RECITAL 2004, Fès, 21 avril 2004

Un système adaptable pour l'initialisation automatique d'une base lexicale interlingue par acceptions

Aree Teeraparbseree
GETA, CLIPS, IMAG (UJF & CNRS)
385, rue de la Bibliothèque
B.P. 53 - 38041 Grenoble Cedex 9, France
aree.teeraparbseree@imag.fr

Mots-clefs – Keywords

base lexicale multilingue, construction automatique de lexies et axies, acception interlingue
multilingual lexical database, automatic building of lexies and axies, interlingual acception

Résumé - Abstract

Cet article présente une stratégie de construction semi-automatique d'une base lexicale interlingue par acception, à partir de ressources existantes, qui utilise en synergie des techniques existantes de désambiguïsation. Les apports et limitations de chaque technique sont présentés. Notre proposition est de pouvoir composer arbitrairement des techniques, en fonction des ressources disponibles, afin d'obtenir une base interlingue de la qualité souhaitée. Jeminie, un système adaptable qui met en oeuvre cette stratégie, est introduit dans cet article.

This article presents a strategy for the semi-automatic building of an interlingual lexical database, based on existing resources, and using existing disambiguisation techniques in synergy. The pros and cons of each technique are given. Our proposal is to be able to compose techniques arbitrarily, according to the available resources, in order to produce an interlingual database of the desired quality. Jeminie, an adaptable system that implements this strategy, is introduced in this article.

Introduction

Le travail présenté ici se place dans le cadre du projet Papillon (Mangeot-Lerebours *et al.*, 2003). Ce projet vise à la construction collaborative d'une grande base lexicale multilingue. La motivation initiale du projet est le manque de dictionnaires, à la fois pour humains et machines, entre le français et de nombreuses langues asiatiques. La construction d'une base lexicale multilingue nécessite de réunir des linguistes spécialistes de toutes les langues concernées par la base,

ce qui peut être difficile voire impossible. Des logiciels d'aide automatique à la construction de bases lexicales s'avèrent donc nécessaires.

Nous proposons dans cet article un système logiciel d'aide à la construction d'une base lexicale interlingue par acceptions, et plus précisément à la production d'un "premier jet" d'une organisation des vocables en *lexies* (sens de mot) et à la transformation des liens traductionnels entre vocables en liens interlingues (*axies*) entre lexies. Cette première version de la base est ensuite corrigée et améliorée par les linguistes.

Cet article présente d'abord la structure de la base lexicale du projet Papillon, puis une stratégie générale et des techniques automatiques utilisables pour la construction d'une base lexicale interlingue, et enfin Jeminie, un système adaptable pour la construction de telles bases.

1 Papillon : base lexicale interlingue par acceptions

Le projet Papillon[1] a pour but de créer une base lexicale multilingue comprenant l'allemand, l'anglais, le français, le japonais, le lao, le malais, le thaï, le vietnamien et, très récemment, le chinois. Papillon utilise l'approche interlingue, où un langage intermédiaire artificiel (dit aussi langage pivot) est utilisé pour établir des liens entre les langues. Une telle approche a été utilisée dans d'autres projets, par exemple, le projet KBMT-89 (Goodman & Nirenburg, 1991) et le projet ULTRA (Farwell *et al.*, 1992). La macrostructure de Papillon repose sur une structure pivot définie dans (Sérasset, 1994). Une "base Papillon" est composée d'un ensemble d'acceptions monolingues (*lexie*s) pour chaque langue et d'un ensemble de liens interlingues qui relient les lexies. Ces liens interlingues peuvent aussi être reliés à une liste d'éléments de chaque système extérieur qui sont plus ou moins équivalents ou voisins (synsets de WordNet, universal words d'UNL, ...). Cette structure a déjà été expérimentée dans PARAX (Blanc, 1999). La microstructure (structure lexicale des entrées de chaque dictionnaire) est tirée des travaux de Polguère sur le format lexicographique DiCo, qui est une simplification de celui du Dictionnaire Explicatif et Combinatoire associé à la théorie sens-texte (Mel'cuk *et al.*, 1995).

La construction du contenu se fera en deux phases (Mangeot-Lerebours *et al.*, 2003). (1) La *phase d'amorçage* a pour but d'obtenir, à partir de dictionnaires existants, une première base lexicale contenant de nombreuses entrées associées à des informations minimales. Puis dans (2) la *phase de contribution*, des internautes bénévoles coopèrent à travers Internet pour apporter des modifications (ajout, correction ou suppression) à la base lexicale du serveur Papillon. Cette phase ne peut commencer que lorsqu'il existe un ensemble minimal d'informations lexicales sur la base du serveur Papillon. Le travail présenté ici concerne la phase d'amorçage, et plus particulièrement son automatisation partielle ou complète.

2 Construction d'acceptions interlingues

Pour mener à bien ce travail, nous disposons de plusieurs dictionnaires monolingues (4000 entrées françaises DiCo de l'Université de Montréal, 10 000 entrées thaï de l'Université Kasetsart), de dictionnaires bilingues (70 000 entrées japonais-anglais et 10 000 entrées japonais-français

[1] La base est accessible principalement sur le site *http ://www.papillon-dictionary.org/*. Ce site contient aussi un collection de dictionnaires "classiques", accessibles par une interface unifiée.

de JMDICT en format XML de J. Breen, 8000 entrées japonais-thaï de SAIKAM) et de dictionnaires multicibles (50 000 entrées anglais-français-malais de FeM).

Nous proposons d'initialiser la "base Papillon" en deux étapes : (1) pour chaque dictionnaire monolingue on extrait l'information des vocables, par exemple les lemmes (simples ou composés), les parties du discours et les définitions, et on *crée des lexies* ; (2) à partir des lexies de plusieurs langues et des ressources bilingues et multilingues, on *crée des axies* qui relient ces lexies et on fait évoluer si nécessaire l'ensemble des lexies de chaque langue. Si les bases monolingues sont volumineuses, le temps nécessaire pour relier à la main ces bases monolingues en liens interlingues sera très élevé. De plus, le travail de création des axies est très coûteux. Il nécessite des personnes connaissant de nombreuses langues. Notre objectif est de développer un logiciel pour aider à la construction de ces liens afin de réduire le coût de construction. Plus précisément, dans l'étape (1) le logiciel automatisera la construction des lexies, mais il faudra quand même filtrer les lexies à la main pour vérifier leur qualité ; dans l'étape (2) le logiciel sera semi-automatique car la construction est automatique mais l'utilisateur devra tout de même paramétrer l'outil en fonction des ressources et de la qualité désirée, et du coût des traitements. Le problème principal est de relier automatiquement des lexies qui ont la même signification. Cette section décrit donc quelques techniques existantes de désambiguïsation que nous considérons pour la construction automatique des axies, et les problèmes liés à chaque technique.

La technique de **traduction bilingue (TR_BILING)** utilise un ou plusieurs dictionnaires bilingues pour créer des axies qui relient toutes les lexies de deux langues dont les mots sont la traduction l'un de l'autre. Les axies obtenues devraient être filtrées ultérieurement en utilisant d'autres techniques pour éventuellement dégrouper une axie en plusieurs axies ou enlever des mauvaises lexies dans les axies. La technique de **traduction bilingue multiple (TR_MBILING)** est une technique de traduction bilingue qui se base sur plusieurs dictionnaires bilingues, p.ex. japonais-anglais et anglais-français et leurs dictionnaires inverses, pour compléter ou pallier l'absence de dictionnaires bilingues entre deux langues, p.ex. entre le japonais et le français (Tanaka & Umemura, 1994). Cependant, ces deux techniques produisent une base lexicale au niveau des mots et pas du sens des mots. La technique de **comparaison de deux vecteurs conceptuels (CMP_VECTOR)** associe à chaque lexie un vecteur de concepts, appellé *vecteur conceptuel*, dont chaque dimension correspond à un concept feuille d'un thésaurus. M. Lafourcade (Lafourcade, 2002; Mangeot-Lerebours *et al.*, 2003) expérimente pour le français et l'anglais, en se basant sur les 873 concepts feuilles du thésaurus Larousse. La distance angulaire entre vecteurs mesure la proximité thématique entre lexies. La difficulté est de trouver des thésaurus pour ces langues et de faire correspondre les concepts feuilles de ces thésaurus. Par contre, le résultat est directement de niveau lexie et cette technique est relativement efficace. La technique de **filtrage de synonymes (FILT_SYN)** regroupe dans une même axie des lexies de la même langue qui ont le même sens d'après un dictionnaire de synonymes. Le manque de dictionnaires de synonymes des langues considérées est un problème pour cette technique.

Une seule technique n'est pas suffisante pour initialiser une base lexicale interlingue de la qualité qu'on souhaite. En revanche, ces techniques sont complémentaires : pour atteindre la meilleure qualité possible, nous proposons de *composer plusieurs techniques ensemble*. Nous présentons ci-après deux exemples de composition de techniques pour illustrer et motiver la nécessité de composer des techniques. Les ressources disponibles sont : (1) des dictionnaires bilingues : FA1 (français-anglais), FA2 (français-anglais), AT (anglais-thaï), AJ (anglais-japonais), (2) des bases lexicales monolingues français, anglais, japonais, thaï, (3) l'information sur la partie du discours pour chaque lexie, (4) un vecteur conceptuel associé à chaque lexie, (5) un dictionnaire de synonymes français SF. La liste des techniques disponibles est celle de la section

2. On peut alors définir la composition de techniques suivante, où les techniques sont appliquées les unes après les autres :

```
TR_BILING(FA1); TR_BILING(FA2); FILT_SYN(SF); CMP_VECTOR; TR_BILING(AJ);
TR_MBILING(AJ); TR_BILING(AT); TR_MBILING(AT)
```

L'exécution de chaque technique modifie la base en fonction des paramètres passés, p.ex., le nom d'un fichier dictionnaire bilingue pour TR_BILING. L'ordre d'exécution peut avoir un impact important sur le temps de calcul. Par exemple, si une base lexicale monolingue français a beaucoup de mots synonymes, il faudra probablement utiliser la technique de filtrage par dictionnaire de synonymes pour diminuer le nombre d'axies avant d'utiliser d'autres filtrages, pour diminuer le temps de calcul de la suite d'exécution. Si les dictionnaires FA2 et AJ, et le dictionnaire monolingue japonais ne sont pas disponibles, l'ensemble des techniques utilisables est plus restreint. On peut par exemple définir la composition de techniques suivante :

```
TR_BILING(FA1); CMP_VECTOR; TR_BILING(AT); TR_MBILING(AT)
```

A cause du manque de ressources disponibles, cette deuxième composition produit probablement une base lexicale interlingue de qualité inférieure, avec une plus grande proportion d'axies incorrectes, nécessitant plus de corrections ultérieures par les linguistes.

Ces exemples montrent qu'il est nécessaire de pouvoir utiliser des compositions des techniques différentes, car le choix d'une composition est un compromis entre la qualité souhaitée de la base produite, le coût de la mise en oeuvre des techniques (ressources techniques nécessaires), et les ressources lexicales nécessaires. Une propriété essentielle d'un système logiciel de construction d'axies est donc de pouvoir supporter la mise en oeuvre de *compositions arbitraires* de techniques de construction d'axies. Le système Jeminie, présenté dans la section suivante, a été conçu dans cet objectif.

3 Système Jeminie

Jeminie est un canevas logiciel (framework) que nous avons conçu pour automatiser la construction de bases lexicales interlingues par acceptions. Jeminie met en oeuvre les deux étapes définies dans la section 2 : normalisation des données avec production d'un premier ensemble de lexies, puis production d'axies et amélioration de l'ensemble des lexies.

Dans l'étape de normalisation des données, on ne manipule que des lexies, en convertissant vers le modèle de données (modèle d'objets) de Jeminie les données qui proviennent de ressources monolingues différentes à partir de fichiers au format XML Papillon, mais le système peut être étendu pour importer des dictionnaires dans d'autres modèles de données. On enrichit ces lexies initiales de chaque langue par des informations tirées des ressources monolingues, par exemple, des informations sur la partie du discours, une définition, etc. C'est dans cette phase qu'on contrôle la différence entre les informations qui proviennent de ressources différentes. Par exemple, on transforme l'information de partie du discours de chaque lexie en une des valeurs communes, par exemple, on convertit l'information "*nom masculin*" en "*nom*".

Dans l'étape de production d'axies et d'évolution des lexies, dans Jeminie, nous considérons une technique de création ou de filtrage des axies comme un module logiciel. Jeminie permet de composer arbitrairement des modules pour créer une base lexicale interlingue de la meilleure qualité possible. Nous appelons *processus de production de base multilingue* une séquence d'exécutions de modules. Un seul processus n'est pas applicable dans tous les cas. Il

peut être différent en fonction des ressources et outils disponibles, des propriétés des langues, des objectifs linguistiques, etc. Le système a deux types d'acteurs différents, dont le point de vue, les préoccupations et les tâches sont indépendantes : (1) un informaticien qui programme des modules, (2) un linguiste qui compose des techniques par la description d'un processus.

Jeminie a une architecture en couches. L'*interpréteur de processus* interprète les processus spécifiés par les linguistes, et déclenche l'exécution des *modules*. L'interpréteur et les modules sont développés en utilisant le *noyau* de Jeminie, qui est une bibliothèque de programmation Java de base qui implante aussi la normalisation et l'importation de dictionnaires. La syntaxe que nous avons définie pour spécifier des processus est illustrée dans les exemples de "composition" de la section 2. Cette syntaxe est simple, car destinée aux linguistes non informaticiens. Jeminie peut être étendu en développant de nouveaux modules. Les données sont représentées ici par des objets Java, p.ex. un objet de type Axie est lié à un ensemble d'objets Lexie, accessibles via des objets de type InterlingualDatabase représentant des bases multilingues. Les objets sont sauvegardés dans une base de données relationnelle via un intergiciel de persistance d'objets Java, Hibernate[2], sur lequel s'appuie le noyau.

Considérons par exemple deux dictionnaires monolingues français et anglais. Pour chaque dictionnaire, une étape de normalisation crée un objet LexieDatabase, qui contient un objet Lexie pour chaque acception avec pour chacun un objet ConceptualVector, etc. Ensuite, un objet InterlingualDatabase "vide" est créé. Lors de l'exécution d'un processus, l'objet InterlingualDatabase et les objets LexieDatabase sont passés en paramètre à chaque module exécuté. Un module peut avoir des paramètres supplémentaires spécifiques, comme un nom de fichier de dictionnaire bilingue pour le module TR_BILING. Un module exécuté crée ou supprime des objets Axie dans l'objet InterlingualDatabase. Par exemple, considérons un processus qui exécute successivement les modules TR_BILING et FILT_SYN. Le module TR_BILING crée un objet Axie pour chaque paire d'objets Lexie pour les acceptions des mots "affection" en français et "affection" et "disease" en anglais, mots qui sont traductions l'un de l'autre selon le dictionnaire bilingue passé en paramètre, et ainsi de suite pour toutes les Lexies françaises. Ensuite, le module FILT_SYN fusionne en un même nouvel objet Axie tous les objets Axie qui sont liés aux objets Lexie qui sont les acceptions des mots français "affection" et "maladie", car ils sont synonymes selon le dictionnaire de synonymes passé en paramètre, et ainsi de suite pour toutes les Axies. Les objets sont automatiquement persistants et conservés dans la base de données. Cette approche à objets permet de développer facilement des modules.

Nous avons implanté des fonctions de normalisation des données capables d'extraire des informations liées aux lexies telles que les lemmes, les parties du discours, les définitions. Nous avons validé la partie de normalisation des données avec une base lexicale monolingue pour le français qui regroupe des informations de plusieurs dictionnaires monolingues. Nous avons aussi normalisé des lexies en anglais extraites de Wordnet[3]. Le résultat obtenu est une base de lexies avec pour chaque lexie son lemme, sa partie du discours, sa définition, et son vecteur conceptuel stockée dans une base de données PostgreSQL. Nous avons pour l'instant normalisé 21400 mots / 45500 lexies du français à partir de la resource Semantic CV Service, et 52900 mots / 94660 lexies de l'anglais à partir de WordNet. Trois modules / techniques de construction d'axies sont en cours d'implantation : traduction bilingue, comparaison de partie du discours et comparaison de deux vecteurs conceptuels.

[2]http ://hibernate.sf.net/

[3]http ://www.cogsci.princeton.edu/~wn/index.shtml

4 Conclusion et perspectives

Cet article présente une stratégie de construction semi-automatique d'une base lexicale interlingue à partir de ressources existantes et introduit un système logiciel adaptable qui supporte cette stratégie. L'idée principale est de permettre à des utilisateurs linguistes de choisir et composer des techniques disponibles pour construire une base de lexies reliées par des axies. Le système est ouvert pour implanter des nouvelles techniques de construction ou de filtrage des axies. Le système devra permettre d'évaluer automatiquement un processus en cours d'exécution en mesurant sa consommation de ressources (temps de calcul, etc.) et la qualité des axies obtenues. Nous sommes en train d'identifier des critères possibles pour évaluer la qualité des axies. Ces critères seront utiles pour comparer des processus de construction pour pouvoir produire des axies de la meilleure qualité possible.

Références

BLANC E. (1999). PARAX-UNL : A large scale hypertextual multilingual lexical database. In *Proceedings of 5th Natural Language Processing Pacific Rim Symposium*, pp. 507–510, Beijing : Tsinghua University Press.

FARWELL D., GUTHRIE L. & WILKS Y. (1992). The automatic creation of lexical entries for a multilingual MT system. In *Proceedings of COLING'92*, pp. 532–538, Nantes.

GOODMAN K. & NIRENBURG S. (1991). *The KBMT Project : A Case Study in Knowledge-Based Machine Translation*. San Mateo, California : Morgan Kaufmann.

LAFOURCADE M. (2002). Automatically populating acception lexical databases through bilingual dictionaries and conceptual vectors. In *Proceedings of Séminaire Papillon 2002*, Tokyo.

MANGEOT-LEREBOURS M., SÉRASSET G. & LAFOURCADE M. (2003). Construction collaborative d'une base lexicale multilingue - le projet Papillon. In *Proceedings of TAL'2003*, number 2, pp. 151–176.

MEL'CUK I., CLAS A. & POLGUÈRE A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Louvain-la-Neuve : Duculot.

SÉRASSET G. (1994). Interlingual lexical organisation for multilingual lexical databases in NADIA. In *Proceedings of COLING'94*, pp. 278–282, Kyoto.

TANAKA K. & UMEMURA K. (1994). Construction of a bilingual dictionary intermediated by a third language. In *Proceedings of COLING'94*, pp. 297–303, Kyoto, Japan.