Le Traitement Automatique du Langage Naturel au Service de l'Ingénieur de la Connaissance : le Système READER

Sylvain Delisle
Département de mathématiques et d'informatique
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada G9A 5H7
Sylvain_Delisle@uqtr.uquebec.ca

Résumé

Nous présentons le système linguistique d'un modèle informatique d'acquisition de connaissances selon lequel on ne peut posséder *a priori* les connaissances que l'on vise justement à acquérir d'un texte. Ce système semi-automatique est entièrement implémenté et il constitue un outil informatique utile à l'ingénieur de la connaissance qui doit identifier les éléments linguistiques et conceptuels d'un domaine pour en construire une base de connaissances. Nous discutons aussi des résultats de récentes évaluations de notre système et de travaux futurs.

Mots-clés

Analyse de texte/corpus, acquisition de connaissances, analyse syntaxique, analyse sémantique.

1 Introduction

Nous décrivons READER, un environnement logiciel qui facilite le travail d'un ingénieur de la connaissance qui cherche à identifier et définir les éléments linguistiques et conceptuels fondamentaux d'un domaine à partir de corpus. L'objectif ultime de nos travaux est la construction assistée par traitement automatique du langage naturel (TALN) d'un modèle conceptuel du domaine décrit dans un corpus et ce, avec les caractéristiques suivantes :

i) Le corpus est composé de textes réels et non de textes spécialement préparés ou adaptés. Nous nous sommes intéressés particulièrement aux textes techniques explicatifs rédigés en anglais (p.ex. manuel de l'usager, guide de l'impôt, introduction aux phénomènes météo).

ii) Nous visons à analyser le corpus avec aussi peu de connaissances spécifiques *a priori* que possible. En l'absence de telles connaissances spécialisées—que nous visons justement à extraire du texte—notre système repose essentiellement sur l'exploitation des formes syntaxiques du corpus pour proposer des interprétations à l'usager (un ingénieur de la connaissance ou un langagier).

iii) L'analyse sémantique doit être pilotée par un usager coopératif. Comme le processeur sémantique mémorise ses résultats (y compris ceux mis à jour par les interventions de l'usager), le système atteint progressivement un état de saturation de sa base de connaissances à mesure qu'il achève l'analyse du corpus. Ce faisant, la tâche de l'usager s'allège graduellement.

iv) Nous supposons que le système a accès à des ressources publiques d'informations lexicales générales : le dictionnaire *The Collins* (Karp *et al*. 92) et la base de données lexicales *WordNet* (Miller 90). *Aucun dictionnaire sémantique spécifique n'est initialement nécessaire à l'analyseur sémantique*.

2 Le Système READER

Le système READER est implémenté en Quintus Prolog 3.1 et en C sur des stations de travail Sun. Il constitue l'extension de nos travaux dans le cadre des projets TANKA (Delisle 94; Copeck *et al.* 92) et CASSCAD (Delisle 95). READER est indépendant de tout domaine particulier et est organisé en trois principaux sous-systèmes nommés LEXICOGRAPHER, DIPETT et HAIKU. Le premier assiste la construction de lexiques à partir de corpus; le lexique est ensuite utilisable par l'analyseur syntaxique DIPETT. L'analyseur sémantique HAIKU traite les arbres d'analyse syntaxique pour déterminer les relations sémantiques qui lient leurs composants.

READER est un système d'analyse de texte semi-automatique—LEXICOGRAPHER et HAIKU comportent des aspects interactifs—qui permet d'acquérir de façon incrémentale des connaissances lexicales et conceptuelles à partir de corpus. READER peut traiter linéairement tous les énoncés d'un corpus ou seulement ceux sélectionnés par l'usager. Les principaux résultats de son traitement sont : base textuelle indexée, lexique spécialisé, arbres d'analyse syntaxique, structures sémantiques Casuelles, et caractérisation syntaxique et sémantique des verbes. De plus, la représentation produite par HAIKU peut facilement être transformée en un réseau sémantique qui pourrait être fusionné à un réseau conceptuel du corpus analysé.

2.1 LEXICOGRAPHER

L'ingénierie linguistique implique la construction et la mise à jour de lexiques et de dictionnai-

res—sur leur rôle en TALN, voir (Guthrie *et al.* 96). En complément aux dictionnaires généraux, l'ingénieur de la connaissance est appelé à construire des lexiques spécifiques (soit à un domaine, soit à une langue de spécialité) à partir d'un corpus donné. Cette tâche de construction est fastidieuse, surtout lorsque le corpus est de taille ou de complexité importante. LEXICOGRAPHER permet de construire facilement un lexique à partir d'un corpus composé de textes bruts ne nécessitant aucun traitement préliminaire. Une entrée du lexique contient les informations grammaticales de base, un pointeur à une occurrence représentative dans le corpus et, optionnellement, une définition et une catégorie taxinomique (p.ex. empruntées à *WordNet).*

LEXICOGRAPHER accepte en entrée un corpus sous forme de fichier ASCII et le décortique en mots (jetons), calcule la fréquence d'occurrence de chaque mot et associe à chacun la liste de ses contextes d'occurrences (ou concordances). Lorsque l'usager procède à la construction du lexique, le système associe automatiquement à chaque mot les informations suivantes : ses fonctions grammaticales potentielles selon le dictionnaire *The Collins*, la liste de ses concordances, et ses définitions potentielles selon la base de données lexicales *WordNet*. Optionnellement, l'usager peut consulter un étiqueteur (Brill 92) ou les entrées déjà créées avec LEXICOGRAPHER dans d'autres lexiques, ou même effectuer des recherches de co-occurrences particulières dans le corpus considéré. En tout temps, l'usager contrôle la construction de chaque entrée du nouveau lexique à partir de ces diverses sources d'information qui lui sont fournies par le système.

LEXICOGRAPHER favorise la construction de lexiques minimaux qui ne contiennent que les mots et les fonctions grammaticales rencontrés dans un corpus. Ceci permet de cerner avec précision le vocabulaire d'un corpus, en plus de contribuer à réduire les conséquences nuisibles de l'ambiguïté lexicale lors de l'analyse syntaxique. Si un mot du corpus n'apparaît ni dans *The Collins* ni dans *WordNet*, il s'agit probablement d'un mot de spécialité pour lequel l'usager devra identifier de façon interactive la valeur des paramètres de son entrée lexicale (au lieu de sélectionner parmi les suggestions du système).

2.2 DIPETT

L'analyse syntaxique nous permet d'accéder aux patrons syntaxiques dont nous avons besoin pour l'analyse sémantique. DIPETT (Domain-Independent Parser for English Technical Texts) effectue une analyse syntaxique détaillée du corpus intrant (Delisle & Szpakowicz 95; Delisle 94). En l'absence de connaissances sémantiques *a priori*, une analyse syntaxique détaillée et indépendante du domaine est le seul guide fiable vers le sens des énoncés d'un corpus.

La grammaire de DIPETT est neutre du point de vue de la théorie syntaxique et son expertise linguistique est basée sur des grammaires descriptives générales de la langue anglaise, principalement (Quirk *et al.* 85). Nous avons surtout développé le sous-ensemble de l'anglais qui était pertinent à l'analyse de corpus de type technique explicatif. Les règles de la grammaire de DIPETT sont implémentées en Prolog sous la forme d'une grammaire DCG de quelques centaines de règles. Le parseur forme une grammaire fonctionnelle dont l'analyse est basée sur les rôles syntaxiques et non seulement l'ordre strict des mots. Ainsi, une phrase à la forme passive est automatiquement convertie en forme active en cours d'analyse.

DIPETT analyse un corpus dont les mots n'ont pas été lexicalement étiquetés ou annotés au préalable. Il accepte en entrée une phrase (ou un fragment) et produit en sortie un arbre d'analyse unique. Cet arbre unique n'est pas toujours parfait : le parseur utilise ses heuristiques et biais internes pour produire son analyse mais, comme il n'a accès à aucune donnée sémantique *a priori*, il peut construire un arbre plus ou moins correct du point de vue sémantique. C'est pourquoi l'usager pourra éventuellement apporter des modifications simples à l'aide du module de rattachement (travaux en cours), si cela devait s'avérer nécessaire, par exemple, pour corriger l'attachement d'un syntagme prépositionnel.

2.3 HAIKU

La partie centrale de l'analyse sémantique est l'analyse Casuelle semi-automatique et interactive de HAIKU. Les fondements de cette analyse sont présentés dans (Delisle *et al*. 96; Delisle 94); (Barker & Szpakowicz 95) présentent le *Clause-Level Relationship Analyser*, module de HAIKU qui détecte certaines relations entre les propositions. Les Cas (Fillmore 68; Somers 87) représentent les relations sémantiques entre le verbe principal d'une proposition et ses arguments syntaxiques : sujet, objet, syntagmes prépositionnels, et adverbes. Les relations nommées par les Cas correspondent à des rôles dans l'action associée au verbe. Nous avons adopté ce type d'analyse sémantique car elle permet d'établir un lien explicite entre la syntaxe et la sémantique, en plus de s'effectuer dans des termes relativement simples et intuitifs qui en rendent les concepts accessibles à l'usager du système.

HAIKU utilise un système de Cas général, indépendant de tout domaine particulier (Barker *et al*. 93). Il comporte 28 Cas (les abréviations des

Cas apparaissent après le '/'): Agent/agt, Beneficiary/benf, Experiencer/expr, Instrument/inst, Object/obj, Recipient/recp, Cause/caus, Effect/eff, Opposition/opp, Purpose/purp, Frequency/freq, Time_at/tat, Time_from/tfrm, Time_to/tto, Time_through/ttru, Direction/dir, Location_at/lat, Location_from/lfrm, Location_to/lto, Location_through/ltru, Orientation/ornt, Accompaniment/acmp, Content/cont, Exclusion/excl, Manner/man, Material/matr, Measure/meas, Order/ord.

L'analyseur Casuel de HAIKU accepte en entrée un arbre d'analyse produit par DIPETT et y associe, semi-automatiquement, le patron Casuel qui représente le mieux le sens de la phrase. Les Cas sont réalisés dans la syntaxe de deux façons : 1) de façon *lexicale* (via un marqueur explicite dans la syntaxe de surface), par exemple lorsqu'une préposition introduit un syntagme prépositionnel, ou 2) de façon *positionnelle*, via un marqueur implicite associé au sujet (psubj), à l'objet direct (pobj) ou à l'objet indirect (piobj). Tout comme pour les Cas, ce sont les symboles associés à ces marqueurs qui serviront à la construction des patrons syntaxiques.

Un *patron syntaxique* (PSY) est une liste des marqueurs syntaxiques, implicites et explicites, issus de l'analyse syntaxique et qui dénotent les Cas d'une proposition. Ces marqueurs sont unis par des tirets : p.ex., psubj-pobj-at est associé à une proposition dans laquelle le verbe principal possède un sujet, un objet direct et un syntagme prépositionnel introduit par la préposition 'at'. Toute proposition se voit associer un seul PSY. Un *patron sémantique* (PSÉ) est une liste des Cas associés à une proposition sous la supervision de l'usager. Ces marqueurs sont unis par des tirets : p.ex., agt-obj-lto est associé à une proposition dont le PSY est psubj-pobj-at et dans laquelle le verbe principal possède un sujet qui tient le rôle agent, un objet direct qui tient le rôle objet et un syntagme prépositionnel qui tient le rôle de location_to (destination). Toute proposition se voit associer un seul PSÉ.

HAIKU utilise quatre "dictionnaires" *qui peuvent être vides au début de l'analyse d'un corpus.* Dans ces circonstances, la contribution de l'usager sera plus importante initialement pour s'alléger à mesure que HAIKU augmentera ses dictionnaires de façon incrémentielle et qu'il fera des suggestions de mieux en mieux informées à l'usager. Chaque proposition se voit attribuer un PSÉ unique à la suite de l'intervention de l'usager, soit après une approbation de la suggestion du système, soit après une modification effectuée par l'usager. Pour faire une suggestion à l'usager (p.ex. "le PSÉ de la proposition courante est agt-obj-lto : veuillez approuver ou corriger"), HAIKU fouille ses dictionnaires dans le but de trouver le PSÉ qui correspond le mieux au PSY de la proposition sous considération. Pour ce faire, on utilise un algorithme de contrôle de l'analyse Casuelle (qui dirige l'analyse d'une phrase contenant plusieurs propositions) couplé à un algorithme de filtrage qui permet de trouver le ou les meilleurs PSÉ candidats en fonction du PSY selon une recherche de type "meilleur voisin".

3 Un Exemple Abrégé avec HAIKU

L'analyseur sémantique interactif HAIKU utilise quatre dictionnaires (listes structurées et indexées) qu'il met à jour dynamiquement au cours de l'analyse d'un corpus :

1) Le dictionnaire de sens (mDict) contient des entrées pour les mots individuels : verbes, prépositions ou adverbes. Pour les deux dernières catégories, le mDict contient la liste fixe des Cas qui peuvent être marqués par ces mots (Barker *et al*. 93). Pour les verbes, une entrée contient : la liste des PSY trouvés dans le corpus, ainsi que le nombre d'occurrences de chaque PSY; et la liste des Cas qui ont été associés à chaque marqueur de Cas, ainsi que le nombre d'occurrences de chaque association dans le corpus.
2) Le dictionnaire des patrons de marqueurs de Cas (cmpDict) contient une entrée pour chaque PSY. Chaque entrée associe au PSY la liste des PSÉ qui lui ont été attribués au cours de l'analyse du corpus, de même que le nombre d'occurrences de chacun des PSÉ. Chaque PSÉ est illustré par une phrase du corpus que l'usager, en cours d'analyse, aura considérée comme représentative de ce PSÉ.
3) Le dictionnaire des PSÉ (cpDict) contient une entrée pour chaque PSÉ rencontré dans le corpus et lui associe la liste des verbes qui se sont vus attribuer un tel PSÉ durant l'analyse.
4) Le dernier dictionnaire facilite l'accès aux résultats de HAIKU. Le ccvpIndex met en correspondance toutes les occurrences distinctes de PSY, PSÉ, verbe, et autres détails sur le contexte d'occurrence des patrons : le numéro de l'unité dans laquelle un PSY, un PSÉ et un verbe particuliers ont été rencontrés dans le corpus; la sous-catégorisation de surface du verbe telle que trouvée dans le corpus; et le temps du verbe dans cette occurrence.

Nous ne pouvons montrer ici le détail de toutes les entrées/sorties de READER. La Figure 1 montre une interaction avec HAIKU, après analyse syntaxique et résolution des références pronominales (si nécessaire). On peut y remarquer trois des quatre dictionnaires—le cmpDict n'est pas sollicité dans cet exemple—qui sont automatiquement mis à jour lors de l'analyse d'un texte. Les seules entrées de l'usager sont le **1** et le **Y** au milieu de la Figure 1.

```
HAIKU: Case Analysis of current input's clauses
INPUT STRING #296: "a hurricane begins over the ocean"
CURRENT SUBJ: "a hurricane"     CURRENT VERB: "begin"     CURRENT COMPL  : "over the ocean"

CMP: psubj-over (known CMP not yet associated with known verb; CP is known or new)

  Candidate CPs with example sentences previously encountered in this corpus:
     CP: expr-lat  Ex: " ... , a cloud appears over the spout"
     CP: expr-lto  Ex: "they never get over the mountains"
  current clause's main verb (infinitive):  begin
  current clause's CMP (or a permutation):    psubj-over
  most probable or closest CPs, which are of compatible length: expr-lat (1) ; expr-lto (2) ;

  if appropriate, please select one of these CPs (or 0 if none is appropriate):  1
  CMP & CP will be paired as follows: psubj/expr over/lat ; Correct [n/Y]?      Y

  UPDATING mDict for begin ...
    => begin          [psubj-over:1, psubj-pobj:6, psubj-pobj-adv:1, psubj-with:1]
                      [[adv,[tat:1]], [over,[lat:1]], [pobj,[obj:7]], [psubj,[agt:8,expr:1]], [with,[acmp:1]]]
  UPDATING cpDict for expr-lat ...
    => expr-lat   [arch, begin, blow, explode, soak]
  UPDATING ccvpIndex ...
    => psubj-over  expr-lat  begin  296

CASE STRUCTURES (simplified)
  case_structure(296, begin, psubj/expr:"a hurricane", over/lat:"over the ocean")
```

Figure 1: Interaction simplifiée avec HAIKU dans laquelle l'usager accepte une suggestion. Autrement, l'usager aurait eu à entrer le PSÉ et HAIKU aurait mémorisé cette information. (CMP=PSY et CP=PSÉ)

4 Évaluation et Applications

4.1 Validation de DIPETT et HAIKU

Nous avons effectué une validation expérimentale détaillée (Barker & Delisle 96) de DIPETT et HAIKU impliquant l'analyse en entier d'un texte technique anglais comportant un peu plus de 500 phrases (Larrick 61). Aux fins de cette évaluation, les dictionnaires de HAIKU étaient initialement vides. Nous résumons ici les principaux résultats :

* L'expérience complète a nécessité un total de 34 heures (temps réel). Le temps CPU moyen pour l'analyse syntaxique et sémantique d'une phrase a été, respectivement, de 5.8 secondes et de 2.7 secondes. En moyenne, l'usager a passé un peu moins de 2 minutes (temps réel) par phrase en interaction avec le système.
* Le texte comportait 512 phrases avec une moyenne de 10.42 mots par phrase. DIPETT a produit un arbre d'analyse pour 95% des phrases, dont 80% étaient des arbres complets, avec une sévérité moyenne d'erreur de 1.02 sur une échelle de 0 (parfait) à 3 (totalement erroné)—la justesse de l'arbre a été évaluée manuellement. La répartition des phrases par classe d'erreurs a été : 47% de type 0, 19% de type 1, 19% de type 2, et 15% de type 3. 85% des phrases analysées syntaxiquement étaient exploitables par l'analyseur sémantique.
* Des 266 pronouns nécessitant l'identification interactive de leur référent (grâce au *Pronoun Explicitizer* qui intervient entre l'analyse syntaxique et l'analyse sémantique), 110 ont vu la suggestion unique du système acceptée par l'usager, 82 ont nécessité que l'usager choisisse parmi quelques suggestions, et 74 ont obligé l'usager à identifier explicitement le référent.
* Les 512 phrases ont causé l'analyse de 468 propositions pour lesquelles le PSY était correct 69% des fois. De ces 468 interactions avec HAIKU, l'usager a accepté 55 fois le seul PSÉ suggéré par HAIKU, a choisi 217 fois un des PSÉ suggérés, a dû fournir 167 fois le bon PSÉ, et a abandonné 29 fois l'analyse Casuelle.
* Plus les phrases sont longues, plus la durée de l'interaction est longue elle aussi. Cependant, le temps moyen d'interaction diminue à mesure que l'analyse du corpus progresse.
* 4 des 28 Cas de notre ensemble n'ont pas été utilisés pour ce texte. Aucun Cas n'a semblé manquer à cet ensemble de 28.
* Le nombre moyen de PSÉ suggérés par HAIKU est de 4.47 par PSY, avec un maximum de 14. Il n'y a aucun phénomène d'explosion combinatoire lors de la manipulation des patrons syntaxiques ou sémantiques.
* Les dictionnaires de HAIKU qui étaient vides au départ contenaient le nombre d'entrées suivant à la fin : 140 pour le mDict, 83 pour le cmpDict, 122 pour le cpDict et 314 pour le ccvpIndex.

Dans une expérience complémentaire, nous avons mesuré l'avantage que procure l'analyse syntaxique complète par rapport à l'analyse par fragments aux fins de HAIKU. Lorsque DIPETT ne peut trouver une analyse complète dans un laps

de temps maximum prédéterminé, il trouve rapidement une analyse par fragments en tentant de reconnaître uniquement des phrases simples ou des syntagmes (verbaux, nominaux, prépositionnels, adverbiaux ou adjectivaux). Nous avons donc réanalysé le même texte (Larrick 61) mais exclusivement avec l'analyse syntaxique par fragments. Puis, nous avons comparé manuellement les propositions (l'unité de base pour l'analyse sémantique Casuelle) ainsi obtenues avec celles de la 1ère expérience. Voici les deux principaux résultats en ce qui concerne l'analyse Casuelle :

* Environ 60% des propositions soumises à HAIKU étaient identiques à celles de la première expérience (elles auraient mené à la même analyse Casuelle), incluant le 15% des phrases qui avaient été analysées par fragments lors de la 1ère évaluation.
* Environ 30% de ces mêmes propositions auraient mené à une analyse Casuelle réduite (parce qu'un ou des arguments du verbe manquaient).

4.2 Applications

DIPETT et HAIKU ont été conçus avec un souci d'indépendance face aux domaines d'application. Nous avons déjà appliqué DIPETT et HAIKU à une tâche de performance en apprentissage automatisé. Dans le cadre du projet MaLTe (Delisle *et al*. 94), ces deux modules constituent l'essentiel du processeur linguistique qui a pour fonction de traduire un texte en structures appropriées pour le système d'apprentissage automatisé MaLTe. Ce dernier peut ainsi convertir la sortie de HAIKU en clauses de Horn sur lesquelles il applique certains algorithmes d'apprentissage afin d'acquérir des connaissances structurées à partir du texte et de ses exemples textuels.

READER constitue aussi un système autonome utile. Le contenu des dictionnaires construits par HAIKU peut assister le langagier ou l'ingénieur de la connaissance à faire l'étude des verbes d'un corpus pour en identifier les propriétés syntaxiques et sémantiques. Ainsi, les quatre dictionnaires de HAIKU nous indiquent : les verbes d'un corpus et leurs patrons syntaxiques, sémantiques, et leurs fréquences d'occurrence respectives; leurs sous-catégorisations de surface; les verbes qui ont des patrons identiques ou similaires; les passages du corpus où apparaissent un verbe ou un PSY ou un PSÉ particuliers; les passages du corpus où apparaît un verbe donné avec un PSY ou un PSÉ particuliers; etc. En considérant que les dictionnaires de HAIKU viennent augmenter le lexique de base construit à l'aide de LEXICOGRAPHER, on peut utiliser READER comme un système semi-automatique pour construire des dictionnaires à des fins de TALN (Delisle 95). De plus, il est facile de construire des structures verbe-prédicat à partir des dictionnaires de HAIKU. Par exemple, pour le verbe 'print' (où sr_types = contraintes de sélection) :

```
verb_pred_structure(
    print,                                  % verbe
    [psubj-pobj, psubj-pobj-on],            % PSY
    [agt-obj, agt-obj-lto],                 % PSÉ
    [sr_types(np-np), sr_types(np-np-on)] ).
```

5 Conclusion

5.1 Discussion

Notre approche à l'acquisition de connaissances à partir du texte offre certains avantages par rapport aux approches dites statistiques (p.ex. Manning 93; Ribas 95) et par rapport aux approches basées sur le pré-encodage manuel des connaissances—plusieurs systèmes et approches plus ou moins automatisés sont présentés dans (Buchanan & Wilkins 93).

i) Tous les énoncés sont potentiellement porteurs d'éléments de connaissances, et *non seulement ceux qui sont associés à un certain nombre d'occurrences statistiquement significatif dans un corpus donné*. De plus, comme notre approche est de type symbolique, les résultats du système sont simples à interpréter et contribuent à former un modèle explicatif du corpus analysé.

ii) Nous estimons que l'effort total requis de la part de l'usager est moindre que celui qu'il aurait eu à fournir s'il avait eu à préparer manuellement une base de connaissances équivalente. *L'usager n'a qu'à approuver, compléter ou corriger les suggestions du système* au fur et à mesure que l'analyse du corpus progresse. L'usager est constamment guidé par le système.

iii) L'approche est applicable *même si les dictionnaires de HAIKU sont initialement vides*. Le système améliore la qualité de ses suggestions à mesure qu'il traite le corpus car le module sémantique apprend de ses analyses passées et des interactions de l'usager : il réalise un apprentissage supervisé, incrémentiel et basé sur les similarités (Aha *et al*. 91). L'usager fait partie intégrante de notre approche et sa contribution offre l'avantage de permettre une validation des entrées à la base de connaissances à un niveau qui serait autrement impossible sans une grande quantité de connaissances spécialisées *a priori*.

5.2 Quelques travaux connexes

Alors que nous avons considéré des textes techniques existants comme corpus, (Sanfilippo 94) utilise les définitions du *Longman's Dictionary of Contemporary English* et (Gomez *et al*. 94) celles du *The World Book Encyclopedia*—sur les dictionnaires et lexiques en informatique linguistique, voir (Wilks *et al*. 96; Saint-Dizier & Viegas 95). Certains travaux portent davantage sur les entrées des verbes de ces dictionnaires : (Framis 94; Grishman & Sterling 94; Manning 93) présentent des approches à l'identification automatique des

contraintes de sélection. De même, alors que nous nous sommes intéressés à l'analyse de corpus bruts, d'autres travaux partent de l'analyse de corpus pré-traités (et dépourvus de toute ambigüité) par un processeur lexical et/ou syntaxique (Ogonowski *et al*. 94). D'autres, comme (Hastings & Lytinen 94) supposent la disponibilité *a priori* des connaissances spécifiques au domaine considéré alors que nous visons justement à acquérir ces connaissances. Nous distinguons également notre approche de celles comme (Pugeault *et al*. 94) dont la stratégie d'acquisition repose sur l'analyse partielle du texte. Pour nous, l'analyse partielle est une stratégie de dernier recours après que l'analyseur a vainement tenté de trouver une analyse complète. Enfin, notre approche se démarque de celles de (Pugeault *et al*. 94; Liu & Soo 93), par exemple, dans lesquelles les Cas sémantiques assignés par le système ne peuvent pas être évalués ou corrigés par l'usager de manière à améliorer graduellement la performance ou le taux de succès de l'analyseur sémantique.

5.3 Limites et Travaux Futurs

Nous identifions ici les limites de notre approche (et de READER) et, par le fait même, suggérons des avenues pour de futurs développements et améliorations :

* Bien que semi-automatique à dessein, notre approche ne peut éliminer la possibilité que l'usager introduise des données erronées lors de la construction du lexique ou lors de l'analyse sémantique Casuelle. Par exemple, même si nous avons documenté explicitement notre système de Cas et son interprétation (Barker *et al*. 93), il n'existe pas de critères automatisables pour l'attribution "correcte" des Cas aux arguments du verbe (*via* les marqueurs syntaxiques).
* Actuellement, les connaissances acquises grâce à READER relèvent plus du domaine linguistique que conceptuel. Le processeur conceptuel prévu dans le projet TANKA n'ayant pas encore été complété, nous n'avons pu développer davantage le niveau conceptuel et ainsi mieux tester la valeur des sorties de HAIKU sur ce plan.
* Nos travaux confirment que les verbes peuvent prendre une "coloration" spécifique à un domaine (Basili *et al*. 94). Ce degré de spécificité varie, allant du sens tout à fait nouveau à une variation subtile sur un des arguments du verbe. Il n'est pas facile de détecter automatiquement ces variations ou d'ordonner les verbes selon leur degré de spécificité à un domaine. Peut-on trouver une métrique à cet effet?
* L'évaluation et la validation des sorties de READER de même que du comportement de ses sous-systèmes sont essentiellement manuelles. Il y aurait lieu d'automatiser, au moins en partie, le monitoring du système et certains tests de performance autant pour en faciliter l'entretien que pour en faciliter la comparaison avec d'autres systèmes.
* Nous avons adopté un net biais envers l'analyse syntaxique complète et détaillée. Ce type d'analyse comporte des inconvénients difficiles à maîtriser (p.ex. temps et précision de l'analyse). À la lumière des résultats de l'expérience avec l'analyse par fragments discutée en 4.1, il paraît opportun de reconsidérer le rôle de ce type d'analyse dans notre approche.
* L'analyse Casuelle n'utilise pas encore les catégories taxinomiques tirées de *WordNet* (ou créées par l'usager) lors de la construction du lexique avec LEXICOGRAPHER. Leur utilisation permettrait de rendre plus spécifiques les contraintes de sélection des verbes accumulées par HAIKU et rendrait ainsi l'analyse sémantique plus discriminante et moins superficielle.
* Le parseur DIPETT ne fait aucun usage des informations amassées dans les dictionnaires de HAIKU. Pourrait-on les utiliser (p.ex. les structures verbe-prédicat et leurs fréquences d'occurrence) afin de donner à l'analyse syntaxique un comportement prévisionnel (et probabiliste) basé sur les attentes spécifiques que les PSY nous indiquent pour chaque verbe?
* Une autre question intéressante est celle de la généralisation de notre approche : pourrait-elle être utile à la construction d'un dictionnaire à caractère *général*? Par exemple, un système comme READER pourrait assister le langagier qui souhaite construire une classification de verbes.
* Du point de vue de l'apprentissage automatisé, HAIKU n'est pas associé à des phases explicitement distinctes d'entraînement et de tests. HAIKU peut être perçu comme un système qui oscille continuellement entre ces deux phases, selon que la proposition sous analyse possède des caractéristiques qui ont déjà été rencontrées auparavant ou non. Cette situation rend les comparaisons avec d'autres systèmes similaires ou le calcul de mesures standards *(precision & recall)* assez difficiles.

Remerciements

Je remercie Ken Barker, Terry Copeck et Stan Szpakowicz du groupe de recherche KAML du département d'informatique de l'Université d'Ottawa; André Paquette (UQTR) pour son travail sur LEXICOGRAPHER et l'interface graphique de READER; et le CRSNG pour son support financier.

Références

(Aha *et al*. 91) D.W.Aha, D.Kibler & M.K.Albert, "Instance-Based learning Algorithms", *Machine Learning*, 6, 1991, 37-66.

(Barker & Delisle 96) K.Barker & S.Delisle, "Experimental Validation of a Semi-Automatic Text Analyser", Technical Report TR-96-01, Dept. of Computer Science, Univ. of Ottawa, mars 1996, 63pp.

(Barker & Szpakowicz 95) K.Barker & S.Szpakowicz, "Interactive Semantic Analysis of Clause-Level Relationships", *Proc. of the 1995 PACLING (Pacific*

Association for Computational Linguistics) Conf., Univ. of Queensland (Brisbane, Australie), 19-22 avril 1995, 22-30.

(Barker *et al*. 93) K.Barker, T.Copeck, S.Delisle & S.Szpakowicz, "A Case System for Interactive Knowledge Acquisition from Text", Technical Report TR-93-08, Dept. of Computer Science, Univ. of Ottawa, février 1993, 32pp.

(Basili *et al*. 94) R.Basili, M.T.Pazienza & P.Velardi, "The Noisy Channel and the Braying Donkey", *Proc. of the Workshop "The Balancing Act: Combining Symbolic and Statistical Approaches to Language"* (Las Cruces, New Mexico, USA), 1er juillet 1994, 21-28.

(Brill 92) E.Brill, "A Simple Rule-Based Part of Speech Tagger", *Proc. of the Third Conf. on Applied Natural Language Processing* (Trento, Italy), mars 1992, 152-155.

(Buchanan & Wilkins 93) B.G.Buchanan & D.C.Wilkins, *Readings in Knowledge Acquisition and Learning (Automating the Construction and Improvement of Expert Systems)*, Morgan Kaufmann, 1993.

(Copeck *et al*. 92) T.Copeck, S.Delisle & S.Szpakowicz, "Parsing and Case Analysis in TANKA", *Proc. of the 14th Int. Conf. on Computational Linguistics—COLING-92* (Nantes, France), juillet 1992, 1008-1012.

(Delisle & Szpakowicz 95) S.Delisle & S.Szpakowicz, "Realistic Parsing: Practical Solutions of Difficult Problems", *Proc. of the 1995 PACLING (Pacific Association for Computational Linguistics) Conf.,* Univ. of Queensland (Brisbane, Australie), 19-22 avril 1995, 59-68.

(Delisle 94) S.Delisle, "Text Processing without A-Priori Domain Knowledge: Semi-Automatic Linguistic Analysis for Incremental Knowledge Acquisition", Ph.D. thesis, Dept. of Computer Science—Ottawa-Carleton Institute for Computer Science, TR-94-02, Univ. of Ottawa, janvier 1994.

(Delisle 95) S.Delisle, "La construction de dictionnaires à partir de l'analyse informatisée de corpus bruts : un outil pour le langagier", à paraître dans *les Actes des Quatrièmes Journées Scientifiques (Lexicomatique et Dictionnairiques)*, Université Lumière Lyon-2 (Lyon, France), 28-30 septembre 1995, 11pp.

(Delisle *et al*. 94) S.Delisle, K.Barker, J.-F.Delannoy, S.Matwin & S.Szpakowicz, "From Text to Horn Clauses: Combining Linguistic Analysis and Machine Learning", *Proc. of the 10th Canadian Artificial Intelligence Conf. — CAI-94* (Banff, Alberta, Canada), 16-20 mai 1994, 9-16.

(Delisle *et al*. 96) S.Delisle, K.Barker, T.Copeck & S.Szpakowicz, "Interactive Semantic Analysis of Technical Texts: Case Pattern Acquisition", à paraître dans le numéro de mai 1996 (vol.12, #2) de *Computational Intelligence*.

(Fillmore 68) C.Fillmore, "The Case for Case" In E. Bach & R.T. Harms (éditeurs), *Universals in Linguistic Theory*, Holt, Rinehart & Winston, 1968.

(Framis 94) F.R.Framis, "An Experiment on Learning Appropriate Selectional Restrictions from a Parsed Corpus", *Proc. of the 15th Int. Conf. on Computational Linguistics—COLING-94* (Tokyo, Japan), août 1994, 769-774.

(Gomez *et al*. 94) F.Gomez, R.Hull & C.Segami, "Acquiring Knowledge from Encyclopedic Texts", *Proc. of the Fourth Conf. on Applied Natural Language Processing* (Stuttgart, Germany), octobre 1994, 84-90.

(Grishman & Sterling 94) R.Grishman & J.Sterling, "Generalizing Automatically Generated Selectional Patterns", *Proc. of the 15th Int. Conf. on Computational Linguistics—COLING-94* (Tokyo, Japan), août 1994, 742-747.

(Guthrie *et al*. 96) L.Guthrie, J.Pustejovsky, Y.Wilks & B.M.Slator, "The Role of Lexicons in Natural Language Processing", *Communications of the ACM*, 39(1), January 1996 (special section on Natural Language Processing), 63-72.

(Hastings & Lytinen 94) P.M.Hastings & S.L.Lytinen, "The Ups and Downs of Lexical Acquisition", *Proc. of AAAI-94*, 754-759.

(Karp *et al*. 92) D.Karp, Y.Schabes, M.Zaidel & D.Egedi, "A Freely Available Wide Coverage Morphological Analyzer for English", *Proc. of the 14th Int. Conf. on Computational Linguistics—COLING-92* (Nantes, France), juillet 1992, 950-955.

(Larrick 61) N.Larrick, *Junior Science Book of Rain, Hail, Sleet and Snow*, Champaign:Garrard Publishing Company, 1961.

(Liu & Soo 93) R.-L.Liu & V.-W.Soo, "An Empirical Study on Thematic Knowledge Acquisition Based on Syntactic Clues and Heuristics", *Proc. of the 31st Annual Meeting of the ACL* (Columbus, Ohio, USA), juin 1993, 243-250.

(Manning 93) C.D.Manning, "Automatic Acquisition of a Large Subcategorization Dictionary From Corpora Manuals", *Proc. of the 31st Annual Meeting of the ACL* (Columbus, Ohio, USA), juin 1993, 235-242.

(Miller 90) G.A.Miller, (éditeur), "WordNet: An On-Line Lexical Database", *International Journal of Lexicography*, 3(4), 1990.

(Ogonowski *et al*. 94) A.Ogonowski, M.-L.Herviou & E.Dauphin, "Tools for Extracting and Structuring Knowledge from Texts", *Proc. of the 15th Int. Conf. on Computational Linguistics—COLING-94* (Tokyo, Japan), août 1994, 1049-1053.

(Pugeault *et al*. 94) F.Pugeault, P.Saint-Dizier & M.-G.Monteil, "Knowledge Extraction from Texts: a method for extracting predicate-argument structures from texts", *Proc. of the 15th Int. Conf. on Computational Linguistics—COLING-94* (Tokyo, Japan), août 1994, 1039-1043.

(Quirk *et al*. 85) R.Quirk, S.Greenbaum, G.Leech & J.Svartvik, *A Comprehensive Grammar of the English Language*, Longman, 1985.

(Ribas 95) F.Ribas, "On Learning more Appropriate Selectional Restrictions", *Proc. of the Seventh Conf. of the European Chapter of the ACL—EACL-95* (Dublin, Ireland), mars 1995, 112-118.

(Saint-Dizier & Viegas 95) P.Saint-Dizier & E.Viegas (éditeurs), *Computational Lexical Semantics*, Cambridge Univ. Press, 1995.

(Sanfilippo 94) A.Sanfilippo, "Word Knowledge Acquisition, Lexicon Construction and Dictionary Compilation", *Proc. of the 15th Int. Conf. on Computational Linguistics—COLING-94* (Tokyo, Japan), août 1994, 273-277.

(Somers 87) H.L.Somers, *Valency and Case in Computational Linguistics*, Edinburgh Univ. Press, 1987.

(Wilks *et al*. 96) Y. A.Wilks, B.M.Slator & L.M.Guthrie, *Electric Words (Dictionaries, Computers, and Meanings)*, MIT Press, 1996.