Notes de lecture

Rubrique préparée par Denis Maurel

Université François Rabelais Tours, LI (Laboratoire d'informatique)

Frank MORAWIETZ, Two-Step Approaches to Natural Language Formalism, *Mouton de Gruyter*, 2003, 246 pages, ISBN 978-3-11-017821-0.

Lu par **Christian RETORÉ**

LaBRI (CNRS et Université de Bordeaux) & INRIA Bordeaux Sud-Ouest

Ce livre présente clairement ce qui est aujourd'hui connu des langages de chaînes et d'arbres nécessaires à la description de la syntaxe du langage naturel, avec une contribution essentielle : l'approche à deux niveaux. Il approfondit ainsi le lien entre les deux points de vue sur la syntaxe : model theoretic syntax *et* generative-enumerative syntax *selon la dénomination proposée par Geoffrey Pullum et Barabara Scholz. Le premier point de vue, à l'intitulé difficile à traduire en français, consiste à voir l'ensemble des chaînes ou des arbres d'un langage formel comme l'ensemble des modèles finis d'une théorie logique : certaines grammaires issues des* Head-Driven Phrase Structure Grammar*, comme les grammaires de propriétés de Philippe Blache, peuvent se voir comme un ensemble de contraintes. Le second point de vue, sans doute plus habituel en tout cas pour décrire les langages de chaînes, consiste à utiliser des grammaires de réécriture. Heureusement, cette différence de point de vue n'a pas de rapport avec la description de la langue contrairement à ce qu'on entend parfois. En effet, il arrive souvent qu'un même ensemble de chaînes ou d'arbres puisse se voir d'une part comme l'ensemble des structures satisfaisant un ensemble de contraintes et d'autre part comme l'ensemble des arbres produits par une grammaire. Un exemple classique est la classe des langages réguliers (rationnels) d'arbres : un tel langage (dont les feuilles définissent un langage hors contexte de chaînes) peut être décrit par une grammaire régulière ou par une formule de la logique monadique du second ordre dont ils sont les modèles. Néanmoins, il est généralement admis que les langages hors contexte de chaînes (et donc les langages réguliers d'arbres) ne suffisent pas à décrire la syntaxe du langage naturel, et il n'est pas aisé d'étendre ces résultats classiques à des formalismes syntaxiques pertinents (légèrement contextuels) : c'est l'objet de ce livre.*

L'ouvrage est découpé en trois parties. La première se compose de deux chapitres introductifs ; l'un présente les motivations linguistiques et l'autre définit proprement les objets mathématiques sur lesquels porte l'ouvrage : alphabets, chaînes, arbres, algèbres de termes typés à la William Lawvere. La deuxième partie présente les méthodes utilisées, les résultats classiques sur les descriptions dans la logique monadique du second ordre et applique cela à des fragments de *Government and Binding (GB)* de Noam Chomsky, comme l'avait fait James Rogers en

développant une description effective dans la programmation logique par contraintes. La troisième partie présente les deux étapes, appelées *lifting* et *reconstruction* et les met en œuvre pour les bien connues grammaires d'arbres adjoints (TAG) d'Aravind Joshi, ainsi que pour les grammaires minimalistes (MG) d'Edward Stabler.

Logique du second ordre monadique (MSOL) pour la description de la syntaxe du langage naturel

Cette partie reprend et développe tout ce qui est connu de la logique monadique du second ordre pour la description de la syntaxe du langage naturel. Le premier chapitre définit la logique monadique du second ordre, en particulier sur les arbres finis avec la logique $L^2_{K,P}$ (intertraductible avec SnS) introduite par James Rogers pour étudier des arbres des approches de type GB. Les relations de base sont la dominance et la précédence. Dans ce langage, il définit l'effet des règles, les projections maximales et quelques principes de GB. Suit un chapitre sur les machines à états finis, automates et transducteurs, tout d'abord pour les chaînes, puis pour les arbres : les automates d'arbres et les *walking tree automata*, les *top-down tree transducers* et les *macro-tree transducers*. Le chapitre suivant est consacré à la décidabilité de MSOL et à la définissabilité dans MSOL. On y établit assez joliment la correspondance célèbre entre automates d'arbres et langages d'arbres définissable dans MSOL et on présente les transductions définissables en MSOL. Le dernier chapitre de cette partie est consacré aux aspects pratiques et notamment à la mise en œuvre de GB. A priori, la co-indexation à l'œuvre dans les mouvements de GB nécessite une infinité d'indices et empêche une formulation complète en MSOL. Néanmoins, avec un nombre borné d'indices, on obtient une description MSOL d'une approximation de la grammaire qu'on peut convertir en un automate d'arbres : cette conversion et l'automate pouvant être d'une complexité démesurée, l'auteur utilise MONA et des *Binary Decision Diagrams* pour rester efficace. Des mesures convaincantes sont données, ainsi que d'autres applications de ces mêmes méthodes à la vérification de logiciels ou à l'interrogation de bases de données.

Extension de l'utilisation de la logique monadique du second ordre aux formalismes syntaxiques contextuels

La troisième partie développe un point de vue résolument original développé à Tübingen par l'auteur avec Uwe Mönnich, Jens Michaelis et quelques autres : si on ne peut décrire les arbres d'un langage linguistiquement pertinent par une formule MSOL, peut-être sont-ils l'image par une transduction définissable en MSOL d'un langage régulier d'arbres (évidemment définissable en MSOL) ? C'est effectivement le cas pour les grammaires d'arbres adjoints (TAG) – dont Mönnich a montré qu'ils correspondent aux grammaires hors contexte d'arbres (CFTG) qui sont linéaires et monadiques – et pour les grammaires minimalistes d'Edward Stabler, en utilisant leur équivalence (sur les chaînes) avec les *Multiple Context Free Grammars* (MCFG), c'est-à-dire les *Range Concatenation Grammars* (RCG) de Pierre Boullier qui sont positives et simples.

La première étape consiste à construire l'arbre de dérivation, ce qui se fait en logique monadique du second ordre, et à calculer son *lift*, ce qui consiste *grosso modo* à le typer et à décomposer la suite des opérations à appliquer pour arriver aux arbres dérivés. Ensuite, l'auteur montre comment divers types de transductions, *walking tree automata*, *macro tree transducer* ou transduction MSOL, permettent de retrouver les arbres dérivés habituels. On utilise la décomposition des opérations pour calculer les relations de dominance et de précédence entre les feuilles du *lift* qui correspondent à des feuilles de l'arbre dérivé, c'est-à-dire à des mots. Calculer dominance et précédence suffit bien sûr à définir complètement l'arbre dérivé. Concernant les formalismes linguistiques, l'auteur effectue cette transformation en deux étapes pour les TAG (vus comme des CFTG monadiques linéaires) et pour les grammaires minimalistes (vues comme des MCFGS). Il montre ainsi que MSOL, ou des automates et transductions, peuvent rendre compte de formalismes syntaxiques utilisés pour la syntaxe du langage naturel, comme les TAG ou les MG, malgré leur capacité générative non contextuelle – rappelons au passage que ces formalismes admettent néanmoins des algorithmes d'analyse polynomiaux.

Le livre conclut par les questions mathématiques découvertes en chemin et par les nombreuses perspectives que ce travail ouvre au développement de grammaires, d'analyseurs syntaxiques et de description linguistique.

Notre avis

Alors qu'on peut lire ou entendre nombre de propos obscurs sur *model theoretic syntax*, ce livre démontre la force et la netteté de ces techniques, surtout pour les langages formels qui admettent les deux types de descriptions (TAG, MG). Il souligne aussi l'importance des arbres plus que des chaînes dans la structure syntaxique des énoncés. Il prouve ainsi qu'en utilisant deux niveaux de description, on peut sortir des langages hors contexte tout en gardant des descriptions logiques. Et, en lisant cet ouvrage, on prend conscience que bien peu est connu sur les langages d'arbres nécessaires à décrire la syntaxe des langues humaines.

On peut regretter deux manques, pourtant inévitables, concernant les grammaires minimalistes. Le premier est l'absence, dans l'ouvrage, du détail de la traduction des MG en MCFG, mais il s'agit d'un résultat important d'un autre chercheur, Jens Michaelis. On se demande aussi si cette construction peut s'effectuer avec des tuples d'arbres plutôt que de chaînes, afin de mieux connaître la structure des arbres d'analyse. La réponse est oui, mais ce résultat n'a été établi qu'en 2007 par Uwe Mönnich ainsi que par Greg Kobele, Sylvain Salvati et nous-mêmes.

Ce livre est clairement rédigé, avec juste ce qu'il faut de détails, que ce soit pour comprendre la grande avancée de l'auteur, l'approche à deux étapes, ou pour profiter de sa présentation des grammaires d'arbres et de leur description en logique monadique du second ordre, notions dont il justifie pleinement la pertinence linguistique.

Peter JACKSON, Isabelle MOULINIER, Natural Language Processing for Online Applications. Text retrieval, extraction and categorization, *John Benjamins Publishing Company*, 2007, 231 pages, ISBN 978-90-272-4992-0.

Lu par **Emmanuel CARTIER**

LDI UMR 7186

L'ouvrage de Peter Jackson et Isabelle Moulinier a pour objectif de présenter les technologies et les techniques de Traitement Automatique des Langues (TAL) utilisées et utilisables dans les applications commerciales. Contrairement à ce que laisse entendre le titre, l'ouvrage n'est pas limité aux applications Web, mais il parcourt un large éventail de problématiques où des traitements linguistiques sont essentiels.

Un chapitre introductif présente les concepts et techniques fondamentaux du traitement automatique des textes. Les chapitres suivants explicitent quelques grands domaines d'applications : moteurs de recherche (chapitre 2), recherche d'information (chapitre 3), catégorisation de documents (chapitre 4) et extraction d'information (chapitre 5). Chacun de ces chapitres est organisé en trois sections : une première section explicite la problématique spécifique, puis les solutions techniques existantes et/ou utilisables sont décrites dans le détail ; une dernière section évoque les problèmes non résolus par les technologies actuelles et les perspectives. Enfin, deux niveaux de lecture sont proposés, des encarts et notes détaillant certains points ressentis comme importants par les auteurs.

L'ensemble de l'ouvrage est d'une lecture agréable, l'anglais reste accessible, évitant tout jargon. La construction de l'ouvrage est particulièrement didactique, permettant au novice de se familiariser avec les différents domaines d'applications et au professionnel ou au chercheur de maîtriser et d'approfondir les technologies et les techniques mises en œuvre. L'ouvrage est une réédition, enrichie des avancées technologiques et techniques survenues depuis la première édition, et a été rendu moins académique dans sa présentation pour permettre à un public professionnel de l'utiliser. Les références citées ont également été mises à jour.

Le premier chapitre est une introduction au Traitement Automatique des Langues, à sa terminologie et à ses concepts de base. Les auteurs y décrivent tout d'abord les objectifs majeurs du livre : décrire les techniques et technologies matures ou proches d'arriver à maturité, dans les domaines où le traitement automatique des langues s'avère être le verrou technologique majeur.

Ils décrivent ensuite le problème majeur du TAL, à savoir l'ambiguïté. Ils insistent enfin sur l'importance capitale de l'évaluation des systèmes et la nécessité de protocoles objectifs pour les réaliser. Certains concepts de base de la linguistique sont alors présentés : syntaxe, sémantique, pragmatique et contexte, dans la tradition du pragmatisme américain. Sont évoquées alors les deux approches majeures en

TAL : l'approche statistique et l'approche linguistique. Vient ensuite une présentation des grands domaines d'applications TAL. Reprenant une architecture explicitée par P. Jackson dans l'*Encyclopedia of Language and Linguistics*, présentant les applications TAL selon deux axes majeurs *reproduction – transformation*, *reconnaissance – génération*. L'ensemble permet de placer les différentes applications du TAL dans une matrice. Enfin, les auteurs présentent les outils de base de la linguistique informatique : segmenteurs, raciniseurs, analyseurs morphosyntaxiques, analyseurs syntaxiques. L'ensemble est correctement décrit.

Le chapitre 2 s'intéresse au domaine des moteurs de recherche. Les auteurs commencent par présenter le fonctionnement général de ces systèmes : processus d'indexation et traitement des requêtes. Une présentation très détaillée des technologies statistiques et probabilistes utilisées pour ces opérations est proposée, même si on regrette la surabondance d'équations non commentées pour le novice. Ensuite, les auteurs consacrent toute une section aux différentes métriques utilisées pour l'évaluation des systèmes. Puis, ils détaillent enfin différentes pistes technologiques utilisées par les professionnels pour améliorer leurs systèmes : expansion des requêtes par le biais de thesaurus et de réseaux sémantiques, modèles d'espaces vectoriels ou de calculs probabilistes, utilisation des classements liés aux nombres de liens vers les pages, ou encore utilisation du retour des utilisateurs. Le chapitre est fort bien détaillé, on regrettera juste que les parties liées à des modèles statistiques soient un peu trop techniques.

Le chapitre 3 concerne l'extraction d'information, dont l'objectif est non plus de retourner des documents liés à une requête de l'utilisateur, mais des informations contenues dans les documents liées à des types d'informations prédéfinies. Les auteurs s'appuient sur la série des sept conférences MUC qui ont eu lieu dans les années 90 à 2000. Ces conférences, subventionnées par les services de renseignements américains, mettaient en place des tâches d'extraction liées à des objets (entités nommées principalement), mais également des relations et des événements liés à ces entités, *via* le remplissage de schémas d'informations (*templates*). Par ailleurs, ces conférences, basées sur des corpus de travail, puis de tests, et sur des techniques d'évaluation, ont permis de comparer les technologies et de faire émerger les techniques les plus appropriées.

La présentation des technologies est, comme partout ailleurs dans le livre, très claire et progressive : expressions régulières, automates à états finis et langages réguliers (illustrés par le système GATE), grammaires hors contexte, algorithme CYK (Cocke, Young, Kasami). Les difficultés des meilleurs systèmes sont explicitées et les perspectives d'avenir, notamment l'apprentissage automatique des ressources linguistiques et des modèles statistiques, sont évoquées. Ce chapitre est sans doute avec le précédent l'un des plus aboutis et des plus informatifs sur l'état de l'art, même si on regrettera que l'apprentissage automatique ne soit pas plus développé.

Le chapitre 4 concerne la « catégorisation de documents », c'est-à-dire le groupement des documents dans certaines classes prédéfinies ou découverts par les algorithmes. Ce domaine a de multiples applications : routage des documents, ajout de méta-informations aux documents, indexation. Là encore, les auteurs présentent les deux grandes familles de technologies utilisées : approches statistiques permettant de découvrir des espaces multidimensionnels congruents, en liaison avec la fréquence d'ensembles de mots-clés ; approches « linguistiques » basées sur des mots-clés et des séquences textuelles particulières. Les auteurs insistent dans ce chapitre sur les techniques permettant d'apprendre, automatiquement ou semi-automatiquement, les ressources linguistiques. Par ailleurs, des métriques d'évaluation des systèmes, complémentaires des mesures de précision et de rappel, sont présentées, principalement à partir des travaux effectués dans le cadre des conférences TREC.

Le chapitre 5 porte sur la fouille de données non structurées (*Text mining*). Il s'agit ici, pour les auteurs, de faire en quelque sorte la fusion des différentes applications présentées auparavant. Une présentation conceptuelle du domaine de la fouille de données textuelles est tout d'abord faite, permettant de pointer sur la spécificité de ce domaine : contrairement à l'extraction d'information qui se contente de trouver et de présenter des fragments des textes originaux, la fouille de données en effectue une analyse dans deux directions principales : en utilisant d'autres informations extraites dans d'autres documents, et en faisant l'analyse de ces informations avec une modélisation de l'« état du monde » extérieur. Il en découle que la fouille de données doit traiter les problèmes linguistiques de la référence et de la coréférence. Les auteurs reprennent alors les travaux sur la reconnaissance des entités nommées pour insister sur les tâches de fusion et de gestion des simples extractions (regroupement d'entités coréférentes notamment). Les auteurs présentent ensuite les principaux travaux menés sur la résolution de la coréférence dans les textes, à partir des résultats de la 7e conférence MUC. Là encore deux groupes de techniques : heuristique et inductive, basées sur la généralisation de patrons linguistiques à partir de corpus existants ; statistiques, basées sur des calculs probabilistes. Enfin, les auteurs détaillent le domaine du résumé automatique, se fondant principalement sur la SUMMAC *Summarization Conference*, ainsi que la détection de thèmes. Ce dernier chapitre, plus prospectif que les autres, laisse un goût d'inachevé tant il brasse différents types d'applications sans véritablement en faire une présentation exhaustive.

Au final, l'ouvrage est une bonne introduction au TAL du point de vue de ses applications et des technologies majeures actuellement utilisées. Certains chapitres, notamment le 2 et le 3 (auquel il faudrait rattacher dans le 5 la partie sur les entités nommées), peuvent servir d'introduction pour les novices ou ceux désireux de disposer d'un état des lieux en 2007. Cependant, les personnes sans bagage mathématique seront vite perdues dans les équations statistiques proposées, mais trop incomplètement commentées. De même, on regrettera que les auteurs se soient limités aux approches utilisées outre-Atlantique, et n'aient pas développé plus avant

d'autres techniques pourtant appliquées en Europe, notamment la construction d'ontologies ou encore l'apprentissage automatique. Au final, ce document comble le manque d'un état des lieux sur l'analyse automatique des textes et mériterait son pendant pour les applications basées sur les techniques liées à la génération automatique de documents, qui est un autre champ d'applications commerciales particulièrement développé.

Bruno BACHIMONT, Ingénierie des connaissances et des contenus : le numérique entre ontologies et documents, *Hermes-Lavoisier*, 2007, 278 pages, ISBN 978-2-7462-1369-2.

Lu par **Benoît HABERT & Marie GUÉGAN**

ICAR & ENS LSH, Lyon ; LIMSI

Bruno Bachimont part du postulat p. 10 que « la connaissance n'est pas un objet, mais qu'elle ne s'appréhende qu'à travers des objets [les inscriptions] dont elle est l'interprétation ». Pour lui, la nature double des inscriptions – contenus culturels à interpréter et objets techniques à élaborer – conduit à distinguer au sein de l'ingénierie des connaissances entre une ingénierie des représentations, *visant à formaliser le sens des inscriptions, et une* ingénierie des contenus, *visant à formaliser la forme d'expression des inscriptions. En écho à cette distinction, qui suppose d'ailleurs intrinsèquement liées les deux approches (p. 44), après une première partie* Ingénierie du numérique, *deux parties :* Connaissances et ontologies *d'une part, et* Contenus et documents *d'autre part, sont abordées.*

Le livre de Bruno Bachimont ne constitue pas un panorama « neutre » de l'ingénierie des connaissances dans son rapport aux ontologies et aux documents. C'est un point de vue impliqué, déterminé, d'où son caractère stimulant comme certaines de ses limites. On s'attachera aux positionnements majeurs, sans suivre l'ordre de l'ouvrage, pour revenir, *in fine*, sur les insatisfactions ressenties.

Bruno Bachimont s'écarte des conceptions « classiques » de l'ontologie comme reflet du réel ou de la pensée (p. 81). Pour lui, « [...] la modélisation formelle du sens des inscriptions est comprise comme une technique permettant de rendre des informations manipulables par l'inférence logique dans les systèmes informatiques, sans conférer un quelconque statut cognitif aux représentations formelles élaborées pour cela » (p. 44). Bruno Bachimont insiste (p. 91) également sur la dépendance des connaissances et des modes de raisonnement aux domaines dont ils relèvent. Les ontologies sont alors forcément « régionales », c'est-à-dire relatives à un domaine et à une tâche (p. 146). « [...] Il ne peut y avoir une ontologie matérielle de toute la réalité, car celle-ci comporte des régions distinctes et incommensurables » (p. 114). Élaborer une ontologie régionale, c'est alors organiser et rationaliser les connaissances pour un domaine et une tâche afin de les rendre manipulables. Bruno

Bachimont propose p. 132 de « partir de l'expression linguistique des connaissances pour proposer une modélisation linguistique, l'ontologie différentielle, que l'on formalise en un modèle formel, l'ontologie référentielle, qui s'opérationnalisera en une ontologie computationnelle ». Les formulations linguistiques récurrentes sont l'indice de connaissances à modéliser et formaliser. La méthodologie, ARCHONTE, dérivée de la sémantique interprétative de F. Rastier (p. 142-144), produit un arbre de concepts disjoints (sans héritage multiple, donc). La volonté d'opérationnaliser pousse ensuite à normaliser, à stabiliser les signifiés, à leur donner des contextes de référence, là où la sémantique interprétative insiste sur le remodelage en fonction des contextes des ensembles de sèmes associés à un signifiant et sur la non-compositionnalité du sens. L'approche linguistique s'avère en outre inadaptée aux concepts possédant un contenu perceptif (p. 159). Le passage à une ontologie référentielle revient par ailleurs à changer profondément l'interprétation des concepts. Ils relèvent dans une ontologie différentielle d'une explicitation linguistique et de l'association à d'autres concepts. Dans une ontologie référentielle, chacun d'eux renvoie à une dénotation dans un modèle, c'est-à-dire à un ensemble d'individus. Enjeu de ce passage : « [...] les ontologies doivent être des représentations permettant de naviguer entre des extraits documentaires, des documents de référence, des exemples, des échantillons, etc., qui permettent de mener le raisonnement idoine ou de trouver l'information pertinente. Ainsi, l'inférence ontologique ne se substitue pas au raisonnement, mais facilite un rapprochement que l'utilisateur peut toujours récuser le cas échéant » (p. 160). En somme, chemin faisant, c'est un déplacement de l'idée de Web sémantique qu'opère Bruno Bachimont. Il ne s'agit pas d'obtenir une « ontologisation globale » qui permette aux applications des inférences « machinales » dont les humains seraient déchargés, mais des opérationnalisations « régionales » qui facilitent pour l'utilisateur les rapprochements, les synthèses et en définitive l'interprétation (p. 160-161).

Si pour Bruno Bachimont, p. 24, « Le numérique possède une double tendance : fragmenter les contenus pour les recomposer d'une part, dé-sémantiser les contenus pour les ré-investir de sens d'autre part », il faut alors « poursuivre la mutation numérique par la constitution d'une culture en travaillant de manière raisonnée sur l'instrumentation des contenus et l'outillage des utilisateurs [...] » (p. 222). D'où par exemple un élargissement de la notion d'indexation, entendue comme « tâche plus générale d'instrumentation du contenu » (p. 194). Bruno Bachimont plaide pour une ingénierie des connaissances pour laquelle « l'enjeu n'est pas de modéliser la pensée, mais de l'instrumenter et de l'équiper en lui donnant les outils aptes à développer ses capacités » (p. 47), à l'aide de la « raison computationnelle » (p. 73), par analogie avec la raison graphique de J. Goody. Il conclut d'ailleurs p. 243 son ouvrage ainsi : « [...] l'enjeu n'est pas de raisonner en imitant l'utilisateur pour assurer à sa place la relation au monde dans lequel résoudre le problème, mais de construire la représentation permettant à l'utilisateur d'effectuer lui-même les tâches qui lui reviennent, en abordant le réel à travers la médiation des outils des représentations construites par les outils de l'ingénierie des connaissances. L'enjeu

est de passer d'une représentation formelle des connaissances à une médiation formelle des expressions non formelles de connaissances ». En ligne d'horizon, un humanisme numérique et l'émergence d'« avisés » du numérique (p. 226-228), capables de s'orienter « dans le dédale de la connaissance numérisée en réseau ».

On suit volontiers Bruno Bachimont quand il affirme, p. 12 : « l'ingénierie est une dimension originale du savoir humain qui excède la posture scientifique » et dans sa volonté de contribuer à une culture endogène des innovations techniques (p. 70). On lui sait gré d'essayer de lier les positionnements globaux évoqués ci-dessus sur l'ingénierie des connaissances et les débats techniques actuels (par exemple sur les enjeux respectifs de MPEG-4 et de MPEG-7), de multiplier les excursus suggestifs (par exemple sur l'indexation par similarité p. 201). On apprécie les larges panoramas historiques et philosophiques (sur Hilbert, Husserl et Aristote). On peut partager son agnosticisme quant au réalisme métaphysique ou cognitif des ontologies. Cette distanciation appréciable fait d'autant plus regretter que les tâches et le public visés restent flous, que le Web sémantique « historique » ne soit pas analysé en détail, dans ses projets comme dans ses réalisations et enfin que les positions choisies ne soient pas plus précisément resituées dans les débats actuels sur le Web 2.0. Les apports, réels, de l'ouvrage en seraient plus immédiatement et clairement perceptibles.

Trude HEIFT, Mathias SCHULZE, Errors and Intelligence in Computer-Assisted Language Learning : Parsers and Pedagogues, *Taylor et Francis*, 2007, 283 pages, ISBN 978-0-415-36191-0.

Lu par **Claude PONTON**

Laboratoire Lidilem, université Stendhal

L'ouvrage de Heift[1] et Schulze[2] propose, au travers de l'analyse d'erreurs, de la production de rétroaction et de la modélisation des apprenants, un large panorama des recherches actuelles concernant l'utilisation du TAL en l'ALAO (Apprentissage des Langues Assisté par Ordinateur). Fondé sur une large bibliographie, malheureusement quasiment anglo-saxonne et allemande, il constitue un ouvrage de base pour toute personne abordant ce champ de recherche éminemment pluridisciplinaire que constitue l'association TAL et ALAO.

1. Trude Heift est professeure associée de linguistique à l'université Simon Fraser de Burnaby (Canada).

2. Mathias Schulze est professeur associé d'Allemand à l'université de Waterloo (Canada).

TAL en ALAO

Le premier chapitre de l'ouvrage est entièrement consacré aux liens entre les deux domaines du TAL et de l'ALAO. L'ALAO est un domaine éminemment pluridisciplinaire qui a donc évolué à la fois en fonction des théories de l'apprentissage, de celles de la psychologie (développementale et cognitive), de la technologie et du TAL. Heift et Schulze situent clairement l'apport du TAL en terme d'analyse sophistiquée des erreurs pour permettre, entre autres, un travail réflexif des apprenants sur leurs erreurs, une modélisation et un classement de ces apprenants, mais également dans des perspectives de recherche. On regrette un peu que les autres apports ne soient pas plus cités ; on pense ici à la génération automatique d'activités, à la recherche de documents pédagogiques, à l'exploration de corpus, etc.

Centré donc sur l'analyse d'erreurs d'apprenant, l'ouvrage s'intéresse aux spécificités des grammaires et des algorithmes mis en œuvre dans les systèmes d'ALAO. Si les techniques de relâchement de contraintes semblent actuellement les plus prometteuses et les plus utilisées, les auteurs soulignent toutefois les nombreuses difficultés restantes. En effet, quand l'analyse classique a essentiellement un objectif de robustesse, l'analyse d'erreur se doit de considérer à la fois les entrées bien formées et celles mal formées dans un but d'interprétation et de diagnostic.

Malgré ces difficultés, de nombreux projets d'ALAO basés sur un analyseur ont vu le jour ; l'ouvrage en recense 119 d'une grande diversité. À travers l'étude de ces projets Heift et Schulze concluent ce chapitre sur le fait que, parmi l'ensemble des technologies TAL (analyse, génération, traitement de la parole), l'analyse est certainement la plus employée en ALAO, notamment sur les niveaux lexical et morphosyntaxique, tout en restreignant le plus souvent le contexte pour éviter la prise en compte des niveaux sémantique et pragmatique. Cet état de fait, dû aux faiblesses du TAL lui-même, n'est pas un problème incontournable à partir du moment où, lors du développement de ces systèmes, le domaine même de l'apprentissage des langues est réellement pris en compte. C'est l'une des voies défendues ici pour la réalisation de systèmes mieux adaptés.

Analyse d'erreurs et description

Après un rapide bilan des correcteurs orthographiques et grammaticaux commerciaux, l'ouvrage aborde le processus d'analyse et de description des erreurs qui constitue l'un des éléments centraux de tout système d'ALAO basé sur du TAL.

Dans une première partie, il retrace l'évolution du domaine de l'analyse d'erreurs, depuis les travaux de Lado en 1957 sur l'analyse contrastive, jusqu'aux travaux actuels sur l'interlangue. Dans la deuxième partie, le focus est mis sur le problème de la classification des erreurs ; prérequis nécessaire à la production de rétroaction adaptée. À travers la large diversité des approches, les auteurs distinguent trois grandes difficultés dans la constitution de ces classifications : la

couverture (toutes les erreurs ne peuvent être anticipées), l'homogénéité des descripteurs et les caractéristiques même de la classification (choix des catégories morphosyntaxiques).

La dernière partie du chapitre présente un bref aperçu de l'un des domaines actuels de recherche qui est l'usage de corpus d'apprenants. Ces corpus, en effet, présentent à la fois un intérêt lors de la conception des outils TAL et lors de l'évaluation de ces mêmes outils. On regrette toutefois que le lien entre l'étude de corpus et la classification des erreurs ne soit pas suffisamment abordé car il constitue à notre avis une voie de recherche prometteuse pour répondre aux trois problèmes évoqués précédemment. Bien évidemment, comme le soulignent les auteurs, cette voie nécessite la constitution de larges corpus d'apprenants ce qui n'est pas le cas actuellement.

Rétroaction

Le quatrième chapitre traite de la production de rétroaction vers l'apprenant suite à l'analyse de sa production. La qualité de la rétroaction, en terme de pertinence pour l'apprentissage, est fonction de multiples facteurs. Dans ce cadre, Heift et Schulze proposent une analyse de la rétroaction aussi bien à travers le prisme de l'interaction homme/machine que celui des théories de l'apprentissage et de l'enseignement des langues. Ils montrent par la suite qu'en ALAO différents types de rétroactions sont possibles : fournir simplement la correction, pointer l'erreur, anticiper les erreurs, produire des rétroactions préenregistrées et enfin, utiliser une analyse automatique des erreurs pour générer des rétroactions plus précises. Les conclusions de cette étude tendent à montrer que, pour être pertinent, la rétroaction produite doit être précise, courte et comportant un seul message.

Cette étude est suivie d'une description relativement détaillée du système « German Tutor » dont l'une des spécificités est la génération de rétroaction en cas d'erreurs multiples au sein d'une même phrase. Ce système, basé sur une grammaire HPSG, utilise une structure contenant la liste des erreurs classées selon des priorités calculées sur des critères pédagogiques et sur le modèle utilisateur.

La dernière partie du chapitre aborde les autres outils de guidance que l'on retrouve dans les différents systèmes d'ALAO, comme l'accès à des aides contextuelles ou à des dictionnaires. L'étude de l'appropriation de ses aides par les apprenants dépend de nombreux facteurs comme le type des activités, les capacités de langage des apprenants et les modes de rétroaction. Finalement, au vu de l'importance de la rétroaction dans les dispositifs d'apprentissage, les auteurs s'étonnent de voir cet aspect peu développé. En effet, la plupart des systèmes ne dépassent pas les rétroactions du type « vrai/faux » alors que, toujours pour Heift et Schulze, il est possible de produire des rétroactions plus pertinentes même dans le cadre de systèmes non basés sur du TAL.

Modélisation des apprenants

Le cinquième chapitre s'intéresse aux modèles d'apprenant dans les systèmes d'ALAO. Selon les auteurs, ces modèles sont nécessaires pour permettre un apprentissage individualisé. Leurs utilisations sont diverses : corrective (production de rétroaction adaptée), élaborative (extension des connaissances de l'apprenant), stratégique (guider les interventions pédagogiques), diagnostique (déterminer l'état de connaissance des apprenants) et évaluative (mesurer le niveau de l'apprenant). Après une large partie consacrée à une présentation des modèles dits « utilisateur », les auteurs montrent la spécificité des modèles d'apprenant qui sont d'une complexité supérieure due notamment à l'inconsistance des connaissances de ces apprenants et donc à la difficulté de maintenir des modèles pertinents.

En guise de conclusion, les auteurs appellent à plus d'interdisciplinarité entre informatique, TAL et didactique ; ils pensent que l'apport du TAL à l'ALAO sera réellement significatif dans les trente prochaines années et ouvrira même de nouvelles voies empiriques et théoriques en apprentissage des langues.

Ludovic TANGUY, Nabil HATHOUT, Perl pour les linguistes : programmes en Perl pour exploiter les données langagières, *Hermès*, 2007, 504 pages, ISBN 2746216353.

Lu par **Thierry HAMON**

LIPN UMR 7030, Université Paris13 CNRS

Cet ouvrage aborde la problématique de l'exploitation de données langagières à l'aide de programmes écrits en Perl. Il est principalement destiné aux linguistes et à toute personne souhaitant travailler sur des données textuelles disponibles dans un format électronique. Les auteurs se sont attachés à rendre accessibles à tous, les notions informatiques nécessaires au développement de programmes pour le traitement de données textuelles. Plusieurs grilles de lecture sont ainsi possibles suivant ses compétences en programmation. L'exécution de tous les programmes Perl fournis dans le livre est détaillée et les problèmes particuliers pouvant être rencontrés lors de telles manipulations sont identifiés. Le livre est composé de dix chapitres, pouvant être groupés en deux grandes parties, et de cinq annexes. Les quatre premiers chapitres abordent les notions informatiques de base, indispensables à la compréhension du livre. Les six chapitres suivants présentent plusieurs techniques de manipulation de textes numériques. Les annexes rassemblent les aspects purement techniques du traitement de données textuelles. Le livre est complété par une table des programmes et un site Web (http://perl.liguistes.free.fr) qui met à disposition l'ensemble de ces programmes mais aussi des exemples de données utilisés dans le livre.

L'objectif du premier chapitre est de décrire les différents matériaux nécessaires à l'exploration de données textuelles. Ainsi, ce chapitre commence par une présentation des caractéristiques du format texte et des objets manipulables dans ce

format. L'utilisation d'outils de conversion permettant d'extraire le contenu textuel de documents stockés dans différents formats (PDF, Doc, HTML, etc.) est également abordée. On trouve également un panorama des ressources textuelles et lexicales utilisées ainsi que des outils de TAL nécessaires à la préparation linguistique des textes (segmentation, étiquetage morphosyntaxique et lemmatisation). Deux premières annexes complètent ce chapitre et décrivent d'une part le nettoyage, la normalisation et l'application des outils de TAL sur des données textuelles, et d'autre part la problématique du codage des caractères dans les textes.

Le chapitre 2 présente l'environnement de travail et décrit sa mise en place et son adaptation pour l'utilisation de programmes Perl. L'installation de Perl pour Windows (ActivePerl) est tout d'abord détaillée. Puis, les auteurs présentent les outils s'exécutant en ligne de commande, pour les systèmes d'exploitation Windows ou Unix. Le chapitre se termine sur la présentation de l'installation de ressources supplémentaires telles que des modules Perl, le lexique Morphalou et l'étiqueteur morphosyntaxique et lemmatiseur TreeTagger.

Le chapitre 3 présente le langage Perl et plus généralement les notions élémentaires de programmation : syntaxe de base, structures de contrôle, structures de données, manipulation de fichiers, arguments de programmes et fonctions. Les trois dernières annexes complètent cette présentation en décrivant l'adaptation du langage aux spécificités de la langue traitée, les structures de données complexes, l'utilisation de références et la programmation objet. Dans la dernière annexe, on trouve un récapitulatif de la syntaxe et des différentes notions utilisées en Perl.

Dans le chapitre 4, les auteurs présentent les expressions régulières. Ce mécanisme, disponible à la fois dans les éditeurs de texte et dans des langages de programmation comme Perl, permet de représenter formellement un ensemble de chaînes de caractères. Le chapitre détaille à la fois le formalisme lié à cet outil et son utilisation en Perl. On y trouve également une description des opérateurs de modification, d'extraction ou de substitution, qui constituent le principal mode d'utilisation des expressions régulières lors du traitement de données textuelles.

Le cinquième chapitre est consacré aux premiers traitements à réaliser pour l'exploration ou l'exploitation de données textuelles. Il s'appuie sur les notions informatiques vues précédemment et les met en œuvre pour identifier et extraire les unités linguistiques de ressources textuelles. Les programmes Perl proposés, par ordre de difficulté croissante, permettent la recherche de mots dans un texte segmenté ou étiqueté, la recherche de suites de mots prédéfinis ou correspondant à un patron syntaxique.

Le chapitre 6 s'intéresse à la statistique lexicale et notamment à l'utilisation des fréquences des mots ou des lemmes pour quantifier des faits langagiers. Différents calculs de distribution d'unités linguistiques sont d'abord présentés (fréquences des formes fléchies et des lemmes de mots). Puis les auteurs décrivent l'implémentation du calcul de cooccurrences de mots et de l'information mutuelle.

Le septième chapitre propose l'implémentation de concordanciers plus ou moins complexes. L'objectif est de construire une présentation des contextes d'un mot pivot pour observer son utilisation en corpus. Après la description d'un premier concordancier s'appuyant sur la forme exacte des mots, une extension exploitant les expressions régulières est proposée afin de couvrir un ensemble de formes associées au mot pivot.

Le chapitre 8 s'intéresse à l'exploitation des lexiques comme source principale d'une étude linguistique particulière mais aussi comme complément en vue de l'amélioration des traitements de corpus. Ainsi les auteurs proposent d'abord des programmes permettant la caractérisation de lexiques, l'extraction de sous-lexiques et la fusion de plusieurs lexiques. Le lexique comme outil et objet d'étude est abordé à travers le recensement des verbes défectifs, la création de dictionnaires de rimes et la génération des occurrences possibles d'une construction verbale donnée. Le chapitre se termine par la rédaction de programmes utilisant un lexique pour l'identification de néologismes, le calcul des formes fléchies d'un mot inconnu et la génération de dérivés à partir d'un affixe.

Les techniques de manipulation des données XML sont présentées au chapitre 9. Les programmes présentés dans ce chapitre s'appuient sur une utilisation importante de modules Perl disponibles sur CPAN (dépôt des ressources Perl). Après une description des principes fondamentaux de la norme et de la syntaxe XML, les auteurs détaillent à travers l'utilisation de modules Perl, l'analyse de données XML (après stockage en mémoire ou par flux), mais aussi la création et la modification de documents au format XML.

Le dernier chapitre est consacré à l'utilisation du Web pour la constitution de corpus pour l'étude du fonctionnement de la langue. La nature des documents Web manipulables et notamment des pages rédigées en langage HTML est détaillée. Ce chapitre aborde les traitements nécessaires à la constitution d'un corpus à partir du Web : récupération d'un page Web ou d'un site, nettoyage et extraction du contenu textuel des documents HTML. L'automatisation de l'interrogation de moteurs de recherche y est décrite en vue de l'étude statistique ou de l'extraction de contextes d'emplois attestés.

Ce livre didactique offre un tour d'horizon assez vaste et complet sur les techniques informatiques de manipulation et d'exploration de données textuelles. Les notions sont abordées progressivement et illustrées par des programmes Perl décrits en détail. Ce livre s'adresse aussi bien aux chercheurs (linguistes ou non), qu'aux étudiants en linguistique informatique ou en informatique voulant se familiariser ou se sensibiliser aux problématiques de traitement des données langagières sur support numérique, chacun pouvant adapter sa lecture suivant ses propres connaissances.