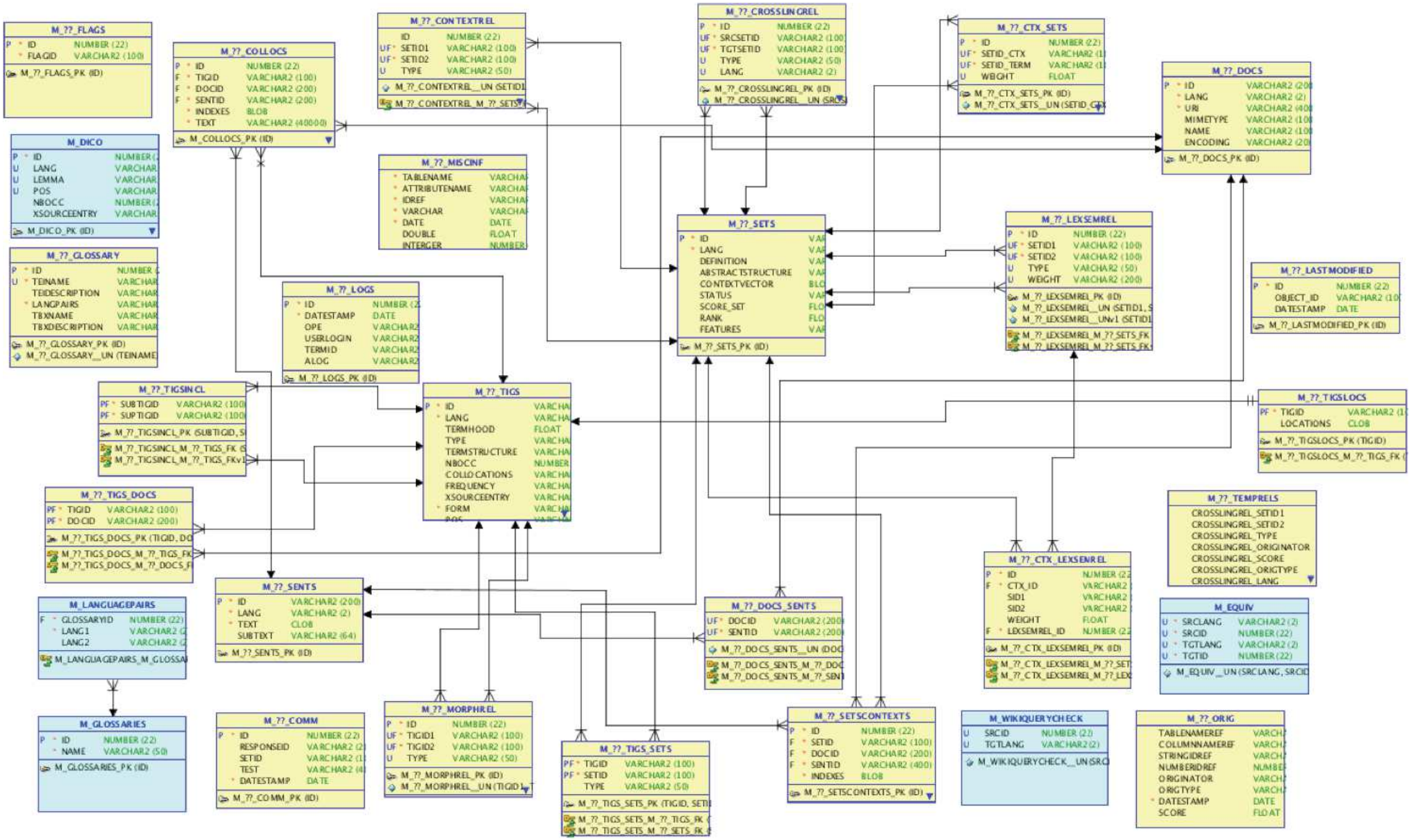


Figure 104 : Modèle de l'affichage en colonnes

Annexe 7 Schéma de la base de données lexicales de LIBELLEX



Annexe 8 Algorithmes de calcul dans PIVAX-3 en pseudo-code

Pour simplifier la présentation, on ne donne que l'algorithme principal de la consultation générale. La consultation avancée est basée sur ces algorithmes, mais utilise des conditions supplémentaires.

```
Entrée : terme (t) et langue (lg)
Sortie : page affichée de la liste de consultation de toutes les langues

/* Initialisation de la liste des traductions de 1er niveau */
Collection_par_axie <entree_lexie>;
/* Initialisation de la liste des traductions de 2e et 3e niveau */
Collection_par_proaxie <entree_lexie, niveau>;

Consultation_générale (t, lg){
  Collection (Collection_par_axie, Collection_par_proaxie)
           = Collect_des_liens (t, lg)
  Affichage (Collection_par_axie, Collection_par_proaxie)
}

/* Collecter tous les liens possibles dans la BDLex */

Collect_des_liens (t, lg) {
  Pour chaque volume de lexies (vol_lex)
    Si la langue de vol_lex = lg
      Pour chaque lexie entrée (lex) dans vol_lex

          // Chercher par la table d'indexation (cdm-headword)
          Si le mot-vedette (hw) = t

          /*
          Collecter tous les liens de lex dans sa table LINKS par la clé
          étrangère. Pour chaque enregistrement reçu, on a le type de lien
          (type), le volume cible (volumetarget), la référence cible
          (targetId), et l'étiquette (label).
          */

          Select type, volumetarget, targetId, label from vol_lex_LINKS where
          entryid = objectid_de_lex
          Si le type de lien (type) = axeme
            Liste_traduc_1
                   = Chercher_dans_volumes (targetId, volumetarget);
            Pour chaque élément (ele) de la Liste_traduc_1
              Si ele n'existe pas dans Collection_par_axie
                Collection_par_axie += ele
              fSi
            fPour
          fSi
          Si le type de lien (type) == prolexeme
            Liste_traduc_2
                   = Chercher_dans_volumes_prolexeme (targetId, volumetarget,
                     lable_src)
            Pour chaque élément (ele2) de la Liste_traduc_2
              Si ele2 n'existe pas dans Collection_par_pro
                Collection_par_proaxie += ele2
              fSi
            fPour
          fSi
        fSi
      fPour
    fSi
  fPour
  Return Collection (Collection_par_axie et Collection_par_proaxie)
}
```

```
/* Trier et afficher */
Affichage (Collection_par_axie, Collection_par_proaxie){
  Pour chaque entre lexie de Collection_par_axie
    Chercher le mot-vedette (headword) dans sa table d'indexation
    Ajouter le headword dans List_traduction_niveau_1
  fPour
  Pour chaque entrée lexie de Collection_par_proaxie
    Si cette lexie est la traduction de deuxième niveau
      Chercher le mot-vedette (headword) dans sa table d'indexation
      Si ce headword n'est pas dans la List_traduction_niveau_1
        Ajouter le headword dans List_traduction_niveau_2
      fSi
    fSi
  fPour
  Si List_traduction_niveau_1 n'est pas vide
    Ajouter les élements de la List_traduction_niveau_1 dans le module du
    premier niveau de traduction dans la page Xhtml.
  fSi
  Si List_traduction_niveau_2 n'est pas vide
    Ajouter les élements de la List_traduction_niveau_2 dans le module du
    deuxième niveau de traduction dans la page Xhtml
  fSi
  Collection_tous = Distinct (Collection_par_axie + Collection_par_proaxie)
  Pour chaque lexie de Collection_tous
    Nœud_Xhtml
          = Mise en forme de nœud Xml de lexie à l'aide de feuilles de style
    Ajouter Nœud_Xhtml dans le module du troisième niveau de traduction dans
    la page Xhtml
  fPour
}

/*Chercher récursivement les traductions par axème et axie*/

Chercher_dans_volumes (targetId, volumetarget) {
  Chercher les liens vers les axèmes
  puis vers les axies
  puis vers les axèmes cibles
  puis chercher les lexies cibles /* type de lien = final, s'arrêter*/
  return toutes les entrées trouvées
}

/*Chercher récursivement les traductions par prolexeme et proaxie et traitement
d'étiquette*/

Chercher_dans_volumes_pro (targetId, volumetarget, lable_src) {
  Chercher les liens vers les prolexèmes
  puis vers les proaxies
  puis vers les prolexèmes cibles
  puis chercher les lexies cibles /* type de lien = final, s'arrêter*/
  Si l'étiquette de prolexème cible vers lexie cible = lable_src
    Marquer que c'est la traduction de deuxième niveau
  Sinon
    Marquer que c'est la traduction de troisième niveau
  fSi
  return toutes les entrées trouvées avec leurs niveaux
}
```

Annexe 9 Expérience sur les appels de TRADOH par SECTRA via ACTIVEMQ pour les prétraductions

1. Contexte

Nous avons brièvement présenté cette expérience au IV.1.3.3. Cette expérience concerne quatre spécifications. Selon l'ordre des appels (SECTRA→ACTIVEMQ→TRADOH→ACTIVEMQ→SECTRA), il s'agit de : (1) application émettrice de SECTRA, (2) application réceptrice de TRADOH, (3) application émettrice de TRADOH, et (4) application réceptrice de SECTRA.

Le texte source chinois a été produit par L. X. Wang. Il s'agit d'une partie d'un corpus économique.

Dans cette expérience, l'entrée est un fichier TXT, chaque ligne contient un segment. L'application émettrice de SECTRA lit ce fichier, puis récupère les segments et envoie un par un vers la queue de TRADOH (via ACTIVEMQ). Chaque requête contient donc un seul segment[1].

TRADOH envoie les traductions une à une dans la queue de SECTRA (via ACTIVEMQ). Puis l'application réceptrice de SECTRA regroupe toutes les traductions grâce à un identifiant de groupe de requêtes (`JMSXGroupID`, voir plus bas).

2. Description détaillée et algorithme

Pour bien comprendre les algorithmes, on présente d'abord les interfaces de ACTIVEMQ[2], puis les descriptions détaillées de `message`. Ces présentations sont purement techniques, mais nous aident à comprendre comment fonctionne ACTIVEMQ.

Table 15 : Interfaces d'ACTIVEMQ

Interfaces	Description
`ConnectionFactory`	Un objet administré par le fournisseur (*provider*) pour créer une Connection.
`Connection`	Une connexion active vers un JMS fournisseur.
`Destination`	Encapsule l'identité d'une destination.
`Session`	Contexte (mono-processus) pour l'émission et la réception de messages.
`MessageProducer`	Objet pour l'envoi de messages vers une Destination (créé par la Session).
`MessageConsumer`	Objet pour la réception de messages d'une destination (créé par la Session).
`Message`	Encapsulation d'un message, Chaque `message` contient une structure à trois niveaux : entête, propriété et corps (voir la table ci-dessous).

[1] Dans notre expérience, on peut envoyer le fichier entier dans une seule requête. Mais alors, il faut gérer les files d'attente du côté de TRADOH. Donc on délègue toute la gestion des travaux à ACTIVEMQ, et on minimise les requêtes (c'est-à-dire que chaque requête peut être executée directement sur TRADOH).

[2] Ces sont les mêmes interfaces que celles de JMS (Java Message Service). JMS est la technique sous jacente d'ACTIVEMQ.

Table 16 : Structure de `message`

Niveaux	**Description**
Entête (`header`)	Attributs communs (voir IV.1.3.1.2), destinés au routage et à l'identification du message.
Propriété (`property`)	Champs supplémentaires pour les propriétés optionnelles, par exemple, `JMSXGroupID` (groupe de messages), `JMSXRcvTimestamp` (date de réception), etc.
Corps (`body`)	Il existe 5 sous-types : `StreamMessage`, `MapMessage`, `TextMessage`, `ObjectMessage`, et `BytesMessage`. Cela permet d'accpeter non seulement les objets JAVA, mais aussi d'autres types messages. Par exemple, pour `TextMessage`, ça pourrait être de type `String` ou un document XML.

Les applications émettrices sont basées sur 2 étapes de traitement.

1. Créer la connexion avec le serveur ACTIVEMQ, préciser les contextes de session (par exemple, c'est un émetteur et c'est une instance de SECTRA).
2. Envoyer le(s) message(s).

Les applications réceptrices sont également basées sur 2 étapes de traitement.

1. Créer la connexion avec le serveur ACTIVEMQ, préciser les contextes de session, écouter récursivement. S'il y a un message arrivé, on appelle le traitement spécifique (la méthode principale).
2. Par exemple, pour l'application réceptrice de TRADOH, on a la méthode `call_Tradoh` qui est un "interprète" de messages (chaque message est sous une forme prédéfinie, ex. `segment_source|langue_source|langue_cible|système_TA|id_segment`[1]). Cette méthode `call_Tradoh` s'occupe également l'envoi de la requête à l'API de TRADOH. Après reçu le résultat de TRADOH, elle envoie le symbole `acknowledge` à ACTIVEMQ. À la fin, elle vérifie s'il y a la valeur pour l'attribut `ReplyTo`. Si oui, elle appelle l'application émettrice de TRADOH pour envoyer la réponse.

Voici l'algorithme d'application réceptrice de TRADOH.

```
/*Application réceptrice de Tradoh*/
Receiver_Tradoh {
/* Le code de connexion est un code de « formule » pour connecter ActiveMQ à un
récepteur */
/* On doit mettre ce code dans toutes les applications réceptrices */

  /* Création connectionFactory */
  ConnectionFactory connectionFactory;

  /* Initialisation connection */
  Connection connection = null;

  /* Création Session */
  Session session;
```

[1] Ce format est un résultat de négociation, et est construit par l'application émettrice de SECTRA. C'est le contenu du message. Par exemple,

```
String msg = segment+ "|zho|fra|Google|"+id_segment ;
TextMessage message = session.createTextMessage(msg);
```

```
/* Création destination */
Destination destination;

/* Création consumer */
MessageConsumer consumer;

/* Établissement de la connection avec le serveur ActiveMQ */
connectionFactory = new ActiveMQConnectionFactory(
          /*nom d'administrateur, configurer dans le fichier conf */
          "aximag2",
          /*mot de passe, configurer dans le fichier conf */
          "iMAG04",
          /*protocole et adresse du serveur*/
          "tcp://46.105.41.94:61616");

/* Initialisation des délais temporels */
TIMEOUT = 180s ;
TIMESLEEP = 30s ;
TIMEWAIT = 1s ;

/* Préciser les contextes de session. */
try {
  connection = connectionFactory.createConnection();
  connection.start();

  /* On utilise ACKNOWLEDGE comme mode de session pour s'assurer du
  bon traitement de la requête */
  session = connection.createSession(Boolean.TRUE,
          Session.AUTO_ACKNOWLEDGE);

  /* On utilise Queue (file d'attente de messages) comme mécanisme de
  traitement et on précise la destination (Tradoh). Si Tradoh n'a pas de
  Queue associée sur le serveur, ActiveMQ la crée automatiquement.
  Ensuite ActiveMQ crée la connexion avec la Queue Tradoh, existante ou
  juste créée.*/
  destination = session.createQueue("Tradoh");

  /* Initialisation du « consommateur » */
  consumer = session.createConsumer(destination);

  /* Écouter en boucle infini */
  while (true) {
    /* Essayer de recevoir un message */
    Message message = consumer.receive();

    if (message != null) {

      /* On utilise un Timer pour éviter d'attendre un message « mort »*/
      Timer T1 = SystemTime ;

      /* Appeler le programme spécifique, et recevoir la confirmation
      de l'appel*/
      boolean res = call_Tradoh(message.getText()) ;

      /* Attendre le résultat avec une limite de temps.
      Il y a 3 situations : une bonne réponse, une mauvaise réponse, pas
      de réponse */
      while (true) {
        if(res){
          /*Recevoir un bon résultat, on envoie le symbole acknowledge */
          message.acknowledge();
          break ;
```

```
          }else if(res==null){
            Timer T2 = SystemTime ;
            /*On attend, si on n'a pas reçu la réponse dans TIMEOUT*/
            /*Si le temps est dépassé, on renonce au message. */
            if (T2-T1<TIMEOUT){
              sleep (TIMEWAIT) ;
            }else{
              break ;
            }
          }else { /*res==false*/
              /* mauvaise réponse, on ne peut pas recevoir la réponse de
              Tradoh*/
              break ;
          }
        }
      } else {
        /*S'endormir TIMESLEEP si pas de message*/
        sleep(TIMESLEEP) ;
        break;
      }
    }
  } catch (Exception e1) {
    Débug e1;
  }
}

/* Code spécifique pour Tradoh*/
call_Tradoh(Message msg){
  msg_type = msg.getType() ;
  if(msg_type==TextMessage){

    /* message est sous forme
    segment_source|langue_source|langue_cible|système_TA|id_segment */

    param = message.split("\\|");
    segment = param[0];
    ls = param[1];
    lc = param[TM/2];
    TA = param[3];
    id_segment = param[4];

    /* API de Tradoh*/
    url =
"http://46.105.41.94/tradoh/?text="+segment+"&source="+ls+"&target="+lc+"&M
T="+TA+"&piv=&verbose=0&outFormat=text&nocache=1";

    /* Appeler d'API et récupérer le résultat par la méthode get*/
    Document doc ;
    String segment_traduction;
    try{
      doc = Connect(url).timeout(TIMEOUT).get();
      segment_traduction = doc.body().text();
    }catch (Exception e){
      Débug e ;
      Return false ; /*mauvaise réponse*/
    }

    /*Vérifier si le message a besoin de renvoyer la réponse */

    Destination destination = msg.getReplayTo() ;
    if (destination != null){
```

```
      /*reconstruire le résultat sous la forme
      segment_cible|langue_source|langue_cible|système_TA|id_segment */

      message_traduction = segment_traduction|ls|lc|TA|id_segment ;
      Sender_Tradoh.sender(message_traduction, "TextMessage", destination);
    }
  }

  /*Traitement pour d'autres types de message ; dans nos premières
  implémentations, on utilise seulement le type TextMessage */
  /*
  if(msg_type==MapMessage){…}
  if(msg_type==BytesMessage){…}
  …
  */
}
```

L'algorithme du côté de l'application émettrice ressemble à l'algorithme ci-dessus, mais est plus simple, nous ne le détaillons pas.

3. Fichier des requêtes en source

```
Start Time : 2015/09/18 15:19:52
贸易和发展理事会|zho|fra|Google|1
商品和服务贸易及初级商品委员会|zho|fra|Google|2
专业服务和管理框架对贸易和发展 的影响专家会议|zho|fra|Google|3
2005 年 1 月 17 日至 19 日，日内瓦|zho|fra|Google|4
专业服务和管理框架对贸易和发展的 影响专家会议的报告|zho|fra|Google|5
2005 年 1 月 17 日至 19 日 在日内瓦万国宫举行|zho|fra|Google|6
目 录|zho|fra|Google|7
章 次 页 次|zho|fra|Google|8
一、主席的总结 3|zho|fra|Google|9
二、组织事项 26|zho|fra|Google|10
附 件|zho|fra|Google|11
出席情况 27|zho|fra|Google|12
第 一 章|zho|fra|Google|13
主席的总结|zho|fra|Google|14
1. 专业服务和管理框架对贸易和发展的影响专家会议于 2005 年 1 月 17 日至 19 日在日内瓦举行。
|zho|fra|Google|15
对于发达国家和发展中国家的政府如何能够在国家层次和多边谈判中为促进专业服务贸易发挥积极作用，专家们提出了自己的看法。|zho|fra|Google|16
贸发会议秘书处编写的背景说明“专业服务和管理框架对贸易和发展的影响”和题为“专业人员走出国界：互认协定与《关贸总协定》”的讨论文件受到了称赞。|zho|fra|Google|17
对于这些文件涉及的问题，各方表示了不同的意见。|zho|fra|Google|18
从下文中可以看出，表示的意见和提出的建议是丰富多样的。|zho|fra|Google|19
一、专业服务与发展|zho|fra|Google|20
全球市场的走向|zho|fra|Google|21
2. 全球所有专业服务市场的总额在 2002 年达到 10,000 亿美元以上。|zho|fra|Google|22
专业服务是全世界各经济体中增长最迅速的一个部门。|zho|fra|Google|23
这些服务中的就业增长速度超过了经济其他部门。|zho|fra|Google|24
发展中国家专业服务贸易的相对重要性与发达经济体不同，出现了下降，而发达国家的公司在所有的全球专业服务市场都占有主导地位。|zho|fra|Google|25
在公司和国家层次，正在出现许多专业服务集中化的趋势。|zho|fra|Google|26
同时，摆在专业服务提供者面前的是生产的全球化，区域贸易和南南贸易份量的不断增加以及竞争相关问题重要性的不断增加。|zho|fra|Google|27
虽然服务业对发展中国家国内生产总值的贡献不断加大，但在贸易总量中的份额仍然落在货物贸易的后面，与发达国家相比尤其如此。|zho|fra|Google|28
```

专业服务的作用|zho|fra|Google|29
3. 专业服务作为增强经济竞争力的一种基础设施服务发挥着重要的作用。|zho|fra|Google|30
这类服务是所有商业活动的投入，其质量和竞争力决定着外溢效应的规模。|zho|fra|Google|31
专业服务具有的一种关键发展影响在于对人们往往称之为“知识经济”的贡献。|zho|fra|Google|32
经济学者发现，人力资本、增加值和经济增长率之间存在着密切关联，而专业服务往往处在创新的前沿。|zho|fra|Google|33
这类服务，尤其是在发展中国家和最不发达国家，对于实现《千年宣言》的社会经济发展目标具有直接影响。|zho|fra|Google|34
另外，服务贸易问题与《千年宣言》的减贫承诺、目标 8 的实现以及建立全球发展伙伴关系的努力都直接相关。|zho|fra|Google|35
4. 专业服务包括经认证的专业，如律师、医生、会计师、建筑师和工程师，也包括未经认证的或自由操业专业。|zho|fra|Google|36
专业服务的范围和定义各国有所不同。|zho|fra|Google|37
多数发展中国家接受的是专业服务的一种广义定义，其中包括与土著产品贸易相联的新服务或与其他服务贸易联系在一起的服务，如海事服务，尤其是计算机和相关服务。|zho|fra|Google|38
例如，在卢旺达，除了传统的专业人员以外，银行人员和保险业人员也被认为是专业人员。|zho|fra|Google|39
专业服务贸易与发展中国家|zho|fra|Google|40
5. 发展中国家的经验清楚地显示了专业服务贸易可为其带来的发展收益。|zho|fra|Google|41
专业服务出口在 2002 年达到 2,700 亿美元，发展中国家占总出口的 15%。|zho|fra|Google|42
由于缺乏充分的分类数据，现在无法正确评估发展中国家专业服务出口的走势和贸易方向。|zho|fra|Google|43
看来，商业机会主要为大型跨国专业公司所用，这类公司存在于若干市场，与当地公司相联，是全球网络的一部分。|zho|fra|Google|44
在订有相互承认协定时，专业服务提供者提供服务就可得到便利。|zho|fra|Google|45
对于其他专业服务提供者来说，在国外提供服务的机会相当有限。|zho|fra|Google|46
所有服务提供者，包括中小型企业，都有利基市场，因为专业服务带来的是大量商机(例如，联合王国的专业服务占国内生产总值的 15%)，而且并不仅仅限于大型公司。|zho|fra|Google|47
对于多数发展中国家而言，专业服务出口的规模仍然不大，但在有些情况下实现了业绩突出的增长率。|zho|fra|Google|48
例如，哥伦比亚就属于这种情况，该国的商业服务出口在近些年来实现了 13%的年增长率，安第斯各国吸收了其总出口的 31%。|zho|fra|Google|49
另外，像巴西和印度这类发展中国家正在成为专业服务的全球角色。|zho|fra|Google|50
印度的外包公司就是成功融入世界市场的一个例子。|zho|fra|Google|51
全球服务外包预期到 2006 年将达到 1 万亿美元，其中只有 2%是国际间的交易。|zho|fra|Google|52
美国市场占印度外包的 60%，但这在美国服务总进口中只不过是 1%。|zho|fra|Google|53
印度的服务外包值从 1999-2000 年期间的 5 亿 6,500 万美元增加到了 2003-2004 年期间的 36 亿美元。|zho|fra|Google|54
联合王国已经成为印度外包服务的第二大市场，并且成为 480 家印度公司的业务基地，这些公司设在联合王国，以便进入欧盟市场。|zho|fra|Google|55
由此而来的两国之间双边投资流动近来已经达到了平衡状态。|zho|fra|Google|56
就巴西而言，服务总出口的 36%是商业和专业服务，其中大部分是工程和建筑服务。|zho|fra|Google|57
6. 经济规模小的发展中国家，如加勒比共同体各国和毛里求斯，往往依赖服务部门对国内生产总值和就业作出主要贡献。|zho|fra|Google|58
小岛屿发展中国家趋于保持一种开放型贸易制度，金融服务和旅游服务占有最大的贸易份额，是经济和发展的主要推动力量，而专业服务是这些部门的重要投入。|zho|fra|Google|59
小岛屿发展中国家的进一步贸易和发展必须明确无误地依赖于在国内生产总值中已经占三分之二以上的服务业。|zho|fra|Google|60
一些最不发达国家，如卢旺达，成功地保持了高经济增长率，其原因在很大程度上并不仅仅是电信、旅游和运输服务的壮大，而且还有包括咨询服务在内的专业服务的增长。|zho|fra|Google|61
新动态：外包|zho|fra|Google|62
7. 发展中国家外包产业的发展为其社会经济发展带来了直接或间接的收益。|zho|fra|Google|63

例如，在印度，预计到 2008 年在这一产业中就业的人将达到 200 万，按照系数为 3 的乘数效应，在次级就业中还将创造出更多的工作职位。|zho|fra|Google|64
外包方式通过在其他国家建立起专营店，推动了提供计算机教育培训的民办教育领域的贸易。|zho|fra|Google|65
这些服务已经发展到了提供基本的计算处理和更高级培训的程度，因为大学动作太慢，无法适应新技术的需求和针对迅速变化的市场作出反应。|zho|fra|Google|66
外包产业提供了薪金较高的工作岗位，带来了收入的增长，推动了消费。|zho|fra|Google|67
这一部门的年工资增长率为 15%。|zho|fra|Google|68
雇员中的高减员率表明，外包服务产业创造的技能是“便携式”的。|zho|fra|Google|69
因此，这些技能就能够向国内产业转移。|zho|fra|Google|70
电信法规改革及电信业自由化降低了电信服务的费用，这得到了全球外包产业需要的推动。|zho|fra|Google|71
为了加强在出口市场中的地位，外包服务业的公司力求得到设在自己出口市场的公司开出的高质量证明，以求保持产品的质量水准和满足西方国家客户的需要。|zho|fra|Google|72
由于这一服务产业中的人员有 40%至 60%是女性，外包服务通过提供过去几乎没有的更高增值就业机会对于提高女性地位有着突出的影响。|zho|fra|Google|73
这一部门还诞生出了印度的多国公司，既 TCS、Infosys 和 Wipro, 这些公司在范围广泛的服务中正在取得国际竞争力，并在全球扩张其业务活动。|zho|fra|Google|74
最后，信息和通信技术部门的成功对于打造印度品牌作出了贡献，在诸如旅游、娱乐、专业服务和其他服务的部门内产生了积极的外溢效应。|zho|fra|Google|75
贸易和发展基准|zho|fra|Google|76
8. 在专业服务领域，有可能制订贸易和发展基准(见下文)，在预期这些服务对发展所作贡献的基础上评估发展中国家是否正在切实有效地融入国际贸易体系和贸易谈判并从中获得益处。|zho|fra|Google|77
发展中国家专业服务的供应能力和竞争力。|zho|fra|Google|78
这方面的主要决定因素是高等和技术教育的质量和推广。|zho|fra|Google|79
与其他一些服务部门不同，这方面的比较优势在于成本、质量、竞争性劳动力的组合和在相关领域创造、应用和传播专门知识。|zho|fra|Google|80
在新兴的知识经济方面，除其他外，由商业咨询、信息技术服务、研究与发展、以及设计和工程服务构成的这类服务是竞争力的主要因素。|zho|fra|Google|81
由于发展中国家的中小企业规模不大，难以获得充足的资本和信贷，无法承担较大的商业风险，特别是不能像发达国家的中小型企业那样，在发展中国家的国家一级没有特别方案和公共支助措施，因此处于特别不利的地位。|zho|fra|Google|82
加大发展中国家在专业服务贸易中的份额和加强发展中国家对于国际贸易中新兴和有活力部门的参与。|zho|fra|Google|83
应当指出，虽然发展中国家不断加大了对于专业服务进口的参与，而且也加强了对于这方面出口的参与，但这些国家仍然是专业服务的净进口国。|zho|fra|Google|84
有出口的仅仅是少数发展中国家，并局限于传统的专业活动，主要是通过模式 4 和 1 实现的。|zho|fra|Google|85
但是，发展中国家的公司正在开发有竞争力的利基市场，如专业化的保健服务、包括生物技术在内的研发服务、计算机服务和工程服务。|zho|fra|Google|86
对发展中国家出口的市场开往程度及发展中国家市场对其他国家开放。|zho|fra|Google|87
澳大利亚生产力委员会近期的一项研究对于跨专业和跨国家的限制性程度作了比较，其中表明，认证专业的贸易在全世界都仍然面临着严重的限制。|zho|fra|Google|88
专业服务提供者在开业方面往往比已在运行的业务活动面临更多的限制。|zho|fra|Google|89
据指出，在认证专业中，法律服务最受限制，接下来是会计、建筑和工程服务。|zho|fra|Google|90
但受影响最严重的是发展中国家专业人员的流动问题，因为目前存在着市场准入和入境障碍，包括在注册、许可、核证、承认和认证方面的国内法规以及关于居住、国籍或公民权规定的影响。|zho|fra|Google|91
发达国家已经看到了定价和广告限制这类反竞争做法对其竞争力和服务出口产生的负面影响。|zho|fra|Google|92
另外，发达国家市场势力强大的有组织利害关系方实行的反竞争做法也带来了障碍，影响了发展中国家的供应者，除其他外，有可能抵消贸易自由化特别是为中小型企业和消费者带来的积极影响。

|zho|fra|Google|93
就发展中国家的自由化而言，虽然扶持国内供应能力和竞争力以及扩大基本服务准入的政策空间是尤其必要的，但技能转让、投资促进和并入全球网络是尽量扩大贸易对增长和发展的贡献的必要条件。
|zho|fra|Google|94
减贫、男女平等和福利、公共利益、创造就业、技能开发和增值、信通技术传播和技术创新。
|zho|fra|Google|95
专业服务与自己的精英形象正相反，它能帮助人民摆脱贫困，而高效率和得到推广的专业服务也就能够因此而对公共利益作出贡献，通过提供更好的保健和教育及社会服务等等减轻贫困的负面影响。
|zho|fra|Google|96
专业服务的发展在发展中国家是新的和更多的高薪及正规就业的来源。|zho|fra|Google|97
它是由技能发展所推动的，并且有助于提高货物、服务和贸易生产的增加值。|zho|fra|Google|98
多数发展中国家的专业服务有高度的性别敏感力和积极影响。|zho|fra|Google|99
例如，妇女在国内外都积极地从事护士、教师和社会工作者的服务。|zho|fra|Google|100
longueur moyenne 39.66
End Time : 2015/09/18 15:19:52

4. Prétraductions reçues

Conseil du commerce et du développement|zho|fra|Google|1
Commerce des marchandises et des services et produits de base|zho|fra|Google|2
Impact sur les réunions de commerce et d'experts du développement de services professionnels et de cadres réglementaires|zho|fra|Google|3
Le 17 janvier 2005 à 19, Genève|zho|fra|Google|4
Rapport des services professionnels et des cadres réglementaires pour l'impact du commerce et du développement de réunions d'experts|zho|fra|Google|5
Tenue au Palais des Nations 2005 年 17 à 19 Janvier|zho|fra|Google|6
SOMMAIRE|zho|fra|Google|7
Chapitre Page|zho|fra|Google|8
Tout d'abord, le résumé du Président 3|zho|fra|Google|9
Questions d'organisation 26|zho|fra|Google|10
ANNEXE|zho|fra|Google|11
Participation 27|zho|fra|Google|12
Premier chapitre|zho|fra|Google|13
Résumé du président|zho|fra|Google|14
L'impact sur le commerce et expert en développement réunion 1. Les services professionnels et des cadres réglementaires en 2005 17 Janvier au 19 à Genève.|zho|fra|Google|15
Pour le Gouvernement des pays développés et en développement sur la façon de promouvoir le commerce dans les services professionnels peuvent jouer un rôle actif au niveau national et des négociations multilatérales, les experts ont présenté leurs vues.|zho|fra|Google|16
Note d'information établie par le secrétariat de la CNUCED "services professionnels et l'impact des cadres de réglementation sur le commerce et le développement», et intitulé «professionnels là-frontières: accords de reconnaissance mutuelle avec le" "" document de travail du GATT a été salué.|zho|fra|Google|17
Pour toute question concernant ces documents, les parties ont exprimé des opinions différentes.|zho|fra|Google|18
Comme on peut le voir dans ce qui suit, les recommandations et les observations faites dans une variété de spectacles.|zho|fra|Google|19
Un service professionnel et le développement|zho|fra|Google|20
Les tendances du marché mondial|zho|fra|Google|21
2. La quantité totale de tous les marchés de services professionnels à travers le monde a atteint plus de 1 billion $ en 2002.|zho|fra|Google|22
Services professionnels est les économies du monde dans l'un des secteurs à plus forte croissance.|zho|fra|Google|23
Ce emploi dans les services a progressé plus vite que le reste de l'économie.|zho|fra|Google|24
L'importance relative du commerce des services professionnels dans les pays en développement et les pays développés, il a été diminué, tandis que les entreprises dans les pays développés sur tous les marchés mondiaux de services professionnels sont dominants.|zho|fra|Google|25
Dans les entreprises nationales et les niveaux, de nombreux services professionnels centralisés tendances

émergentes.|zho|fra|Google|26
Dans le même temps, placé en face d'un fournisseur de services professionnels est la mondialisation de la production, l'augmentation du commerce régional et l'augmentation des échanges Sud-Sud et les questions liées à la concurrence importance lourde.|zho|fra|Google|27
Bien que la contribution du secteur des services au PIB dans les pays en développement continuent d'augmenter, mais sa part dans le commerce total reste à la traîne commerce des marchandises, par rapport aux pays développés en particulier.|zho|fra|Google|28
Le rôle des services professionnels|zho|fra|Google|29
3. Les services professionnels afin d'améliorer la compétitivité économique comme un service d'infrastructure joue un rôle important.|zho|fra|Google|30
Ce service est mis toutes les activités commerciales, la qualité et la compétitivité de déterminer la taille des effets d'entraînement.|zho|fra|Google|31
Un impact sur le développement crucial des services professionnels réside dans la contribution des personnes ont souvent appelé «l'économie de la connaissance».|zho|fra|Google|32
Les chercheurs en économie ont trouvé que le capital humain, il existe une corrélation étroite entre la croissance économique et l'augmentation de la valeur, et les services professionnels sont souvent à la pointe de l'innovation.|zho|fra|Google|33
Ces services, en particulier dans les pays en développement et les moins avancés, pour obtenir la "Déclaration du Millénaire" des objectifs socio-économiques de développement a un impact direct.|zho|fra|Google|34
En outre, la réduction du commerce des services et de la pauvreté, de l'engagement "Déclaration du Millénaire", la mise en œuvre, et d'établir un partenariat mondial pour les efforts de développement sont directement liés à l'objectif 8.|zho|fra|Google|35
4. Les services professionnels comprennent des professionnels certifiés tels que les avocats, les médecins, les comptables, les architectes et les ingénieurs, ainsi que des non autorisées ou libres professionnels de l'industrie de l'exercice.|zho|fra|Google|36
La portée et la définition des services professionnels varient Unis.|zho|fra|Google|37
La plupart des pays en développement à accepter une définition large de services professionnels, y compris les nouveaux services et produits associés commerciales ou le commerce des liens indigènes avec d'autres services avec des services tels que les services maritimes, notamment informatiques et services connexes.|zho|fra|Google|38
Par exemple, au Rwanda, en plus de professionnels traditionnels, les banquiers et le personnel d'assurance sont considérés comme des professionnels.|zho|fra|Google|39
Professionnel commerce des services avec les pays en développement|zho|fra|Google|40
5. L'expérience des pays en développement montre clairement les avantages des services de développement des affaires professionnelles à sa portée.|zho|fra|Google|41
Les exportations de services professionnels ont atteint 270 milliards de $ en 2002, 15% des exportations totales des pays en développement.|zho|fra|Google|42
En raison de l'absence de données ventilées adéquates, nous ne pouvons pas évaluer correctement le sens de la tendance et le commerce des exportations de services professionnels des pays en développement.|zho|fra|Google|43
Opinion, les occasions d'affaires sont principalement utilisés par les grandes entreprises transnationales professionnels, ces entreprises existent dans un certain nombre de marchés, associée avec des entreprises locales, il fait partie d'un réseau mondial.|zho|fra|Google|44
Accords de reconnaissance mutuelle dans l'ordre, les fournisseurs de services professionnels pour fournir des services peuvent être facilitées.|zho|fra|Google|45
Pour les autres fournisseurs de services professionnels, la possibilité de fournir des services dans un pays étranger est assez limité.|zho|fra|Google|46
Tous les fournisseurs de services, y compris les PME, ont un marché de niche, car un grand nombre de services professionnels pour apporter des affaires (par exemple, les services professionnels Royaume-Uni représentaient 15% du PIB), mais ne se limite pas aux grandes entreprises.|zho|fra|Google|47
Pour la plupart des pays en développement, l'ampleur des exportations de services professionnels est encore petit, mais dans certains cas d'atteindre un taux de rendement exceptionnel de croissance.|zho|fra|Google|48
Par exemple, comme cela est le cas de la Colombie, les services commerciaux, les exportations du pays au cours des dernières années pour atteindre un taux de croissance annuel de 13%, les pays andins d'absorber 31% de ses exportations totales.|zho|fra|Google|49
En outre, les pays en développement comme le Brésil et l'Inde sont de plus le rôle mondial de services professionnels.|zho|fra|Google|50
Entreprises indiennes d'externalisation est un exemple d'intégration réussie dans le marché

mondial.|zho|fra|Google|51
Externalisation mondiale devrait à 2006 atteindra $ 1000000000000, dont seulement 2% a été négociée à l'échelle internationale.|zho|fra|Google|52
Le marché américain a représenté 60 pour cent de la sous-traitance de l'Inde, mais le service aux États-Unis dans les importations totales est seulement un pour cent.|zho|fra|Google|53
Inde valeur de la sous-traitance de 1999 - 500 millions de $ 65 millions au cours de la période de 2000 à 2003 - $ 3,6 milliards en 2004.|zho|fra|Google|54
Le Royaume-Uni est devenu le deuxième plus grand marché pour les services d'externalisation indiennes, 480 en Inde et est devenu la base de l'entreprise de la société, ces sociétés basées au Royaume-Uni, afin d'entrer dans le marché de l'UE.|zho|fra|Google|55
L'investissement bilatéral coule entre les deux pays résultant de l'équilibre a récemment atteint.|zho|fra|Google|56
Pour le Brésil, desservant 36 pour cent du total des exportations de services commerciaux et professionnels, dont la plupart sont d'ingénierie et de construction.|zho|fra|Google|57
6. La petite taille de l'économie des pays en développement, tels que les pays de la Communauté des Caraïbes et l'île Maurice, ont tendance à compter sur le secteur des services domestiques PIB et l'emploi à faire des contributions majeures.|zho|fra|Google|58
PEID ont tendance à maintenir un système commercial ouvert, les services financiers et les services touristiques représentent la plus grande part du commerce, est la principale force motrice pour le développement économique et, et les services professionnels sont des éléments importants dans ces secteurs.|zho|fra|Google|59
Davantage le commerce et le développement des petits États insulaires en développement doivent sans ambiguïté dépend du produit intérieur brut a plus que deux tiers des services.|zho|fra|Google|60
Certains des pays les plus développés, tels que le Rwanda, a réussi à maintenir un taux élevé de croissance économique, qui est en grande partie la raison est non seulement télécommunications, le tourisme et l'expansion des services de transport, y compris, mais également des services consultatifs, y compris les services professionnels croissance.|zho|fra|Google|61
Nouveaux développements: Externalisation|zho|fra|Google|62
7. Le développement de l'industrie de l'externalisation dans les pays en développement pour le développement social et économique a apporté des avantages directs ou indirects.|zho|fra|Google|63
Par exemple, en Inde, est attendu pour 2008, l'emploi dans cette industrie atteindra les 2 millions, selon le coefficient de l'effet multiplicateur de 3 dans l'emploi secondaire va créer plus d'emplois.|zho|fra|Google|64
L'externalisation par les magasins de franchise établis dans d'autres pays, afin de promouvoir le commerce fournir une éducation de formation en informatique et dans le domaine de l'enseignement privé.|zho|fra|Google|65
Ces services ont été développés dans la mesure de fournir l'informatique de base et une formation plus poussée, parce que l'université trop lent, incapable d'adapter aux besoins de la nouvelle technologie et de réagir contre un marché en évolution rapide.|zho|fra|Google|66
Outsourcing fournit un emploi salariaux élevés, ce qui porte la croissance des revenus, et de promotion de la consommation.|zho|fra|Google|67
La croissance annuelle des salaires dans ce secteur est de 15%.|zho|fra|Google|68
Les employés des taux d'attrition élevés suggèrent que l'externalisation industries pour créer compétences sont "portable" en.|zho|fra|Google|69
Par conséquent, ces compétences seront en mesure de transférer à l'industrie nationale.|zho|fra|Google|70
Télécommunications réforme de la réglementation et la libéralisation du secteur des télécommunications dans la réduction du coût des services de télécommunications, qui a été la promotion des besoins de l'industrie mondiale de l'externalisation.|zho|fra|Google|71
Afin de renforcer sa position dans le marché de l'exportation, des services d'impartition société a cherché à se mettre en place dans leurs marchés d'exportation de sociétés de haute qualité à prouver, en vue de maintenir les normes de qualité des produits et de répondre aux besoins des clients dans les pays occidentaux.|zho|fra|Google|72
En raison de ce service personnel de l'industrie ont 40-60 pour cent sont des femmes, en offrant des services à plus forte valeur ajoutée sous-traitance dans le passé presque pas de possibilités d'emploi pour améliorer le statut des femmes a une influence importante.|zho|fra|Google|73
Le ministère a également né de sociétés multi-nationales de l'Inde, à la fois TCS, Infosys et Wipro, ces entreprises une large gamme de services qui sont faits l'expansion compétitive au niveau international et mondial de ses activités commerciales.|zho|fra|Google|74
Enfin, l'information et le secteur de la technologie de communication pour la marque de construction réussie en Inde a contribué dans des secteurs tels que le tourisme, les loisirs, les services professionnels et autres services ont eu un effet d'entraînement positif.|zho|fra|Google|75

Repères de Commerce et développement|zho|fra|Google|76
8. Dans le domaine des services professionnels, la possibilité de développer des référentiels de commerce et de développement (voir ci-dessous), afin d'évaluer si les pays en développement sont effectivement intégrés dans les négociations relatives au système commercial et commerciales internationales sur la base de la contribution attendue au développement de ces services sur et en tirer des avantages .|zho|fra|Google|77
La capacité d'offre et la compétitivité des services professionnels dans les pays en développement.|zho|fra|Google|78
Le principal facteur déterminant à cet égard est la qualité de l'éducation et de promotion supérieur et technique.|zho|fra|Google|79
Contrairement à d'autres secteurs de services diffèrent, l'avantage comparatif à cet égard est que la combinaison de coût, de qualité, travail concurrentiel et les domaines connexes à la création, l'application et la diffusion de l'expertise.|zho|fra|Google|80
Dans la nouvelle économie du savoir, en plus d'autres choses, les services de conseil en affaires et technologie de l'information, recherche et développement, ainsi que des services de conception et d'ingénierie de tels services constituent un facteur important de la compétitivité.|zho|fra|Google|81
En raison de la petite taille des PME dans les pays en développement, il est difficile d'obtenir des capitaux et de crédit suffisant, ne peuvent se permettre les risques commerciaux plus importants, en particulier les petites et moyennes entreprises dans les pays développés ne peuvent pas être comme ça, pas de programmes spéciaux au niveau national dans les pays en développement et des mesures de soutien publiques, donc à un désavantage particulier.|zho|fra|Google|82
Augmenter la part des pays en développement dans les échanges de services professionnels et de renforcer la participation des pays en développement dans le commerce international dans des secteurs nouveaux et dynamiques.|zho|fra|Google|83
Il convient de noter que tandis que les pays en développement continuent d'accroître la participation des importations de services professionnels, mais aussi de renforcer la participation des exportations à cet égard, mais ces pays sont encore des importateurs nets de services professionnels.|zho|fra|Google|84
Il ya seulement quelques pays en développement, les exportations et les activités professionnelles traditionnelles limitées, principalement par le mode 4 et 1 mise en œuvre.|zho|fra|Google|85
Toutefois, les entreprises des pays en développement est de développer des marchés de niche concurrentiels, tels que les services de santé spécialisés, y compris la recherche en biotechnologie et des services de développement, de services informatiques et de services d'ingénierie.|zho|fra|Google|86
Pour les exportations des pays à destination du degré d'ouverture des marchés et le développement de marchés à d'autres pays.|zho|fra|Google|87
Une étude australienne a comparé la récente Commission de la productivité pour limiter l'ampleur de la multi-disciplinaire et de cross-country, ce qui montre que le commerce professionnel certifié dans le monde est toujours confronté à de sérieuses contraintes.|zho|fra|Google|88
Les prestataires de services sont souvent confrontés à en termes d'ouverture plus limitée que ce qui a été l'organisation d'activités d'affaires.|zho|fra|Google|89
Il a été noté que, dans le Certified Professional, les services juridiques les plus limitées, suivis par les services de comptabilité, d'architecture et d'ingénierie.|zho|fra|Google|90
Cependant, le plus touché est le flux de professionnels dans les pays en développement, car actuellement il ya l'accès aux marchés et de barrières à l'entrée, l'enregistrement, les licences, la certification, ainsi que la législation nationale sur la résidence, la nationalité ou la reconnaissance de la citoyenneté et de la certification, y compris dispositions de.|zho|fra|Google|91
Les pays développés ont déjà vu les prix et de publicité restrictions telles pratiques anti-concurrentielles sur la compétitivité et les exportations de services ont généré un impact négatif.|zho|fra|Google|92
En outre, les puissants marchés organisés des pays développés à mettre en œuvre les pratiques anti-concurrentielles des parties prenantes ont également apporté trouble qui affecte les fournisseurs dans les pays en développement, notamment, il est possible de compenser la libéralisation des échanges en particulier pour les petites et moyennes entreprises et Consumer impact positif.|zho|fra|Google|93
Sur la libéralisation des pays en développement, bien que le soutien aux capacités nationales d'approvisionnement et la compétitivité, et élargir l'accès aux services de base est un espace politique particulièrement nécessaire, mais le transfert des compétences, la promotion des investissements et de l'intégration dans le réseau mondial de maximiser les échanges sur la croissance et contribution au développement des conditions nécessaires.|zho|fra|Google|94
Réduction de la pauvreté, l'égalité des sexes et le bien-être, l'intérêt public, la création d'emplois, développement des compétences et la diffusion des TIC à valeur ajoutée et l'innovation technologique.|zho|fra|Google|95
Un service professionnel avec leur image élite contraire, elle peut aider les gens à sortir de la pauvreté, et la

haute efficacité et des services professionnels sera promu peut donc contribuer à l'intérêt public en fournissant de meilleurs soins de santé et l'éducation et les services sociaux, etc. atténuer les effets négatifs de la pauvreté.|zho|fra|Google|96
Développement de services professionnels dans les pays en développement sont sources d'emplois bien rémunérés et informel nouvelles et supplémentaires.|zho|fra|Google|97
Elle est entraînée par le développement des compétences, et aider à augmenter la valeur ajoutée des biens, des services et de la production commerciale.|zho|fra|Google|98
Services professionnels dans la plupart des pays en développement sont très sensibles au genre vigueur et l'impact positif.|zho|fra|Google|99
Par exemple, les femmes sont activement engagés dans des infirmières nationaux et étrangers, les enseignants et les travailleurs des services sociaux.|fra|Google|100
End Time : 2015/09/18 15:20:33

Annexe 10 Spécifications de LEXTOH

1. Configurations et descripteurs des outils utilisables

Notre fichier "RESSOURCES" permet de décrire les outils par langue ou par outil. Nous avons déjà montré un exemple de "ressource par outil" au IV.2.3.2.4. Voici un exemple de "ressource par langue".

```
                  <!-- Ressources connues de LEXTOH, par langue -->
<RESSOURCES dateCreation = "20140221" dateDerniereModif = "20150508"
            typeRessources = "parLangue">
                     <!-- Lemmatiseurs ou autres du français -->
  <LANGUE codeLG = "FRA">
    <LM codeLM = "XIP" codeEnt = "utf8" codeLGxip = "fr"
      transcription = "telQuel" sortiesPossibles = "liste"
      url = "https://open.xerox.com/Services/XIPParser/Pages/Using%20XIP"
      requeteType = "API" requete = "http://atoum.imag.fr/getalp/Services/
          Web/CREATDICO/callXip.php?lang=fra&text=$fichEnt "/>

    <LM codeLM = "Ariane_FR4" codeEnt = "utf8" codeLGariane = "FR1"
      transcription = "transIntermFr" sortiesPossibles = "arbre_A xml"
      url = "http://www.taranis-software.com" requeteType = "CURL"
      requete = "curl \"http://www.taranis-software.com/Heloise/Ying/test.
          php\" --data \"InputText=$fichEnt&CoupleLangue=FR4-$fichSort\""/>

    <LM codeLM = "XELDA" codeEnt = "utf8" codeLGxelda = "fr"
      transcription = "telQuel" sortiesPossibles = "liste xml"
      url = "www.xrce.xerox.com/About-XRCE/History/Historical-
          projects/XeLDA/" requeteType = "EXEC"
      requete = "xelda MorphoAnalysis plaintext $fichSort $fichEnt
          fra FST FSTPOSTag " />
    ...
  </LANGUE>

                    <!-- Lemmatiseurs ou autres de l'anglais -->
  <LANGUE codeLG = "ENG">
    <LM codeLM = "XELDA" codeEnt = "utf8" codeLGxelda = "en"
      transcription = "telQuel" sortiesPossibles = "liste xml"
      url = "www.xrce.xerox.com/About-XRCE/History/Historical-
          projects/XeLDA/" requeteType="EXEC"
      requete = "xelda MorphoAnalysis plaintext $fichSort $fichEnt
          eng FST FSTPOSTag " />
...
  </LANGUE>
</RESSOURCES>
```

D'autre part, un descripteur permet de configurer les paramètres fixés, par exemple, la langue de dialogue pour l'interaction (pour l'instant, il n'y a que le français), les statistiques pour voir les paramètres préférés pour chaque outil appelable.

```
                              <!-- Langue de dialogue -->
<LANGUE dateCreation = "20140221" dateDerniereModif = "20140515"
CodeLang = "fr" Mode="ACTIVE"/>
<!-- LANGUE dateCreation = "20140221" dateDerniereModif = "20140221" CodeLang =
"en" Mode=""--/>
<!-- LANGUE dateCreation = "20140221" dateDerniereModif = "20140221" CodeLang =
"zh" Mode=""--/>
                           <!-- Statistiques statiques -->
<Outil codeLM = "XELDA" nbrCall = "1138" >
  <LANGUE codeLang = "fr" typeSortie = "liste" nbrCall = "541"/>
  <LANGUE codeLang = "fr" typeSortie = "xml" nbrCall = "80"/>
  <LANGUE codeLang = "en" typeSortie = "liste" nbrCall = "462"/>
  <LANGUE codeLang = "en" typeSortie = "xml" nbrCall = "55"/>
<Outil/>
<Outil codeLM = "Ariane_FR4" nbrCall = "847" >
...
<Outil/>
```

2. Spécification des entrées

On a présenté les entrées de façon "naturelle" au IV.2.2.1. Voici la spécification des paramètres et des métadonnées concernant l'entrée. La syntaxe utilisée est celle d'ARIANE-G5 pour la déclaration des variables.

```
-CHA- text == (inDefaut = 'textIn'). ** chaîne qui peut avoir la syntaxe
d'un chemin.
-EXC- lang == (fra, zho, eng, deu, esp, por, pol,…). ** EXC : exclusive
-NEX- lemmat == (jieba, xelda, ariane-heloise, xip,…). ** NEX : non
exclusive, 1 ou plusieurs valeurs possibles dans la liste, ensemble vide =
lemmat0.
-EXC- output == (txt, json, xml).
-EXC- formalism == (reduit, graphe-Q).
-EXC- debug == (debug1). **debug0 est la valeur par défaut.
-NUM- window == (inDefaut = 'NumberIn'). ** NUM : number, fonctionne
uniquement pour le Delaf.
-EXC- form == (1,0).
```

3. Spécification des sorties

Comme on l'a présenté plus haut, LEXTOH permet de suivre le flot de traitement de la sortie native à la sortie finale par le mode avancé. Dans le mode normal, on affiche uniquement la sortie finale.

a. Sortie native

C'est le résultat original du lemmatiseur appelé, sans aucune modification. Pour le DELAF, la sortie native est une requête MYSQL. On la transmet dans un tableau pour l'afficher sur l'interface.

Voici deux sorties natives (XIP et DELAF) pour *bonjour, tu vas bien ?*.

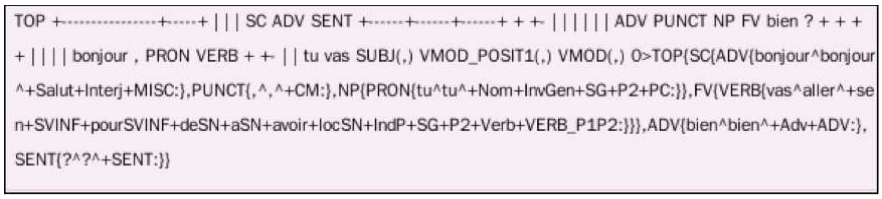


Figure 105 : Sortie native de XIP

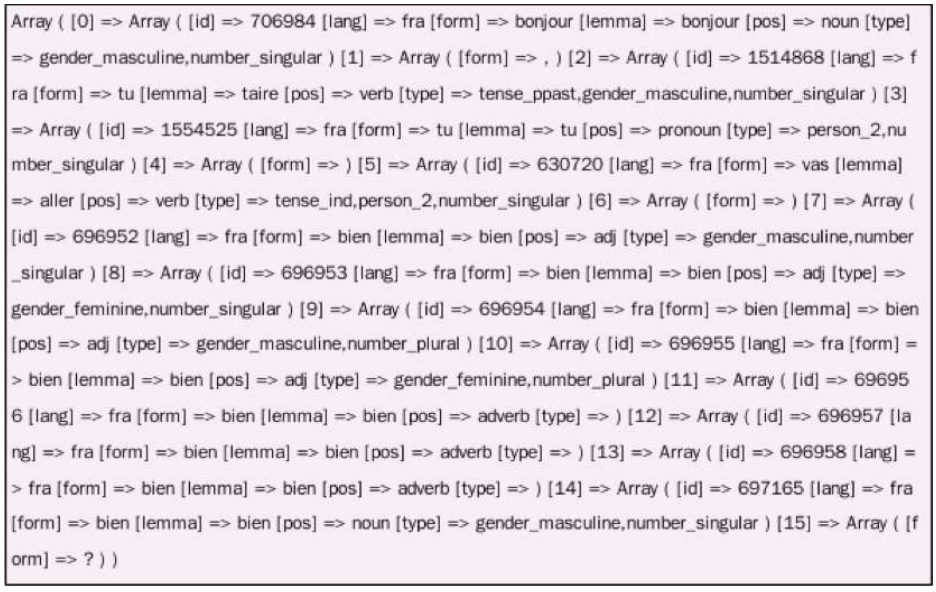


Figure 106 : Sortie native du DELAF

b. **Sortie brute**

La sortie brute est un tableau (`Array`). Dans cette étape, le système uniformise les clés (ex. `lang`, `form`, `pos`, `allTag`, `start` et `end`) pour l'étape suivante (sortie en forme `lemmatix`). Pour cette uniformisation, nous pouvons utiliser TRACOMPL pour décrire les règles de transformation. Pour l'instant, nous avons fait de façon ad hoc. Le système garde encore toutes les informations de la sortie native. Voir l'exemple ci-dessous.

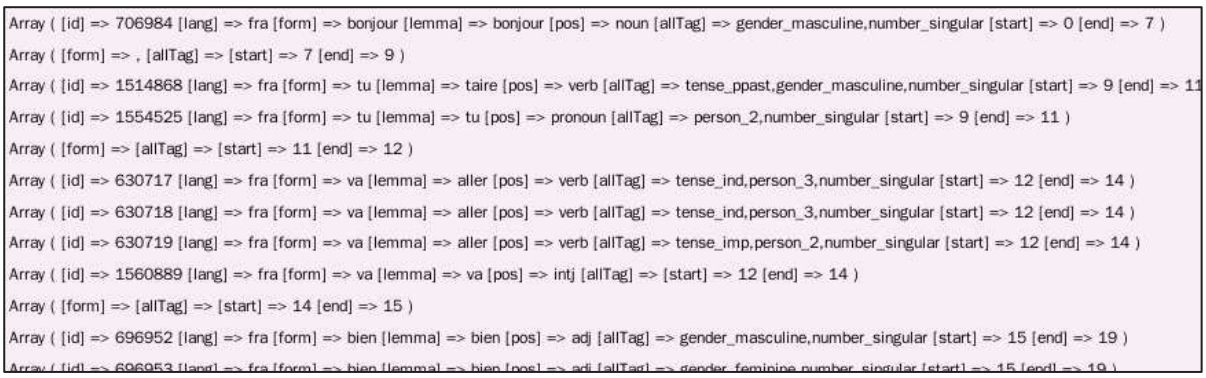


Figure 107 : Exemple d'une sortie brute

c. **Langage `Lemmatix`**

Avant de parler du format suivant (le format informatisé), on présente d'abord la forme (ou le langage) `Lemmatix`, utilisée dans LEXTOH, pour normaliser les sorties des différentes formes possibles. La normalisation porte sur les noms et les valeurs des attributs. La syntaxe de `Lemmatix` est décrite en XML. On présente sa DTD dans la figure suivante.

```
<!DOCTYPE LEMMATIX [
<!ELEMENT LEMMATIX (lexeme*)>
<!ELEMENT lexeme (entry*)>
<!ELEMENT entry (lemma,pos,morpho-tag-list)>
<!ELEMENT lemma (#PCDATA)>
<!ELEMENT pos (#PCDATA)>
<!ELEMENT morpho-tag-list (morpho-tag?)>
<!ELEMENT morpho-tag (#PCDATA)>

<!ATTLIST LEMMATIX syst CDATA #REQUIRED>
<!ATTLIST LEMMATIX lang CDATA #REQUIRED>
<!ATTLIST lexeme startpos CDATA #REQUIRED>
<!ATTLIST lexeme endpos CDATA #REQUIRED>
<!ATTLIST lexeme form CDATA #REQUIRED>
<!ATTLIST lemmat confidence CDATA #IMPLIED>
<!ATTLIST lemmat type CDATA #IMPLIED>
<!ATTLIST pos confidence CDATA #REQUIRED>
]>
```

Figure 108 : LEMMATIX.DTD

Pour des raisons d'implémentation informatique, on a défini deux sous-formes. La sous-forme `lemmatix intermédiaire` et la sous-forme `lemmatix finale` sont toutes les deux conformes à `lemmatix.dtd`.

d. **Sortie en `lemmatix intermédiaire`**

Les valeurs de la forme intermédiaire sont des valeurs directes qui sont extraites des différents lemmatiseurs. Par exemple :

`GNR(MASC,FEM)` → `<morpho-tag>MASC,FEM<morpho-tag>.`

Figure 109 : Deux exemples de `lemmatix intermédiaire`

e. **Sortie en `Lemmatix finale`**

Les valeurs de la forme finale sont des valeurs traitées et uniformisées des différents lemmatiseurs. Par exemple :

`<pos>+nom</pos>` → `<pos>noun<pos>.`

Pour unifier les résultats de différents analyseurs morphologiques, nous avons utilisé le standard ISOcat DCR [ISOcat, 2008].

f. **Sortie finale (résultat de filtrage)**

Le système produit différentes sorties finales. Chaque sortie finale est une combinaison de format+formalisme. Voir le paragraphe Entrées et Sorties au IV.2.2.1.

Annexe 11 Spécifications de CREATDICO

La conception de CREATDICO ressemble beaucoup à celle de LEXTOH.

1. Configurations et descripteurs des outils utilisables

Comme pour LEXTOH, on a un fichier "RESSOURCE" pour décrire les outils utilisables par langue ou par outil. Par exemple :

```
            <!-- Ressources connues de CREATDICO, par langue -->
<RESSOURCES dateCreation = "20140222" dateDerniereModif = "20151208" Auteur
          = "Ying" Accessible = "Public">
                    <!-- Services de dictionnaires -->
  <SERVEURS>
    <SERV codeServ = "Papillon" codeEnt = "utf8" callType="API" url =
      "http://www.papillon-dictionary.org/papillon/api/$dico/$lang/cdm-
      headword/$headword/cdm-translation?strategy=EQUAL" />
    <SERV codeServ = "Pivax" codeEnt = "utf8" callType="API" url =
     "http://getalp.imag.fr/pivax/api/$dico/$lang/link/manger
      ?strategy=EQUAL" />
    <SERV codeServ = "Pivax" codeEnt = "utf8" callType="EXEC" url = "
      http://iate.europa.eu" requete="php -f Robodico.PHP LS=$ls LC=$lc
      lemme=$text"  />
  </SERVEURS>
                              <!-- LANGUES -->
  <LANGUE codeLG = "ZHO">
    <DICO codeServ = "Papillon" codeDico = "Cedict" nbrEntries = "215424"/>
  </LANGUE>
  <LANGUE codeLG = "ENG">
    <DICO codeServ = "Papillon" codeDico = "Cedict" nbrEntries = "107712"/>
    <DICO codeServ = "Papillon" codeDico = "DicoFulUS" nbrEntries =
      "9997"/>
    <DICO codeServ = "IATE" codeDico = "IATE" nbrEntries = "1282712"/>
  </LANGUE>
</RESSOURCES>
```

Comme présenté au IV.3.2, nous avons un petit bloc de configuration pour chaque sortie dédiée. Cette configuration permet de spécifier les préférences. Par exemple, pour SECTRA, nous préférons utiliser XIP comme outil de lemmatisation pour le français et l'anglais. Nous avons déjà montré un exemple dans la Figure 92.

2. Spécification des entrées

Nous avons mentionné les entrées au IV.3, voici la spécification.

```
-CHA- text == (inDefaut = 'textIn').  ** -CHA- chaîne de caractères.
-EXC- output == (MS, GS, JSON, CCDX, DQ, etc.). ** excusive.
-EXC- detail == (0,1).
-EXC- lang == (zho, fra, eng, etc.).
-NEX- lc == (zho, fra, eng,etc.). **non excusive, 1 ou plusieurs.
-EXC- serv == (Pivax, Papillon, IATE, wiktionary, etc.).
-EXC- dico == (Papillon:Cedict, Papillon:JMDICT, Papillon:Littre, etc.).
-EXC- debut == (0,1).
-EXC- form == (0,1).
-EXC- lemmat == (Jieba, Ariane-Heloise, etc.). ** Outils dispon sur Lextoh.
```

3. Spécification du langage CCDX (Common CreatDico Xml)

Comme LEMMATIX pour LEXTOH, nous avons utilisé le langage CCDX (Common CreatDico Xml) pour uniformiser et informatiser les différentes sorties natives. Cela nous permet de simplifier les programmes pour construire une nouvelle sortie dédiée (mini-dictionnaire). On transforme d'abord les sorties natives vers les sorties en CCDX, puis on écrit des programmes simples

(plugins) pour transformer les sorties en CCDX vers une sortie dédiée. Voici le fichier CCDX.DTD.

```
<!DOCTYPE CCDX [
<!ELEMENT CCDX (description, dictionnaires*)>
<!ELEMENT description (entree, lemmatiseur?)>
<!ELEMENT entree (#PCDATA)>
<!ELEMENT lemmatiseur (#PCDATA)>
<!ELEMENT dictionnaires (forme, dictionnaire*)>
<!ELEMENT forme (#PCDATA)>
<!ELEMENT dictionnaire (lemme, traductions*)>
<!ELEMENT lemme (#PCDATA)>
<!ELEMENT traductions (traduction*)>
<!ELEMENT traduction (mot-vedette, typeMot?, pronom?, domaine?, def?, pos?,
fiab?, ref?, exemple?, commentaire?)>
<!ELEMENT mot-vedette (#PCDATA)>
<!ELEMENT typeMot (#PCDATA)>
<!ELEMENT pronom (#PCDATA)>
<!ELEMENT domaine (#PCDATA)>
<!ELEMENT def (#PCDATA)>
<!ELEMENT pos (#PCDATA)>
<!ELEMENT fiab (#PCDATA)>
<!ELEMENT ref (#PCDATA)>
<!ELEMENT exemple (#PCDATA)>
<!ELEMENT commentaire (#PCDATA)>

<!ATTLIST CCDX serv CDATA #REQUIRED>
<!ATTLIST CCDX dico CDATA #REQUIRED>
<!ATTLIST CCDX langueSource CDATA #REQUIRED>
<!ATTLIST entree type CDATA #REQUIRED>
<!ATTLIST traductions lang CDATA #REQUIRED>

]>
```

Voici un exemple de CCDX.

segment source = porte-bébé, langue source = fra, langues cibles = eng, esp, outil de lemmatisation = xelda, service dictionnairique = iate.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CCDX SYSTEM "CCDX.dtd">
<CCDX serv="iate" dico="iate" langueSource="fra">
    <description>
        <entree type="txt">porte-bébé</entree>
        <lemmatiseur>xelda</lemmatiseur>
    </description>
    <!--un forme pour un dictionnaire-->
    <dictionnaires>
        <forme>porte</forme>
        <!--un lemme pour un dictionnaire-->
        <dictionnaire>
            <lemme>porte</lemme>
            <!--les traductions d'une langue sont dans un élément « traductions »-->
            <traductions lang="eng">
                <!-- traduction -->
                <traduction>
                    <mot-vedette>door</mot-vedette>
                    <pos>NOUN</pos>
                    <fiab>3</fiab>
                    <ref_tra>The Land Rover Dictionary</ref_tra>
                </traduction>
                <traduction>
                    <mot-vedette>door</mot-vedette>
                    <domaine>Bâtiment et travaux publics</domaine>
                    <def>A building component consisting of an object in the form of a plate,
the door leaf, normally equipped with a frame with which an opening for passage may be closed;
```

```
normally the passage is shaped so as to accommodate persons in the upright position</def>
                    <ref_def>Ing.P.Lassen</ref_def>
                    <fiab>3</fiab>
                    <ref_tra>Ing.P.Lassen</ref_tra>
                    <commentaire>Door usually refers to a door leaf. The word doorset means
door leaf plus door frame(S). Swedish and Danish make a difference between doors for normal
traffic (persons) and doors for e.g. motor vehicle traffic etc. (see gate)</commentaire>
                </traduction>
                <traduction>
                    <mot-vedette>gateway</mot-vedette>
                    <fiab>3</fiab>
                    <ref_tra>Reg. 3381/94 (1) OJ L 367/94 p.104 Notice (2) OJ C 82/90
p.12</ref_tra>
                </traduction>
                <traduction>
                    <mot-vedette>gate</mot-vedette>
                    <def>a region in which the electric field due to the control gate voltage
is effective</def>
                    <ref_def>IEV 521-7-9</ref_def>
                    <fiab>3</fiab>
                    <ref_tra>IEV 521-7-9</ref_tra>
                    <commentaire>of field-effect transistors; REF:IEV 521-7-9</commentaire>
                </traduction>
            </traductions>
            <traductions lang="spa">
                <traduction>
                    <mot-vedette>pasarela</mot-vedette>
                    ....
                </traduction>
                <traduction>
                    <mot-vedette>puerta</mot-vedette>
                    ....
                </traduction>
            </traductions>
        </dictionnaire>
        <!--autre lemme de porte-->
        <dictionnaire>
            <lemme>porter</lemme>
            <traductions lang="fra">
            ...
            </traductions>
            <traductions lang="spa">
            ...
            </traductions>
        </dictionnaire>
    </dictionnaires>
    <dictionnaires>
        <forme>bébé</forme>
        <traductions lang="eng">
            <traduction>
                <mot-vedette>baby</mot-vedette>
                ....
            </traduction>
        </traductions>
        <traductions lang="spa">
            ....
        </traductions>
    </dictionnaires>
</CCDX>
```

4. Spécification des formats de sortie

a. Sorties générales

Le système a trois formats pour les sorties générales : JSON, XML-CCDX et txt (chaîne de caractères). On peut aussi en utiliser deux supplémentaires si on a besoin d'informations détaillées. Au total, on a 5 types de sortie : TEXTE-SIMPLE, TEXTE-DETAILLE, XML-CCDX (il n'existe pas de sortie "simple" pour ce format), JSON-SIMPLE et JSON-DETAILLE (pas encore réalisé).

On a déjà montré des exemples (interfaces) au IV.3.3.1.

La sortie en TEXTE-SIMPLE est sous la forme ci-dessous.

```
Lemme = LemmeSource Translation = Trans1(Lang) Trans2(Lang)
```

Souvent, la longueur des informations détaillées est très grande. On a décidé d'afficher la sortie en TEXTE-DETAILLE en deux étapes. La première étape est des textes bruts sous la forme

```
Lemme = LemmeSource, part-of-speech de lemme = POS,
Translation = Trans1, langue de traduction = Lang1.
```

Voici les détails de traduction (en XML).

La deuxième étape est le nœud XML correspondant.

Voici un exemple pour la sortie en JSON-SIMPLE.

```
Entrée : hello word
var JSONObject=[
{"serv":"wiktionary", "dico":"wiktionary", "lang":"eng", "lc":"fra&esp", "lemmat":"xip",
"sortie":"JSON", "detail":"non"},
{"lemme":"hello", "translation"="bonjour", "langue"="fra"},
{"lemme":"hello", "translation"="salut", "langue"="fra"},
{"lemme":"hello", "translation"="allô", "langue"="fra"},
{"lemme":"hello", "translation"="hola", "langue"="esp"},
...
]
```

Pour les exemples de XML-CCDX, voir la section précédente.

b. Sorties dédiées

Les sorties dédiées (plugins) sont construites principalement par les utilisateurs, on en a montré un exemple pour SECTRA au IV.3.2.1.

Nous présentons maintenant les contraintes techniques pour le développement des plugins.

5. Contraintes techniques pour le développement

En quel langage les plugins pour les sorties dédiées sont-ils écrits ? C'est souvent la première question posée par nos utilisateurs. Notre intergiciel est principalement construit en PHP. Les premiers mini-dictionnaires sont également construits en PHP.

En effet, la construction de mini-dictionnaires concerne principalement des transformations de données en XML-CCDX vers un autre format demandé par le client, par exemple, le renommage des balises pour un nouveau XML. Pour cette partie, nous avons déjà donné des explications détaillées au IV.3.3.2.2c.

Pour des raisons informatiques, on doit avoir un fichier PHP pour chaque sortie dédiée dans le répertoire préfixé : CREATDICO/_PLUGINS/. L'entrée de ce fichier est une sortie générale (souvent XML-CCDX) de CREATDICO. La sortie de ce fichier est la sortie dédiée. Nous n'avons pas de contraintes spéciales pour utiliser tel ou tel langage pour réaliser les spécifications.

Par exemple, nous pouvons écrire un programme PYTHON pour ces spécifications. Pour l'intégrer dans CREATDICO, on procède en deux étapes : (1) créer le fichier PHP dans le répertoire préfixé, (2) appeler le programme PYTHON en utilisant le programme suivant.

```
//Entrée:$XML_CCDX
//Sortie:$n_sortie nouvelle sortie dédiée
$command = "python /home/ying/lingOutils/minidico/minidico_python.py $XML_CCDX ";
$n_sortie = exec($command);
return $n_sortie;
```

De la même façon, nous pouvons également réaliser les transformations par un service d'API. Comme XML_CCDX est long, on a proposé de l'enregistrer dans un fichier. Notre fichier PHP de plugin sera similaire au programme suivant.

```
//Entrée:$XML_CCDX
//Sortie:$n_sortie nouvelle sortie dédiée
$file = fopen("temp_file.xml") or die ("Unable to open file");
fwrite($file, $XML_CCDX);
$url = "http://localhost/Ci-Hai/minidico_test/?Sortie_Orig=".urlencode($file);
$n_sortie = file_get_contents($url);
fclose($file);
return $n_sortie;
```

Dans nos intergiciels LEXTOH et CREATDICO, nous avons procédé de cette façon pour récupérer les résultats des différents services.

Par exemple,

- dans LEXTOH,
  - l'appel à ARIANE-HELOÏSE est réalisé par un plugin appelant `curl`.
  - l'appel à JIEBA est réalisé par un plugin écrit en PYTHON.
- dans CREATDICO,
  - l'appel à PAPILLON est réalisé par un plugin contenant une commande de l'API.

Résumé en français

Notre recherche se situe en lexicographie computationnelle, et concerne non seulement le support informatique aux ressources lexicales utiles pour la TA (traduction automatique) et la THAM (traduction humaine aidée par la machine), mais aussi l'architecture linguistique des bases lexicales supportant ces ressources, dans un contexte opérationnel (thèse CIFRE avec L&M).
Nous commençons par une étude de l'évolution des idées, depuis l'informatisation des dictionnaires classiques jusqu'aux plates-formes de construction de vraies "bases lexicales" comme JIBIKI-1 [Mangeot, M. et al., 2003 ; Sérasset, G., 2004] et JIBIKI-2 [Zhang, Y. et al., 2014]. Le point de départ a été le système PIVAX-1 [Nguyen, H.-T. et al., 2007 ; Nguyen, H. T. & Boitet, C., 2009] de bases lexicales pour systèmes de TA hétérogènes à pivot lexical supportant plusieurs volumes par "espace lexical" naturel ou artificiel (UNL). En prenant en compte le contexte industriel, nous avons centré notre recherche sur certains problèmes, informatiques et lexicographiques.
Pour passer à l'échelle, et pour profiter des nouvelles fonctionnalités permises par JIBIKI-2, dont les "liens riches", nous avons transformé PIVAX-1 en PIVAX-2, et réactivé le projet GBDLEX-UW++ commencé lors du projet ANR TRAOUIERO, en réimportant toutes les données (multilingues) supportées par PIVAX-1, et en les rendant disponibles sur un serveur ouvert.
Partant d'un besoin de L&M concernant les acronymes, nous avons étendu la "macrostructure" de PIVAX en y intégrant des volumes de "prolexèmes", comme dans PROLEXBASE [Tran, M. & Maurel, D., 2006]. Nous montrons aussi comment l'étendre pour répondre à de nouveaux besoins, comme ceux du projet INNOVALANGUES. Enfin, nous avons créé un "intergiciel de lemmatisation", LEXTOH, qui permet d'appeler plusieurs analyseurs morphologiques ou lemmatiseurs, puis de fusionner et filtrer leurs résultats. Combiné à un nouvel outil de création de dictionnaires, CREATDICO, LEXTOH permet de construire à la volée un "mini-dictionnaire" correspondant à une phrase ou à un paragraphe d'un texte en cours de "post-édition" en ligne sous IMAG/SECTRA, ce qui réalise la fonctionnalité d'aide lexicale proactive prévue dans [Huynh, C.-P., 2010]. On pourra aussi l'utiliser pour créer des corpus parallèles "factorisés" pour construire des systèmes de TA en MOSES.

Abstract in English

Our research is in computational lexicography, and concerns not only the computer support to lexical resources useful for MT (machine translation) and MAHT (Machine Aided Human Translation), but also the linguistic architecture of lexical databases supporting these resources in an operational context (CIFRE thesis with L&M).
We begin with a study of the evolution of ideas in this area, since the computerization of classical dictionaries to platforms for building up true "lexical databases" such as JIBIKI-1 [Mangeot, M. et al., 2003 ; Sérasset, G., 2004] and JIBIKI-2 [Zhang, Y. et al., 2014]. The starting point was the PIVAX-1 system [Nguyen, H.-T. et al., 2007 ; Nguyen, H. T. & Boitet, C., 2009] designed for lexical bases for heterogeneous MT systems with a lexical pivot, able to support multiple volumes in each "lexical space", be it natural or artificial (as UNL). Considering the industrial context, we focused our research on some issues, in informatics and lexicography.
To scale up, and to add some new features enabled by JIBIKI-2, such as the "rich links", we have transformed PIVAX-1 into PIVAX-2, and reactivated the GBDLEX-UW++ project that started during the ANR TRAOUIERO project, by re-importing all (multilingual) data supported by PIVAX-1, and making them available on an open server.
Hence a need for L&M for acronyms, we expanded the "macrostructure" of PIVAX incorporating volumes of "prolexemes" as in PROLEXBASE [Tran, M. & Maurel, D., 2006]. We also show how to extend it to meet new needs such as those of the INNOVALANGUES project. Finally, we have created a "lemmatisation middleware", LEXTOH, which allows calling several morphological analyzers or lemmatizers and then to merge and filter their results. Combined with a new dictionary creation tool, CREATDICO, LEXTOH allows to build on the fly a "mini-dictionary" corresponding to a sentence or a paragraph of a text being "post-edited" online under IMAG/SECTRA, which performs the lexical proactive support functionality foreseen in [Huynh, C.-P., 2010]. It could also be used to create parallel corpora with the aim to build MOSES-based "factored MT systems".