Du texte aux clauses de Horn par la combinaison de l'analyse linguistique et de l'apprentissage automatique

SYLVAIN DELISLE

Département de mathématiques
et d'informatique
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada G9A 5H7
email: Sylvain_Delisle@uqtr.uquebec.ca
phone: 819-376-5125 fax: 819-376-5012

KEN BARKER, JEAN-FRANÇOIS DELANNOY,
STAN MATWIN, STAN SZPAKOWICZ

Department of Computer Science
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
email: {kbarker, delannoy,
stan, szpak}@csi.uottawa.ca
phone: 613-564-5420 fax: 613-564-9486

RÉSUMÉ

Cet article décrit un système d'extraction de connaissances à partir de textes techniques en anglais. Notre hypothèse de départ est que dans les textes techniques la syntaxe est un indicateur fiable de la signification. L'interprétation sémantique part donc de la syntaxe. Le sous-système linguistique utilise un analyseur général de l'anglais, indépendant du domaine, et un interpréteur sémantique interactif qui accumule et utilise son expérience. Les structures sémantiques résultantes sont traduites en clauses de Horn, une représentation appropriée pour l'apprentissage à partir d'explications (EBL). Le système apprend au niveau symbolique des représentations de la théorie du domaine et des exemples, tous deux fournis par le sous-système linguistique. Cette approche a été appliquée à une partie du Guide d'Impôt Canadien.

MOTS-CLÉS: Analyse de Textes, Apprentissage automatique, Acquisition de connaissances

1. Introduction

Notre projet a pour but d'acquérir des connaissances à partir de textes. Nous construisons un système qui accumule la connaissance en utilisant un noyau initial de connaissances spécifiques au domaine le plus réduit possible, en nous appuyant sur la participation de l'utilisateur. Le système traite des textes techniques en anglais en appliquant des méthodes de traitement de la langue naturelle et d'apprentissage automatique. Le résultat est une base de règles sous forme de clauses de Horn représentant les relations sémantiques entre les concepts du texte. Nous pensons que cette tâche peut être réalisée sans connaissance préalable, pourvu que l'utilisateur entraîne le système durant la phase initiale d'extraction. Son intervention doit normalement diminuer avec le temps, tandis que le système met à profit l'expérience accumulée au cours des interactions précédentes.

Comme une grande quantité de connaissances est véhiculée par les livres de cours, les manuels techniques et les ouvrages de référence, un système comme le nôtre peut constituer une alternative précieuse aux outils “traditionnels” d'acquisition de connaissances. Dans un domaine pour lequel un texte descriptif est disponible, le traitement de ce texte pourra fournir une première version, peut-être imparfaite, d'une base de connaissances réaliste du domaine, que l'utilisateur pourra ensuite améliorer.

Les principales caractéristiques de notre approche et ses conditions d'application sont les suivantes:

- L'analyse détaillée d'un fragment d'un texte technique[1] est suivie d'une analyse semi-automatique des relations entre propositions, des relations casuelles et des relations au sein des groupes nominaux. La structure sémantique obtenue est transformée, de façon semi-automatique, en clauses de Horn. Nous faisons l'hypothèse que dans les textes techniques la syntaxe donne une bonne indication du sens: une interprétation basée sur la syntaxe de surface est en général applicable (Kieras 1985), et un haut degré de compositionalité est possible. L'analyseur syntaxique utilise une grammaire standard de l'anglais, neutre quant à la théorie linguistique, et basée sur Quirk *et al.* (1985). Les détails de l'analyse et de l'interprétation sémantique à raisonnement par cas (*Case-based*) sont donnés dans Delisle (1994).
- Le système doit être entraîné par l'utilisateur. Il se rappelle et généralise les ajouts et les modifications de ses dictionnaires de schémas sémantiques. On arrive progressivement à une saturation par la connaissance linguistique, et la participation de l'utilisateur tend à se réduire à une simple confirmation. Nous travaillons sur les détails d'une mesure de l'interaction, dans le but de quantifier l'efficacité du système. Elle prendra en compte le nombre d'interventions (par exemple le nombre de mises à jour dans le dictionnaire de schémas sémantiques) et la facilité de l'interaction (en utilisant par exemple les types d'oracle d'Angluin (1988)).
- Nous supposons une certaine richesse de la connaissance syntaxique et une description sémantique détaillée des mots fonctionnels. Le dictionnaire Collins (Karp *et al.* 1992) est utilisé par l'analyseur pour fournir l'information sur les catégories lexicales. WordNet (Miller 1990) sera utilisé pour la désambiguïsation et le regroupement sémantique (*semantic clustering*) grâce à un parcours de l'arbre terminologique que propose WordNet (Feng *et al.* 1994)). Aucune connaissance du domaine n'est nécessaire au départ: le système peut partir d'un dictionnaire de schémas sémantiques vide. Nos premiers essais confirment que l'analyse casuelle d'un texte avec, au départ, des dictionnaires vides peut fournir un pourcentage très acceptable d'hypothèses que l'utilisateur n'aura qu'à confirmer (Delisle 1994).
- Le système acquiert de la connaissance incrémentalement au fur et à mesure qu'il progresse dans l'analyse du texte.

---

[1] Il n'en existe pas de définition consensuelle. Nous avons constitué une liste de propriétés linguistiques des textes techniques, qui sera publié ultérieurement. On considère généralement que les textes techniques sont sensiblement plus faciles à traiter que les autres, même s'il

2. Organisation du système

L'organisation du système est résumée dans la Figure 1; les encadrés ovales représentent les modules, les rectangles en trait fin, les données transmises entre les modules, et les rectangles en trait épais les dépôts d'information permanents.

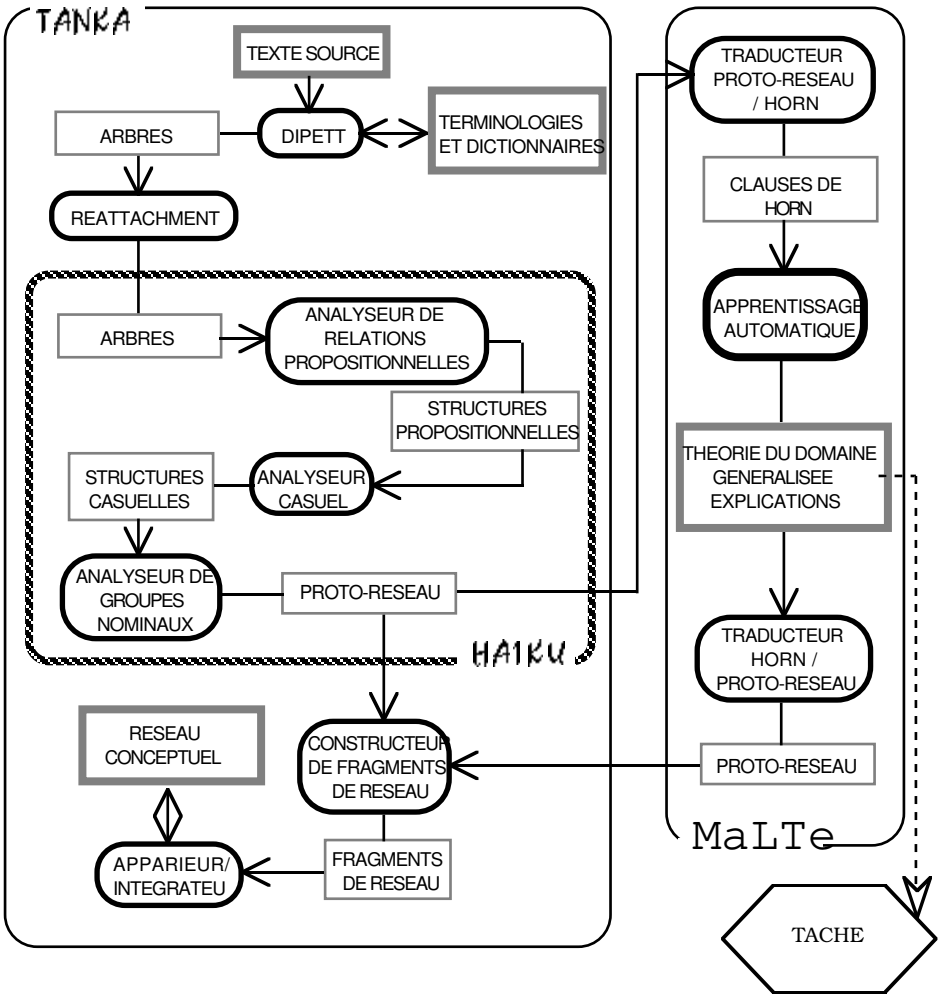

Figure 1. TANKA et MaLTe

Le système MaLTe[2] reçoit les données linguistiques du système TANKA[3]. Un arbre syntaxique de la phrase analysée, produit par DIPETT[4], peut être réorganisé par le module de réattachement de syntagmes. L'arbre d'analyse structurellement correct est traité par HAIKU, un module de traitement sémantique à trois phases. HAIKU suggère des relations sémantiques entre les propositions de la phrase, puis des structures casuelles à l'intérieur des propositions, et enfin des relations à l'intérieur des phrases. L'utilisateur peut confirmer les suggestions ou saisir

[2] Machine Learning from Text.
[3] Text Analysis for Knowledge Acquisition.
[4] Domain-Independent Parser of English Technical Texts.

d'autres données. Les relations qui n'ont pas été rencontrées précédemment sont ajoutées aux dictionnaires sémantiques de HAIKU (non représentés sur la Figure 1).Il en résulte un graphe composite contenant la description syntaxique et sémantique de la phrase, et appelé proto-réseau (*protonetwork*)—c'est-à-dire une représentation préliminaire du réseau. Toujours dans TANKA, il est transmis au Constructeur de Fragments de Réseau (*Network Fragment Builder*) qui le transforme en un réseau conceptuel fragmentaire. Ce réseau fragmentaire sera ensuite incorporé à une réseau plus vaste enrichi incrémentalement (Yang & Szpakowicz 1991) de la fraction de texte traitée jusque-là. Le proto-réseau est simultanément transmis au premier module de MaLTe, qui le traduit en un ensemble de clauses de Horn. La traduction est différente pour les parties narratives d'un texte et pour les exemples. Le module d'Apprentissage Automatique, qui comprend un moteur d'apprentissage à partir d'exemples[5], organise ces clauses de Horn pour bâtir une théorie du domaine.

Le choix de l'EBL semble particulièrement opportun dans le contexte du présent projet de recherche. Dans les textes techniques, les exemples montrent au lecteur comment intégrer la partie déclarative du texte sous forme de concepts opérationnels ou de procédures. Comme les approches traditionnelles au traitement des textes ne réalisent souvent aucun apprentissage, ni ne tentent d'exploiter les exemples contenus dans ces textes, l'EBL vient combler cette lacune. Tout d'abord, en permettant d'expliquer les exemples à partir de la théorie du domaine, et, ensuite, en généralisant ces explications et en opérationalisant la définition des concepts grâce à la compilation des éléments de connaissance nécessaires provenant de la théorie du domaine et les exemples du texte seront convertis en exemples d'entraînement pour le module EBL. L'apprentissage, à partir des exemples d'un texte, contribue donc à augmenter la base de connaissances obtenue par l'analyse linguistique de la partie déclarative du texte, en y ajoutant de nouvelles règles qui n'auraient pu être obtenues à partir de l'analyse de la partie déclarative seule.

La construction d'une théorie du domaine comporte deux tâches principales. En premier lieu, la théorie du domaine est accumulée et organisée par une hiérarchisation des clauses de Horn en un ensemble de règles stratifié dans lequel les niveaux des règles sont clairement distingués. Cela s'effectue par des transformations (via la technique d'absorption, par exemple) des ensembles de clauses, utilisées en programmation logique inductive, pour réorganiser les ensembles de clauses en des programmes logiques. La seconde tâche, qui est essentielle à l'approche suivie dans MaLTe, applique l'EBL à la représentation des exemples sous forme de clauses, qui jouent le rôle de théorie du domaine. De cette manière on obtient une interprétation compilée, généralisée et opérationnelle des exemples, qui inclut la connaissance nécessaire pour les expliquer.

Les clauses de Horn qui représentent le résultat de l'EBL sont transformées en un proto-réseau simple et réinjectées dans le Constructeur de Fragments de Réseau. Une fois que la théorie du domaine est suffisamment riche, elle peut être transmise à une tâche d'application extérieure,

[5] EBL: Explanation-Based Learning.

par exemple un programme à base de règles (ou de connaissances) servant à calculer la somme déductible du revenu. La base de règles embryonnaire contenant la connaissance dans ce programme est alors acquise directement par TANKA/MaLTe à partir du Guide d'Impôt Canadien.

Le système est implémenté en Quintus Prolog sur des stations de travail Sun Sparc. L'analyseur syntaxique et l'analyseur casuel sont entièrement implémentés. Un prototype des mécanismes d'apprentissage automatique, de l'analyseur de relations entre propositions, et du traducteur de proto-réseau en clauses de Horn a été réalisé à la fin 1993. À ce stade, le module de réattachement, le constructeur de fragments de réseau et l'analyseur de relations dans les groupes nominaux ont seulement été conçus.

3. Analyse syntaxique et sémantique

3.1 L'analyseur syntaxique

L'analyseur syntaxique accepte la plupart des phrases des textes techniques. Cette couverture permet une acquisition de connaissances raisonnablement complète. En l'absence d'un modèle sémantique riche, la syntaxe est la seule voie d'accès au sens. DIPETT (Delisle & Szpakowicz 1991, Delisle 1994) traite, intégralement ou en fragments, à peu près 90% des phrases d'un texte *non-édité*.

Les théories syntaxiques comme les GPSG, les HPSG et les LFG utilisent des structures de traits pour coder les objets linguistiques: la forme et le contenu de ces structures dépendent de la théorie sous-jacente. DIPETT n'est lié à aucune théorie linguistique particulière. Son formalisme grammatical, les Grammaires de Clauses Définies (DCG) est neutre quant à la théorie (de même que d'autres formalismes comme PATR-II). La plupart des règles de la grammaire sont basées sur Quirk *et al.* (1985). DIPETT peut être considéré comme une grammaire fonctionnelle, c'est-à-dire une grammaire dans laquelle l'analyse syntaxique est basée sur les rôles syntaxiques et non sur la position des mots dans la chaîne de surface.

Outre ses fonctions standard d'analyse, DIPETT comporte les composants suivants: un étiqueteur lexical (*tagger*) simple, un outil dynamique pour l'ajout d'entrées au dictionnaire, un utilitaire de mémorisation (c'est-à-dire une table de sous-chaînes bien formées ou *passive chart*), et un mécanisme d'explication des erreurs.

3.2 L'analyseur de Cas

Nous avons défini un système de Cas[6] que nous utilisons dans HAIKU. Six Cas sont utilisés dans les exemples de cet article: *Agent* (AGT), *Accompagnement* (ACMP), *Bénéficiaire* (BENF), *Lieu de destination*(LTO), *Objet* (OBJ), *Date* (TAT). La liste complète des 28 Cas, sa motivation

[6] Pour éviter les ambiguïtés, le terme de Cas dans cette acception sera écrit avec une majuscule.

et la discussion d'autres listes de Cas précédemment publiées apparaissent dans Barker *et al.* (1993).

L'Analyseur de Cas (abrégé en CA pour *Case Analyzer*) prend un arbre syntaxique produit par DIPETT et extrait semi-automatiquement le schéma casuel qui en représente le sens. Les Cas sont indiqués par des marqueurs de Cas, qui apparaissent sous deux formes: de façon *lexicale*, c'est-à-dire une préposition qui introduit un groupe prépositionnel, ou par *position*, comme sujet (`psubj`), objet direct (`pobj`) ou objet indirect (`piobj`).

Le CA accumule les schémas casuels dans ses dictionnaires syntactico-sémantiques et se réfère à eux pour traiter de nouvelles phrases. Une proposition dans laquelle on ne trouve guère de ressemblances avec des schémas rencontrés précédemment peut introduire de nouveaux éléments de connaissance qui sont alors intégrés aux dictionnaires. Pour une proposition similaire à d'autres déjà rencontrés, le CA suggère une interprétation sémantique que l'utilisateur peut accepter ou rejeter.

Les principales structures de donnés sont stockées dans le Dictionnaire de Significations, le Dictionnaire de Schémas de marqueurs de Cas et le Dictionnaire de Schémas de Cas (détails ci-dessous). Tous les dictionnaires peuvent être initialement vides pour un nouveau texte.

Les définitions suivantes décrivent les termes relatifs au CA utilisés dans la suite de l'article:

Un **Marqueur de Structure de Cas** (abrégé **CMP** pour *Case-Marker Pattern*) est une liste ordonnée de marqueurs de Cas représentant les marqueur apparaissant dans chaque proposition. A une proposition est associée une analyse syntaxique, à partir de laquelle est obtenu un CMP unique.

Un **Schéma Casuel** (abrégé en **CP** pour *Case Pattern*) est une liste ordonné d'abréviations de Cas, représentant les Cas qui apparaissent dans une proposition. Dans le CA, une proposition a en principe un CP unique, mais peut en avoir plusieurs si la proposition est sémantiquement ambiguë.

Le **Dictionnaire de Significations** (*Meaning Dictionary*) contient des entrées pour des mots individuels. Pour les verbes, ces entrées consistent en une liste de CMP rencontrés avec ce verbe et une liste de Cas associés avec chaque marqueur de ces CMP. Le dictionnaire de significations contient aussi des entrées pour les marqueurs de Cas de nature prépositionnelle ou adverbiale, lesquelles sont des listes fixes de Cas que le marqueur peut indiquer; une description plus précise est donnée dans Barker *et al.* (1993).

Le **Dictionnaire de Marqueurs de Schémas Casuels** (*Case-Marker Pattern Dictionary*) est constitué d'entrées indexées par un CMP et contenant une liste des CPs qui ont déjà été associés à un CMP. Chaque CP est illustré par une phrase exemple. Ce dictionnaire peut être initialisé avec quelques CMPs courant. Pendant l'analyse casuelle, un CP peut être associé à un ou plusieurs exemples, étant donné que plusieurs propositions syntaxiquement différentes peuvent avoir le même schéma casuel.

Le **Dictionnaire de Schémas Casuels** (*Case Pattern Dictionary*) est constitué d'entrées associant un CP à une liste de verbes qui lui sont associés dans le texte.

Ces dictionnaires sont mis à jour en cours de traitement. Comme indiqué ci-dessus, une proposition analysable doit normalement être associée à un CP unique. Pour cela, l'analyseur de Cas recherche d'abord dans les dictionnaires le *CP cible* qui correspond le plus au CMP de la phrase d'entrée. Puis une phrase illustrant le CP cible est prise dans le dictionnaire. Si le CP et la phrase exemple ne sont pas acceptables, le système demande l'assistance de l'utilisateur.

Prenons un exemple tiré du Guide d'Impôt Canadien:

> "Jim is a member of the Canadian Armed Forces and was posted to Lahr in 1989. Jim's wife moved with him to Lahr. He broke all residential ties with Canada. Jim is a resident of Canada because he is serving abroad in the armed forces."
>
> "Jim est membre des Forces Armées Canadiennes et a été posté à Lahr[7] en 1989. La femme de Jim l'a suivi à Lahr. Il a rompu tous liens de résidence avec le Canada. Jim est un résident du Canada puisqu'il sert à l'étranger dans les forces armées."

La dernière phrase ci-dessus contient deux propositions, qui seront traitées séparément par l'analyseur de Cas. Le verbe principal de la première proposition est le verbe d'état "be". Bien que les verbes d'état introduisent des faits portant sur les objets, les activités et leurs propriétés, ils n'ont pas de Cas. Les propositions dont le verbe principal est un verbe d'état ne seront donc pas traitées par le CA mais par des modules ultérieurs – voir Delisle *et al.* (1993). La deuxième proposition est soumise à une analyse casuelle. Le CMP `psubj_adv_in` est associé au verbe principal de la proposition, "`serve`". Le CA consulte le Dictionnaire de Significations pour vérifier si le verbe "serve" a déjà été rencontré dans le texte, et quels CPs lui ont alors été associés. A partir de ces données historiques et de l'information indiquant quels Cas sont dénotés par les Marqueurs de Cas, le CA suggère un CP à l'utilisateur. L'utilisateur peut accepter ce CP ou le remplacer par un autre. Les trois dictionnaires sont alors mis à jour pour refléter cette attribution de Cas. La sortie sera :

```
   case_structure(*statement1*, be, psubj-pobj-of, nil, "Jim", "resident",
"Canada")
   case_structure(*statement2*, serve, psubj-adv-in, agt-lat-benf, "Jim",
"abroad", "the armed forces").
```

3.3 L'analyseur de relations inter-propositionnelles

L'analyse de Cas porte sur l'interprétation sémantique des relations entre un verbe et ses arguments dans une proposition. L'information sémantique est également véhiculée par les relations entre les clauses, notamment les liens de causalité, essentiels dans la construction de règles. Nous achevons actuellement une extension de l'analyseur sémantique qui couvrira les relations inter-propositionnelles (CLRs: *Clause-Level Relationships*). La conception de l'analyseur de clauses (CLRA) est assez semblable à celle de l'analyseur casuel. Une liste de relations sémantiques à été construite à partir d'une étude détaillée des items lexicaux qui les

[7] Lahr est une base canadienne en Allemagne, maintenant progressivement démantelée.

dénotent, et en confrontant le jeu d'indices obtenu à plusieurs travaux de linguistique traditionnelle et informatique. La liste actuelle de CLRs est la suivante: Cause, Facilitation, Implication, Empêchement, Antagonisme, Conjonction, Disjonction, Co-occurrence et Antériorité (détails dans Delisle *et al.* (1993)).

Pendant l'analyse sémantique interactive, le CLRA est activé si la phrase d'entrée contient des propositions entretenant des liens syntaxiques. Le connecteur (une conjonction) est comparé à une liste de marqueurs de CLR potentiels et des Cas qui leur sont généralement associés. Un ou plusieurs CLRs sont suggérées à l'utilisateur, qui peut accepter la suggestion (ou un des ses éléments, si elle est multiple), ou indiquer une CLR différente.

Reprenons la phrase: “Jim is a resident of Canada because he is serving abroad in the armed forces” (“Jim est un résident du Canada puisqu'il sert à l'étranger dans les forces armées”). Le CLRA reconnaît deux propositions reliées par la conjonction “puisque”. Le dictionnaire de marqueurs de CLR lui indique que “because” dénote souvent la Cause, la Facilitation et l'Implication. L'utilisateur peut choisir un de ceux-ci ou saisir une autre CLR. S'il choisit l'Implication, l'analyse résultante sera:

“Jim is serving abroad in the armed forces”
<entails>
“Jim is a resident of Canada”

“Jim sert à l'étranger dans les forces armées”
<implique>
“Jim est un résident du Canada”

Le processus d'analyse sémantique est résumé dans la Figure 2.

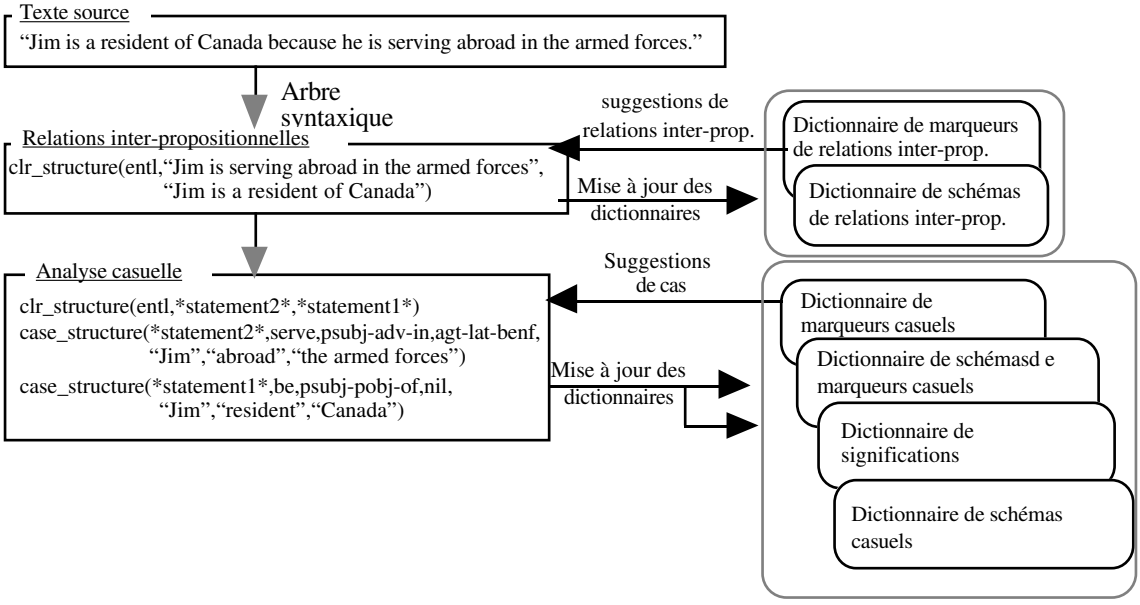

Figure 2. Vue schématique du traitement effectué par HAIKU et de sa sortie

Nous étudions actuellement l'utilisation d'indices linguistiques pour faciliter l'analyse sémantique des CLRs. La structure syntaxique des propositions elles-mêmes peut aider à

identifier leurs relations sémantiques. Par exemple, la modalité des verbes principaux peut être très utile pour distinguer les diverses relations causales, comme dans les exemples suivants: “Si vous demandez la déduction de ces frais il faut qu'ils aient été versés en échange de services” (Implication); “Si vous avez dépensé de l'argent en services, vous pouvez demander leur déduction” (Possibilité).[8]

4. Le traducteur de proto-réseaux en clauses de Horn

4.1 Construction des clauses de Horn

Contrairement à d'autres approches (voir section 5), la chaîne de conversion suivie par MaLTe contient une représentation sémantique linguistiquement justifiée, qui est ensuite traduite en logique du premier ordre. En échange d’un coût de calcul un peu supérieur, le processus est mieux fondé, plus général et portable. La généralité repose sur la séparation du traitement et de l'interaction avec l'utilisateur. La connaissance du domaine vient du texte et des choix faits par l'utilisateur, tandis que l'analyse syntaxique et l'analyse sémantique (en partie) sont pratiquement indépendantes du domaine. La traduction de la sémantique linguistique en logique par le module PtH (*Protonetwork to Horn clauses*) prend place après le réattachement des propositions et la résolution des références pronominales. L'entrée est le proto-réseau, qui comprend la structure de la phrase en termes de relations inter-propositions, la structure casuelle détaillée des propositions, et la description interne de la sémantique des groupes nominaux.

L'aspect d'une proposition (c'est-à-dire verbe d'état ou verbe d'action) est un facteur important. Les propositions statives en contexte technique sont considérées comme des définitions, comme par exemple: “Un enfant éligible peut être votre enfant, l'enfant de votre compagne, etc.”, ou peuvent décrire un attribut d'une instance (“Jim est un membre des Forces armées canadiennes.”). HAIKU n'attribue pas de Cas aux propositions à verbe d’état; elles sont traduites en un prédicat basé sur l'attribut. Aux propositions non-statives (comme “Jim a déménagé à Lahr”) correspondent des Cas dont les étiquettes sont attachées au verbe pour nommer un nouveau prédicat (comme : `serve_agt_lat_benf`, représentant servir + agent + lieu (*'location at'*) + bénéficiaire).

L'algorithme suit une description déclarative de la structure d'arbre produite par HAIKU. Son organisation correspond à peu près à la grammaire d'entrée de DIPETT, excepté que les structures à analyser ne sont pas des séquences mais des termes logiques en Prolog; les foncteurs de ces termes ne sont pas exactement des prédicats puisque leur arité peut varier suivant la présence ou l'absence de constituants syntaxiques facultatifs.

---

[8] “If you claim expenses then you *must* have paid money for services” (Entailment); “If you paid money for services then you *may* claim expenses” (Enablement).

Au niveau de la phrase, la relation inter-propositionnelle est le critère le plus important pour déterminer comment assembler les éléments d'une clause de Horn. Pour les relations de causation, de facilitation et d'implication (cette dernière étant particulièrement importante dans un guide fiscal), une règle est écrite dans la base. Si la relation entre les propositions est une simple conjonction, deux faits indépendants sont assertés. Certaines relations sont plus ambiguës, comme l'antériorité, qui peut être causale ou non. Dans tous les cas, une confirmation est demandée à l'utilisateur. Ainsi, la quatrième phrase de l'exemple:

"Jim is a resident of Canada because he is serving abroad in the armed forces"

"Jim est un résident du Canada puisqu'il sert à l'étranger dans les forces armées"

qui est représentée par:

```
clr_structure(entl, *statement2*, *statement1*),
      case_structure(*statement1*, be, psubj_pobj_of, nil,
                        "Jim", "resident", "Canada"),
      case_structure(*statement2*, serve, psubj_adv_in,
            agt_lat_benf, "Jim", "abroad", "the armed forces")
```

sera traduite sous la forme:

```
is_resident_of(jim, canada) :- serve_agt_lat_benf(jim, abroad,
armed_forces).
```

Si une proposition a une polarité négative ("X n'est pas éligible"), il faut asserter une règle qui comporte une négation explicite. Le plus fréquemment, en l'absence de relation causale ou d'opposition entre les propositions, on va simplement asserter un fait :

```
is_member(jim, canadian_armed_forces).
post_obj_lto_tat(jim, lahr, 1989).
move_agt_acmp_lto(wife, jim, lahr).
break_agt_obj_benf(jim, residential_ties, canada).
```

On peut remarquer qu'une inférence ou une interaction est nécessaire pour mettre en rapport plusieurs représentations des mêmes concepts, comme: `canadian_armed_forces` et `armed_forces`.

4.2 Apprentissage

La traduction logique de la partie descriptive et des exemples d'un texte est transmise au module d'apprentissage qui effectue une généralisation basée sur les exemples (une variante de l'EBL). Comme indiqué dans (Delannoy *et al.* 1993), le module s'appuie sur des transformations comme l'abstraction et l'absorption en Programmation Logique Inductive pour organiser les clauses de Horn en une base de connaissance hiérarchique qui soit représentative. Le processus

d'EBL part des clauses de Horn produites par le module PtH et les représentations des exemples du texte. Par exemple, supposons que la théorie du domaine contient les règles suivantes:

```
claim_child_care_expenses(P, C, E) :-
      person_deduct_expenses(P), eligible_child(C),
      deduct_amount_expenses(E).
person_deduct_expenses(P) :- is_resident_of(P, canada), eligible(P).
…
```

et que des faits supplémentaires sont produits par le module PtH. L'EBL produira un arbre de preuve représenté en partie dans la Figure 3:

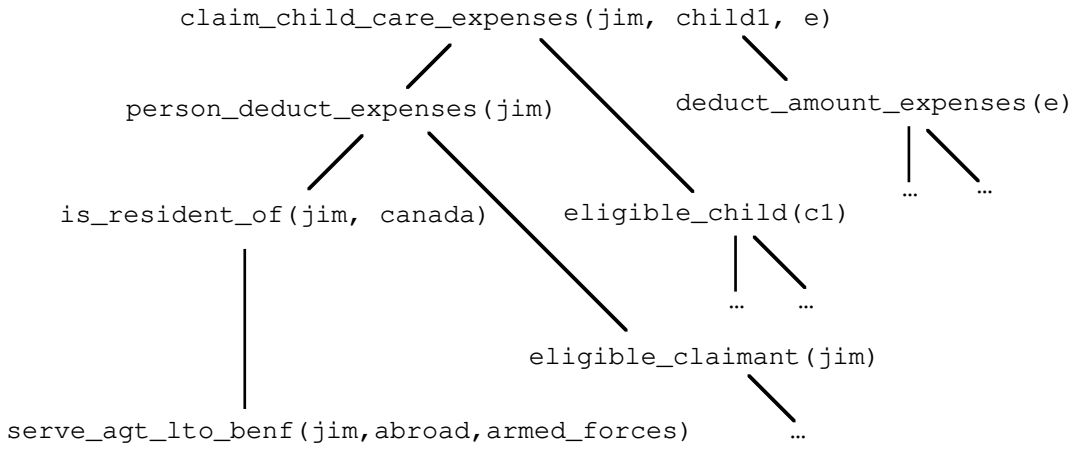

Figure 3. Un fragment de l'arbre de preuve du processus d'apprentissage EBL

Dans l'ensemble des faits disponibles à partir d'un exemple donné, l'EBL extrait ceux qui sont nécessaires pour prouver que l'exemple satisfait la définition du concept—ici une instance du concept `claim_child_care_expenses.` L'EBL met aussi en rapport (compile) toute la connaissance nécessaire pour démontrer l'appartenance de l'exemple au concept. En outre, le processus de généralisation qui a lieu dans la seconde phase de l'EBL, et consistant en une régression de la théorie du domaine dans l'arbre de preuve, produira une généralisation utile (“opérationnelle”):

```
claim_child_care_expenses(P, C, E) :-
   serve_agt_lat_benf(P, abroad, armed_forces, …).
```

Cette règle est ensuite ajoutée à la théorie du domaine, ce qui constitue un apprentissage au niveau symbolique rendant la théorie plus utile que son interprétation générale: elle s'applique maintenant *directement* à tous les membres des forces armées servant à l'étranger. Contrairement aux systèmes inductifs simples, cette généralisation est totalement justifiée par l'état de la théorie du domaine.

5. Travaux apparentés

Certains travaux analysent les textes pour en extraire une connaissance classificatoire, par exemple Silvestro (1988) et Gomez (1989). Moulin & Rousseau (1992) décrivent un système qui cherche des patrons prédéterminés (comme 'si', 'parce que', 'lorsque') dans les phrases d'entrée pour en obtenir des représentations correspondant à des règles de production ou des éléments d'une base de connaissances. Le système SINTESI de Ciravegna *et al.* (1992) extrait de la connaissance de rapports de diagnostic courts (quatre ou cinq phrases) en italien, pour résumer leur contenu technique et participer à la construction d'une base de connaissances en diagnostic de pannes d'automobiles. Tous les objets potentiellement intéressants dans le texte sont décrits dans une base disponible *a priori*; il n'y a donc pas d'enrichissement incrémental de la base.

PALKA (Kim & Moldovan (1993)) est un système d'acquisition semi-automatique conçu pour faciliter la construction d'une grande base de connaissances de schémas sémantiques à partir de corpus textuels. PALKA demande beaucoup de connaissances initiales; une hiérarchie de concepts tant généraux que liés au domaine, et des schémas (*frames*) utilisés en même temps que des mots-clés pour guider la recherche dans le texte. Liu & Soo (1993) décrivent l'attribution de rôles thématiques à des éléments de phrase en utilisant le moins possible de connaissances *a priori*. Leur système utilise des indices syntaxiques pour proposer un ensemble initial de rôles thématiques potentiels, qui est ensuite simplifié par des règles heuristiques et par l'intervention de l'utilisateur.

Considérant que le traitement de la langue naturelle par lui-même demande une grande quantité de connaissances, on peut s'étonner du faible volume de travaux d'apprentissage automatique sur des applications linguistiques. Parmi les plus récents: Hauptmann (1993) décrit un système simple d'apprentissage par mémorisation pour assister l'acquisition d'une correspondance entre syntaxe et sémantique dans un domaine donné; Zelle & Mooney (1993) et Aliprandi (1993) illustrent l'application de différentes techniques (programmation logique inductive et induction standard) à l'attachement des groupes prépositionnels. Une autre communauté de recherche (Powers 1989) met l'accent sur la difficile question de l'usage de l'apprentissage automatique pour atteindre une meilleure compréhension des aspects cognitifs de l'apprentissage humain d'une langue. Cohen (1990) illustre le rôle crucial dans l'apprentissage à partir de textes d'une définition souple de l'opérationalité. Ses travaux s'attachent surtout à l'apprentissage lui-même, et non à une intégration de l'apprentissage et du traitement de la langue naturelle.

6. Conclusion

Nous proposons une combinaison de traitement semi-automatique de textes et d'apprentissage automatique basé sur les explications pour l'extraction de connaissances à partir de textes techniques. Les premières expérimentations montrent que cette association fournit

davantage de connaissances que les méthodes standard d'acquisition de connaissances prises isolément. De nouveaux problèmes et de nouvelles possibilités surgissent simultanément du fait que la théorie du domaine est construite semi-automatiquement, directement à partir du texte. Cet enrichissement mutuel du traitement de la langue naturelle et de l'apprentissage automatique est un champ peu exploré. Des questions intéressantes sont apparues avec la conception de la première version du système. La transformation de fragments de texte en une théorie du domaine, exprimée en logique du premier ordre, et adéquate pour l'apprentissage à partir d'exemples, demande une participation intensive de l'utilisateur pour que l'acquisition de connaissance ainsi effectuée soit productive. Mais l'apprentissage par généralisation inductive des interventions de l'utilisateur doit réduire le taux d'interaction. La caractérisation de ce taux en fonction des propriétés du texte et de la saturation progressive de la base est une question ouverte. L'utilisation de la syntaxe de surface en tant que véhicule du sens est justifiée par la capacité du sous-système linguistique à travailler initialement avec une base de connaissances vide. Parmi les problèmes ouverts en apprentissage automatique figurent l'apprentissage basé sur les explications dans une théorie du domaine inévitablement incomplète, la modification dynamique du critère d'opérationalité en fonction de la tâche d'application, et l'extension du mécanisme d'apprentissage à une logique typée.

Remerciements

Ce travail est subventionné par le Conseil de recherches en sciences naturelles et en génie du Canada. Nous remercions Terry Copeck pour ses commentaires.

Références

Aliprandi, G. Saviozzi, G. (1993), "A Supervised Learning Method to Solve PP-Attachment Ambiguities in Natural Language", *Proc Machine Learning and Text Analysis Workshop, ECML-93*, Vienna, 45-52.

Angluin, D. (1988), "Queries and Concept Learning", *Machine Learning*, 2(4), 319-342.

Barker, K., Copeck, T., Delisle, S., & Szpakowicz, S. (1993), "An Empirically Grounded Case System", soumis au *International Journal of Lexicography* , 36 pages.

Ciravegna, F., Campia, P. & Colognese, A. (1992), "Knowledge Extraction from Texts by SINTESI", *Proc COLING-92*, 1244-1248.

Cohen, W.W. (1990), "Learning from Textbook Knowledge: A Case Study", *Proc AAAI-90*, 743-748.

Delannoy, J.-F., Feng, C., Matwin, S. & Szpakowicz, S. (1993), "Knowledge Extraction from Text: Machine Learning for Text-to-Rule Translation", *Proc Machine Learning and Text Analysis Workshop, ECML-93,* Vienna, 1-7.

Delisle, S. & Szpakowicz, S. (1991), "A Broad-Coverage Parser for Knowledge Acquisition from Technical Texts", *Proc 5th International Conference on Symbolic and Logical Computing — ICEBOL5* (Madison, S.D., USA), April 18-19, 1991, 169-183.

Delisle, S. (1994), "Text Processing without A-Priori Domain Knowledge: Semi-Automatic Linguistic Analysis for Incremental Knowledge Acquisition", thèse de Ph.D., Département d'Informatique—Ottawa-Carleton Institute for Computer Science, University d'Ottawa, 425 pages.

Delisle, S., Barker, K., Copeck, T. & Szpakowicz, S. (1993), "Interactive Semantic Analysis of Technical Texts: Case Pattern Acquisition", soumis à *Computational Intelligence*, 67 pages.

Feng, C., T. Copeck, S. Szpakowicz & S. Matwin (1994), "Semantic Clustering. Acquisition of Partial Ontologies from Public Domain Lexical Sources". *Proc AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop*. Banff (à paraître).

Gomez, F. (1989), "Knowledge Acquisition from Natural Language for Expert Systems Based on Classification Problem-Solving Methods", *Proc 4th AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop*, 15.1-15.18.

Hauptmann, A.G. (1993), "Meaning from Structure in Natural Language Interfaces", Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University.

Jacobs, P.S. & Rau, L.F. (1993), "Innovations in Text Interpretation", *AI Journal* 63(1-2) (Special Issue on Natural Language Processing), October 1993, 143-191.

Karp, D., Schabes, Y., Zaidel, M. & Egedi, D. (1992), "A Freely Available Wide Coverage Morphological Analyzer for English", *Proc 15th International Conference on Computational Linguistics — COLING-92* (Nantes, France), 950-955.

Kieras, D.E. (1985), "Thematic Processes in the Comprehension of Technical Prose", in B.K. Britton & J.B. Black (eds.), *Understanding Expository Text (A Theoretical and Practical Handbook for Analyzing Explanatory Text)*, LEA, 89-105.

Kim, J.-T. & Moldovan, D.I. (1993), "Acquisition of Semantic Patterns for Information Extraction from Corpora", *Proc 9th IEEE Conference on AI Applications*, 171-176.

Liu, R.-L. & Soo, V.-W. (1993), "An Empirical Study on Thematic Knowledge Acquisition based on Syntactic Clues and Heuristics", *Proc 31st Annual Meeting of the ACL*, Columbus, Ohio, 243-250.

Miller, G.A. (1990), (eds.), "WordNet: An On-Line Lexical Database", *International J of Lexicography*, 3(4).

Moulin, B. & Rousseau, D. (1992), "Automated Knowledge Acquisition from Regulatory Texts", *IEEE Expert*, October 1992, 27-35.

Powers, D. M. & Turk, C. R. (1989), *Machine Learning of Natural Language*, Springer-Verlag.

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*, Longman.

Silvestro, K. (1988), "Using Explanations for Knowledge-Based Acquisition", *International Jof Man-Machine Studies*, 29, 159-169.

Yang, L. & S. Szpakowicz (1991), "Inheritance in a Conceptual Networks". Karen S. Harber (ed.) *Proc Sixth Int Symposium on Methodologies for Intelligent Systems (Poster Session)*. Charlotte, NC, 191-202.

Zelle, J. M.& Mooney, R. (1993), "ILP Techniques for Learning Semantic Grammars", *Proc ILP Workshop, IJCAI-93*, Chambéry (France), 83-92.