Verbum XXIX, 2007, 1-2

L'INTÉGRATION DES PRÉDICATS VERBAUX DANS L'ANALYSEUR SÉMANTIQUE *TEXTBOX* : L'EXEMPLE DES VERBES DE COGNITION

Emmanuel CARTIER
Lexiques Dictionnaires Informatique (LDI)
CNRS - Université Paris 13, UMR 7187

RÉSUMÉ

Cet article décrit une application du logiciel TextBox *: le repérage automatique des expressions verbales de cognition. Il décrit tout d'abord brièvement l'architecture générale de* TextBox, *puis ses propriétés, les ressources mobilisables et les processus d'analyse. Dans une seconde partie, l'intégration d'un sous-ensemble des prédicats verbaux – les verbes de cognition – est décrite dans le détail, à partir des descriptions linguistiques de Robert Vivès : transformation des descriptions linguistiques vers un formalisme implémentable, puis exploitation de ces ressources pour le repérage automatique des prédicats verbaux dans les textes.*

ABSTRACT

This article describes an application of the TextBox software package, i.e. automatic detection of verbal expressions of cognition. Firstly, the general architecture of TextBox is described, then its properties and the resources which can be used, and the processes of analysis. In the second part the description focuses on how a subset of verb predicates (verbs of cognition) are described in detail, based on the linguistic descriptions of Robert Vivès, and on how these linguistic descriptions are transformed into a formalism which can be implemented to use these resources to detect verb predicates automatically in texts.

1. PRÉSENTATION DE L'ANALYSEUR TEXTBOX[1]

TextBox est un analyseur linguistique développé par l'auteur en 2006-2007 dans le cadre de ses activités au LDI, qui répond aux exigences suivantes :

1 Pour une présentation détaillée de TextBox, voir (Cartier, 2007).

- complète externalisation des ressources linguistiques,
- mise en place de trois étapes d'analyse linguistique : segmentation typographique en unités textuelles, analyse morphologique, analyse syntacti-co-sémantique,
- adaptabilité du système à des formalismes linguistiques variés.

Tout d'abord, TextBox propose une externalisation complète des ressources linguistiques. Cette exigence différencie TextBox d'autres plateformes d'analyse linguistique, comme NOOJ (Silberztein, 2004, 2005) ou linguastream (Widlöcher & Bilhaut, 2006). En effet, l'externalisation concerne la totalité des ressources linguistiques :

- informations définitionnelles des unités typographiques (mots, phrases, paragraphes, section, titre...), qui sont déclarées dans un fichier distinct ;
- informations morpho-syntaxiques, qui sont également déclarées dans un fichier distinct, et qui permettent une finesse descriptive "libre", en ce sens que le système n'attend pas tel ou tel type d'informations morpho-syntaxiques, c'est au concepteur du système d'analyse linguistique de modéliser ces informations ;
- informations syntactico-sémantiques : ces informations, sous forme de grammaires locales, sont également librement définies, dans un formalisme certes contraint, mais les règles de grammaire dépendent entièrement des informations précédemment explicitées, et permettent la génération d'annotations également librement définies.

Ces différents éléments font de TextBox une plateforme d'analyse linguistique apte à implémenter les descriptions linguistiques fines développées au LDI.

1.1. Architecture du système

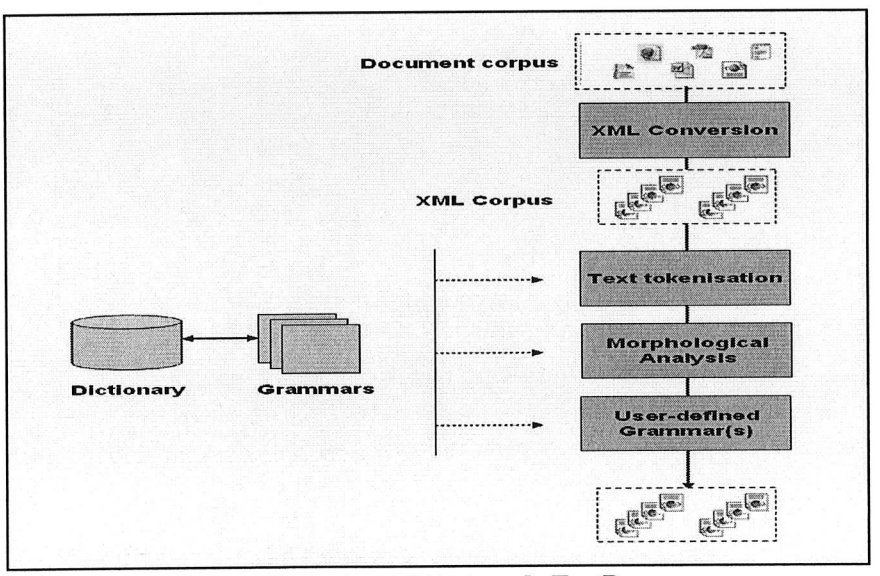

Figure 1 : architecture de TextBox

La figure précédente comprend deux zones : celle de gauche comprend les ressources linguistiques (dictionnaires et grammaires) ; celle de droite comprend les processus d'analyses linguistiques, à partir des documents sources composant le "corpus", jusqu'à la génération des fichiers "résultats", dans un format XML. Il est à noter que tout le système repose sur un fonctionnement permettant, dans le document XML, d'annoter linguistiquement les unités textuelles reconnues lors de la première étape, qu'il s'agisse de sections, de phrases, de groupes de mots ou de mots.

Les sections suivantes détaillent brièvement les processus d'analyse linguistique, puis les ressources linguistiques sont présentées.

1.2. Processus d'analyse linguistique

PROCESSUS PRÉLIMINAIRE : CONVERSION DES DOCUMENTS VERS XML

Le premier processus a pour objectif d'unifier les différents formats de documents en entrée dans un format standard qui rende compte adéquatement du contenu textuel des documents. Le format cible est le format XML, dans un encodage UTF-8.

PROCESSUS 1 : SEGMENTATION DU TEXTE EN UNITÉS TYPOGRAPHIQUES

La segmentation du texte en unités typographiques est le premier processus linguistique. Le fichier XML en entrée est doté d'annotations qui ne vont pas au-delà de la notion de paragraphe. L'objectif est donc ici d'identifier au sein des paragraphes les phrases et les mots, dans un sens typographique.

La segmentation en mots repose sur une grammaire composée de :

- mots composés et séquences figées, que l'utilisateur peut mettre à jour et modifier à sa guise ;
- expressions régulières permettant de reconnaître des classes de "tokens" : expressions numériques (5,7%, 13,2 millions...), expressions temporelles (12/03/2006, 13h34...), expressions spécifiques (urls, mails, etc.).

L'avantage principal de cette grammaire est son externalisation, permettant une adaptation aisée.

PROCESSUS 2 : PROJECTION DES INFORMATIONS DICTIONNAIRIQUES

L'analyse morphologique est classiquement le processus suivant dans une analyse linguistique. Dans TextBox, cette phase, qui opère par projection d'informations contenues dans un dictionnaire, est plus générale qu'une simple projection d'informations morphologiques : en effet, d'une part, l'analyse morphologique peut être réduite à son minimum, selon l'objectif général de repérage de l'utilisateur ; d'autre part, d'autres traits que des traits morphologiques peuvent être attachés à chaque entrée du dictionnaire et donc projetés sur les occurrences dans les textes.

En effet, dans TextBox, cette projection va ajouter des informations aux mots reconnus dans la phase précédente. Mais, à son tour, cette projection va être la base du processus suivant, l'application des grammaires locales. Or ces grammaires n'ont généralement pas besoin de l'ensemble des informa-

tions morpho-syntaxiques et/ou sémantiques mais seulement d'une partie d'entre elles.

Par exemple, si l'on souhaite repérer des noms de personnes, il suffit d'associer aux mots pertinents pour ce type de repérage des informations de type sémantique : prénoms, fonctions (général, cheminot,...), titres (M. Mme...). Par la suite, des grammaires locales permettront de repérer les noms de personnes comme suit :

token[@sem='forename'] + token[1,5[@typo='tc'][2]

Par contre, si l'on souhaite reconnaître des groupes nominaux dans un corpus, il sera nécessaire d'encoder dans le dictionnaire les différentes instances de parties du discours, afin de décrire dans les grammaires locales les différentes séquences syntagmatiques propres au groupe nominal, par exemple, pour les suites Adj + N :

token[@cat='det'] + token[@cat='adj'] + token[@cat='n']

Qu'est ce qui est en jeu ici ? La capacité du linguiste à déterminer son propre jeu d'informations linguistiques à projeter sur les mots de son corpus, jeu lié à son objectif de repérage. L'objectif d'analyser automatiquement l'ensemble des phénomènes linguistiques d'un texte, qui nécessitera un maximum d'informations linguistiques à encoder dans le dictionnaire, n'est dès lors qu'un cas particulier.

Dans TextBox, donc, l'étape d'analyse morphologique est en fait une projection de traits sur les mots (simples ou composés).

Gestion de l'ambiguïté :

Dans le dictionnaire, si l'on définit plusieurs entrées homographes (*porte*, comme nom et verbe par exemple), le système générera les deux possibilités d'analyse, laissant la résolution de l'ambiguïté à l'étape suivante, qui gère et utilise le contexte via les grammaires locales. Voici par exemple la sortie qui apparaîtra pour le token *porte* :

```
<token type='w' typo='lc'>
 <morph lemma='porter' cat='v' mood="subj' tense='pres' pers='1'/>
 <morph lemma='porter' cat='v' mood="subj' tense='pres' pers='3'/>
 <morph lemma='porter' cat='v' mood="ind' tense='pres' pers='1'/>
 <morph lemma='porter' cat='v' mood="ind' tense='pres' pers='3'/>
<morph lemma='porte' cat='n' gender='f' number='s'/>
 porte
</token>
```

2 Brièvement : "token" correspond à des éléments linguistiques (mots simples ou composés) reconnus par segmentation typographique du texte ; dans les crochets droits sont indiquées des contraintes sur les tokens : par ex. "@sem='forename'" indique qu'un attribut 'sem' (sémantique) a pour valeur forename, ie prénom ; "@typo='tc'" indique que du point de vue typographique, le token est en "title case" (première lettre des mots en majuscules). Une quantification des tokens est également possible (par exemple [1,5] signifie qu'il peut se rencontrer ce token entre une à cinq fois.

PROCESSUS 3 : GRAMMAIRES LOCALES

Les grammaires "utilisateurs" ou encore grammaires locales – pour reprendre un terme bien implanté dans la littérature TAL et qui de plus est le fondement de cette étape d'analyse dans TextBox – , sont la partie la plus innovante du système, à la fois pour leur pouvoir expressif et pour leur capacité à se ramener aux standards XPATH/XSL.

De fait, le formalisme utilisé permet théoriquement d'identifier dans un texte *toute séquence textuelle ainsi que toute propriété linguistique*. Cela va de la reconnaissance de séquences discontinues au repérage des antécédents d'anaphoriques, ou encore le calcul des évolutions et ruptures thématiques dans un texte.

En effet, le formalisme utilisé pour exprimer des grammaires locales combine la puissance des expressions régulières et des grammaires de traits, propriétés du langage XPATH, mais y adjoint :

- la gestion des itérations d'éléments ;
- une simplification des expressions XPATH.

Pour plus de détail, nous renvoyons à l'article de Cartier (2007).

Nous avons décrit dans cette section les fondations de l'outil TextBox. Dans la section suivante, nous exposons brièvement le modèle de représentation adopté pour décrire les prédicats verbaux.

Nous présentons ensuite les différentes ressources linguistiques utilisées pour intégrer les prédicats verbaux dans le système, à savoir les dictionnaires puis les grammaires. Une dernière section évoque deux séries de tests menées pour valider les descriptions syntactico-sémantiques.

2. ANALYSE DES DESCRIPTIONS LINGUISTIQUES : MODÉLISATION LINGUISTIQUE

2.1. Modèle adopté

Les descriptions linguistiques des prédicats verbaux sont menées dès le départ selon un "semi-formalisme" intentionnel qui joue à plusieurs niveaux :

1/ Au niveau du modèle descriptif gouvernant les descriptions des différentes classes sémantiques :

Le modèle, initialement très riche, a été ramené à une grille plus simple à mettre en œuvre dans une phase initiale comprenant minimalement :

Attribut	Exemple
Nom de la classe sémantique	Méconnaissance
Définition	"Produire un résultat erroné dans une opération intellectuelle."
Liste des verbes concernés	*abuser (s'), avaler, aveugler (s'), confondre...*
Liste de schémas syntactico-sémantiques décrivant les différentes combinatoires syntactico-sémantiques. Pour chaque instance :	
Schéma formel	N0<hum> V (N1<ina> + le fait que P)
Liste des verbes conformes au schéma	*méconnaître, méjuger, mésestimer, négliger, sous-estimer, sous-évaluer, surestimer, surévaluer*
Exemples	Marc (méconnaît + méjuge + mésestime + néglige + ...) (les difficultés + le fait que le climat se réchauffe)

De plus, d'autres informations sont disponibles, notamment les reconstructions, permettant de rendre compte des variantes passives, causales, etc. Nous ne les exploiterons pas ici.

2/ Au niveau de l'explicitation du schéma syntactico-sémantique propre à un emploi :

Les schémas syntactico-sémantiques décrivent des suites de mots selon leur combinatoire et leurs propriétés sémantiques. Le formalisme adopté consiste à énumérer la suite d'éléments syntagmatiques correspondant à un sens donné, au moyen d'un jeu de catégories morpho-syntaxiques "classiques", telles que Nom, Verbe, Adverbe, Déterminant, etc.

A ces catégories générales peuvent s'adjoindre des informations de sous- catégorisation sémantique, en termes de classes d'objets.

Nous aboutissons alors à des schémas qui, dans leur forme la plus simple, exhibent des groupes sous-catégorisés, comme par exemple N0[hum] V N1[hum]. La numérotation des groupes nominaux explicite les relations de sujet grammatical (N0), d'objet 1 (N1) ou d'objet second (N2).

2.2. Modèle résultant dans TextBox

Le modèle proposé ci-dessus permet d'envisager la création d'une base de données qui pourra être utilisable pour créer et maintenir les ressources développées par les linguistes, mais également pour la reconnaissance automatique dans les textes via TextBox. Voici une proposition de structure de cette base de données, qui devrait être implémentée comme un axe futur des recherches au LDI :

Table générale :
Schéma : Id verbe - verbe lemmatisé
Dans cette table, un id identifie chaque lemme verbal. Toute autre information, qu'elle soit de nature morphologique ou sémantique comme ici, est décrite dans d'autres tables.

Table des classes sémantiques :
Schéma : classe – hyperclasse
La table des classes sémantiques permet de stocker les différentes classes et hyperclasses auxquelles peuvent être liés les lemmes. Elle ne concerne pas spécifiquement les prédicats verbaux, mais plutôt l'ensemble des mots "sémantiques" de la langue.

Table des informations syntactico-sémantiques :
Schéma : Id verbe – classe – schéma syntactico-sémantique
La table des informations syntactico-sémantiques est une table liée à la première via l'id. Dans cette table sont stockées les différentes associations schémas syntactico-sémantiques – classe sémantique (ie sens). Par ailleurs, d'autres informations, telles que la définition "naturelle" ou des exemples, peuvent être insérées ici.

Voici un extrait de cette table avec certains verbes de cognition :

VERBE	CLASSE SÉMANTIQUE	SCHÉMA SYNTACTICO-SÉMANTIQUE
noter	Mémorisation	(N0<hum> + la mémoire de N<hum>)V (que P + N1<ina>)
oblitérer	Oubli	N0<hum> V (N1<hum,ina> + que P)
omettre	Oubli	N0<hum> V (N1<inc> + de Vinf + que Vmodal Vd'action)
opiner	Opinion	N0<hum> V que P
oublier	Oubli	N0<hum> V (N1<inc> + de Vinf + que Vmodal Vd'action)
oublier	Oubli	N0<hum> V (N1<hum,ina> + que P)
passer	Opinion	N0<hum,in> V pour [être] (N1<hum,in> + Adj)
photographier	Mémorisation	(N0<hum> + la mémoire de N<hum>) V (que P + N1<ina>)

Une telle structure peut par ailleurs, comme nous le verrons plus loin, être directement utilisée dans TextBox.

3. INTÉGRATION DES REPRÉSENTATIONS SÉMANTIQUES

Nous présentons ci-dessous les différentes briques mises en place pour intégrer les représentations syntactico-sémantiques des prédicats verbaux

dans TextBox. Cette section se divise en deux parties : une partie "Dictionnaires", qui évoque les ressources dictionnairiques mobilisées pour arriver à l'identification automatique des sens des verbes prédicatifs sur la base des schémas syntactico-sémantiques ; une partie "Grammaires" qui évoque le passage des schémas vers le formalisme TextBox.

Une dernière section évoquera les tests effectués sur corpus pour valider ces schémas.

3.1. Dictionnaires

Les dictionnaires de TextBox sont librement définis par l'utilisateur en fonction de son but de repérage. Dans le cadre de l'intégration des prédicats verbaux, deux niveaux sont nécessaires :

- Dictionnaire pour la segmentation : ce dictionnaire permet de segmenter les textes en tokens, définis somme unités typographiques : dans ce cadre, il s'agit de découper le texte en unités linguistiques minimales sur des bases typographiques : font donc partie de ce dictionnaire l'ensemble des mots composés non ambigus ;
- Dictionnaires permettant d'affecter aux "tokens" des attributs de type morpho-syntaxique et sémantique. Il s'agit ici des attributs qui sont la base des grammaires de reconnaissance, étant donné que les schémas syntactico-sémantiques utilisent des informations morpho-syntaxiques (partie du discours essentiellement) et sémantiques (classe essentiellement).

3.1.1. Dictionnaire des tokens

Les tokens sont les unités linguistiques minimales, définies d'un point de vue typographique : en font donc partie les mots simples, comme *de* ou *chat*, mais également les mots composés non discontinus et non ambigus, comme *pomme de terre* ou *chien de garde,* mais non "est nécessaire de", puisqu'entre l'auxiliaire et l'attribut peuvent s'insérer des adverbiaux.

Par ailleurs, ce dictionnaire utilise des expressions régulières permettant de reconnaître de manière automatique les différents signes présents dans un texte : expressions numériques ("12", "13,4"...), expressions temporelles (12/12/2007, 13h34...), et d'autres types spécifiques (URL, formule mathématique...).

Avec ce dictionnaire, TextBox construit, à partir d'un texte, une représentation XML où chaque token est identifié et pourvu d'un attribut "sign" qui définit sa catégorie de signes parmi les suivants : num (numérique), temp (temporel), word (mot). Des sous-attributs peuvent être définis permettant de sous-catégoriser certains types de tokens (par exemple subtype=time pour les expressions temporelles dénotant une heure, ou subtype=date pour ceux qui dénotent une date).

Après segmentation en tokens, voici la sortie XML correspond à la phrase de départ : *Pierre pensait qu'il était trop tard.*

```
<?xml version="1.0" encoding="UTF-8" ?>
- <doc source="D:/Text_analyzer/text_Analyser/corpus/test.txt" name="test.txt.seg.xml">
  - <p>
      <token typo="tc" sign="word" sem="forename">Pierre</token>
      <token typo="lc" sign="word">pensait</token>
      <token sign="word">qu'</token>
      <token typo="lc" sign="word">il</token>
      <token typo="lc" sign="word">était</token>
      <token typo="lc" sign="word">trop</token>
      <token typo="lc" sign="word">tard</token>
      <token type="other" sign="punct">.</token>
    </p>
  </doc>
```

Figure 2 : TextBox - exemple de sortie XML après segmentation

3.1.2. Dictionnaire morpho-syntaxique

Le dictionnaire morpho-syntaxique comprend des informations morpho-syntaxiques pour chaque lemme, auquel sont associées ses différentes formes. De ce fait, dans les textes, chaque forme (token) qui comprendra une ou plusieurs entrées dans le dictionnaire morpho-syntaxique se verra affecter les différents attributs associés aux lemmes correspondants.

Le dictionnaire utilisé ici est le fruit du travail de Michel Mathieu-Colas. Il comprend plus de 700 000 formes simples et près de 250 000 formes composées.

Cette étape est nécessaire dans le cadre de l'intégration des prédicats verbaux, car les schémas utilisent les notions de groupes ou syntagmes (nominaux, verbaux, adjectivaux etc.) qui ne peuvent être reconnues qu'après projection d'informations morpho-syntaxiques.

Après passage par l'analyseur morphologique, voici la sortie XML correspondant à la phrase de départ : *Pierre pensait qu'il était trop tard.*

```
<?xml version="1.0" encoding="UTF-8" ?>
- <doc source="D:/Text_analyzer/text_Analyser/corpus/test.txt" name="test.txt.morph.xml">
  - <p>
      <token typo="tc" sign="word" sem="forename">Pierre</token>
    - <token typo="lc" sign="word">
        <morph tense="Ind-imp" cat="V" pers="3" number="S" lemma="penser" />
        pensait
      </token>
    - <token sign="word">
        <morph cat="conj" lemma="que" />
        <morph cat="PRO" lemma="que" />
        qu'
      </token>
    - <token typo="lc" sign="word">
        <morph cat="PRO" pers="3" number="S" lemma="il" gender="M" />
        il
      </token>
    - <token typo="lc" sign="word">
        <morph tense="Ind-imp" cat="V" pers="3" number="S" lemma="être" />
        était
      </token>
    - <token typo="lc" sign="word">
        <morph cat="ADV" lemma="trop" />
        trop
      </token>
    - <token typo="lc" sign="word">
        <morph cat="ADV" lemma="tard" />
        tard
      </token>
      <token type="other" sign="punct">.</token>
    </p>
  </doc>
```

Figure 3 : TextBox - exemple de sortie XML après analyse morphologique

3.1.3. Dictionnaire sémantique

Dans ce dictionnaire, chaque mot se verra affecter au minimum deux types d'informations : d'une part son type sémantique, parmi les valeurs suivantes : Prédicat, Argument, Actualisateur ; d'autre part, sa classe sémantique, en termes de classe d'objets.

Il est à noter que le modèle permet une affectation multiple : un mot peut être à la fois prédicat et argument ; un mot peut également – c'est même le cas des noms, adjectifs et verbes les plus fréquents – appartenir à plusieurs classes d'objets.

Voici, à titre d'exemple, après passage par l'analyseur sémantique, la sortie XML correspondant à la phrase de départ : *Pierre pensait qu'il était trop tard.*

```
<?xml version="1.0" encoding="UTF-8" ?>
- <doc source="D:/Text_analyzer/text_Analyser/corpus/test.txt" name="test.txt.morph.xml">
 - <p>
  - <PRED sem="croire(X,P)" schema="SN[sem='hum'] SV[lemma='penser'] que P" X="Pierre" P="qu'il était trop tard">
     <phrase type="NP" head="Pierre" sem="forename">Pierre</phrase>
     <phrase type="VP" head="penser" tense="Ind-imp" cat="V" pers="3" number="S" lemma="penser">pensait</phrase>
    - <clause type="compl">
     - <token sign="word">
        <morph cat="conj" lemma="que" />
        qu'
       </token>
       <phrase type="PRON" head="il" pers="3" number="S" lemma="il" gender="M">il</phrase>
       <phrase type="VP" head="être" tense="Ind-imp" cat="V" pers="3" number="S" lemma="être">était</phrase>
     - <token typo="lc" sign="word">
        <morph cat="ADV" lemma="trop" />
        trop
       </token>
     - <token typo="lc" sign="word">
        <morph cat="ADV" lemma="tard" />
        tard
       </token>
       <token type="other" sign="punct">.</token>
     </clause>
   </PRED>
  </p>
</doc>
```

Figure 4 : TextBox - exemple de sortie XML après analyse sémantique

Après les phases de segmentation en tokens puis la projection des informations morpho-syntaxiques et sémantiques, notre document se voit décoré d'une foule d'informations qui peuvent être exploitées par les grammaires.

3.2. Grammaires

Pour l'intégration des prédicats verbaux, deux jeux de règles de grammaire sont nécessaires :

- une série de règles de reconnaissance des groupes syntagmatiques ;
- une série de règles de reconnaissance des schémas syntactico-sémantiques.

Avant de présenter ces deux séries de règles, nous devons évoquer la transformation à effectuer entre le formalisme des prédicats verbaux et le formalisme TextBox.

3.2.1. Transformation des schémas dans le formalisme TextBox

Les schémas syntactico-sémantiques obéissent à un formalisme décrit dans la section 2. Les grammaires de TextBox répondent à un autre formalisme. Il convient donc de transformer si possible le premier formalisme dans le second, de manière automatique.

Voici un exemple de schéma dans les deux formalismes, pour le verbe *noter* (classe "mémorisation") :

ANALYSE LINGUISTIQUE :

(N0<hum> + la mémoire de N<hum>)V (que P + N1<ina>)

TEXTBOX

1. P[3][type='NP' and sem='hum'] P[type='VP' and lemma='noter'] que C
2. token[.='la'] token[.='mémoire'] token[.='de'] P[type='NP'] P[type='VP' and lemma='noter'] que C
3. P[type='NP' and sem='hum'] P[type='VP' and lemma='noter'] P[type='NP' and sem='ina']
4. token[.='la'] token[.='mémoire'] token[.='de'] P[type='NP'] P[type='VP' and lemma='noter'] P[type='NP' and sem='ina']

Sans entrer dans des considérations trop techniques, la grande différence entre les deux formalismes réside dans l'action subséquente à la reconnaissance du schéma. Pour l'analyse linguistique, il s'agit d'attribuer une classe sémantique à la séquence. Pour TextBox, une règle de grammaire aboutit à une action totalement libre : en l'occurrence, il s'agirait de créer un élément encadrant l'ensemble de la séquence et de lui attribuer une valeur sémantique, et, éventuellement, d'arborer la structure sémantico-syntaxique de l'ensemble. Par exemple (en omettant les balises intermédiaires) :

<pred sem='mémoriser(X,Y)' X='Pierre' P='la porte est ouverte'>

Pierre note que la porte est ouverte

</pred>

Actuellement, la représentation sémantique associée à un schéma se limite à une classe et une définition.

3.2.2. Grammaires de reconnaissance des groupes syntaxiques

Étant donné que les schémas syntactico-sémantiques utilisent comme éléments constitutifs des groupes syntaxiques, il est nécessaire de reconnaître ces derniers préalablement à l'application des schémas eux-mêmes.

---

3 P=phrase (groupe), C=clause (proposition au sens de prédicat + arguments + actualisateurs), NP=noun phrase, VP = verbal phrase.

De ce point de vue, et étant donné que TextBox est encore un logiciel "jeune", développé depuis mi-2006, nous avons utilisé des règles en cours d'élaboration permettant de repérer :

- des groupes nominaux : 17 règles ;
- des groupes adjectivaux : 13 règles ;
- des groupes adverbiaux : 5 règles ;
- des groupes verbaux : 13 règles.

Ces jeux de règles ont été mis en place et testés sur le corpus *Le Monde 1991-2002* annoté au LDI.

Cela signifie que, à l'heure actuelle, TextBox n'est pas capable de reconnaître des phrases. Cela a pour seul effet que les règles subséquentes qui utilisent cet élément ne seront pas reconnues globalement, le système ne sera capable de repérer que le début de la proposition.

3.2.3. Grammaires de reconnaissance des schémas syntactico-semantiques

Le travail sur les verbes de cognition présente 151 lemmes verbaux pour 486 schémas syntactico-sémantiques. Nous renvoyons à l'article de Robert Vivès pour une description détaillée de cette classe de verbes.

4. ÉVALUATION DES SCHÉMAS SYNTACTICO-SÉMANTIQUES

Dans cette dernière section, nous décrivons deux séries de tests conduits sur corpus pour évaluer la précision et le rappel des schémas syntactico-sémantiques.

Nous présentons tout d'abord les objectifs généraux de cette évaluation, puis évoquons les deux évaluations mises en œuvre.

4.1. Objectifs de l'évaluation

Tout d'abord, il faut noter que ce travail liminaire ne prend pas en compte toute la complexité des phrases réelles, puisque les schémas syntactico-sémantiques ne décrivent que des phrases "canoniques", en ce sens qu'en sont exclus tous les phénomènes de reconstruction, mais également toutes les insertions pouvant avoir lieu entre les constituants (par exemple les adverbiaux et les incises).

L'objectif de ce travail est donc seulement de vérifier le principe selon lequel les schémas sont capables de désambiguïser des sens verbaux.

4.2. Évaluation sur corpus interne (exemples liés aux descriptions)

Description du corpus interne :

Le corpus interne est constitué de l'ensemble des exemples fournis dans les descriptions des prédicats verbaux de cognition. Un travail préliminaire a consisté à "nettoyer" ces exemples, afin de leur rendre leur forme "naturelle".

De l'exemple :
Marc a résolu (le problème + le mystère + l'énigme + le cas des sans-papiers)

Nous sommes ainsi passés à quatre phrases :
Marc a résolu le problème.
Marc a résolu le mystère.
Marc a résolu l'énigme.
Marc a résolu le cas des sans-papiers.

Au total, nous sommes arrivés à un total de 1034 phrases exemples.

Résultats :

Pour le corpus interne, sur les 1034 phrases, 973 ont été correctement étiquetées, soit 94,10% d'étiquetage correct.

Un tel résultat est particulièrement remarquable et montre la capacité des schémas syntactico-sémantiques à désambiguïser des structures prédicatives.

Trois grandes familles d'erreurs ont été rencontrées :

1. Erreurs de reconnaissance des groupes nominaux complexes :

En effet, sur certains exemples, principalement dans les cas de deux groupes objets, le premier groupe nominal est incorrectement reconnu :

Marc a enfin trouvé le fin mot de l'histoire dans le passé de la victime.
N0<hum> V[trouver] N1<hum,inc> Prép N2<loc>
Ici, le système n'est pas parvenu à récupérer correctement les deux groupes nominaux objets.

2. Erreurs de reconnaissance dues à une classe sémantique mal étiquetée

N0<hum> V[considérer] N1<hum,in> comme [étant] N2<hum,in>
Marc considère ce tableau comme une croûte
Ici, le mot *croûte* n'avait pas été étiqueté comme étant un inanimé.

3. Erreurs dans les schémas :

N0<hum> V[chercher] N1<hum,in> [Prép N2<loc,in>]
Marc cherche ses clés dans la foule.
Ici, *foule* ne représente pas un lieu au sens concret.

5. CONCLUSIONS ET PERSPECTIVES

Le travail que nous venons de présenter avait pour but d'amorcer une intégration des descriptions sémantiques dans un analyseur automatique. L'expérience menée sur l'intégration dans TextBox des schémas syntactico-sémantiques s'avère probante, puisque à la fois nous avons pu utiliser le formalisme proposé, et vérifier sur un corpus interne la validité des schémas.

Cependant, ce travail liminaire doit se poursuivre dans au moins trois directions :

1/ Evaluation du système sur un corpus externe : nous sommes actuellement en cours de construction d'un corpus externe construit à partir des archives du *Monde* (1995-2002) disponibles sur CD-ROM. Nous en avons extraits 1500 phrases contenant les différentes formes verbales de *penser, trouver, juger, considérer, oublier, rappeler* ;

2/ Formalisation systématique des descriptions et intégration des données constituantes (classes sémantiques, groupes syntaxiques) dans une base de données centralisée : en effet, c'est à ce prix que les représentations des prédicats pourront être pleinement exploitées ;

3/ Formalisation des phénomènes de transformations, ainsi que de l'ensemble des phénomènes d'intégration phrastique : il s'agit ici, comme le révèlera l'évaluation sur corpus externe, d'une direction obligatoire pour rendre pleinement exploitables les données actuellement disponibles.

Aujourd'hui, le LDI a construit et continue d'affiner une base de données unique et exploitable de représentations syntactico-sémantiques des verbes. Il convient de la faire fructifier.

BIBLIOGRAPHIE

BLANCO X. & BUVET P.-A. (2004), "Verbes supports et significations grammaticales. Implications pour la traduction espagnol-français", *Lingvisticae Investigationes*, 27(2), John Benjamins B.V., Amsterdam.

BUVET P.-A., CARTIER E., ISSAC F. & MEJRI S. (2007), "Dictionnaires électroniques et étiquetage syntactico-sémantique", *Actes du colloque TALN 2007*, Toulouse.

CARTIER E. (2007), "TextBox, a Written Corpus Tool for Linguistic Analysis", *Web As Corpus 2007*, Louvain-la-Neuve.

CUNNINGHAM H., MAYNARD D., BONTCHEVA K. & V. TABLAN (2002), "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications", *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia.

GROSS G. & VIVÈS R. (1986), "Les constructions nominales et l'élaboration d'un lexique-grammaire", *Langue française*, 69, Larousse, Paris, 5-27.

GROSS M. (1997), "The Construction of Local Grammars", E. Roche & Y. Schabes (eds), *Finite-State Language Processing*, Cambridge, Mass./London, The MIT Press, 329-352.

LE PESANT D. & MATHIEU-COLAS M. (1998), "Introduction aux classes d'objets", *Langages*, 131, Larousse, Paris.

PIAO S.L., ARCHER D., MUDRAYA O., RAYSON P., GARSIDE R., MCENERY T. & WILSON A. (2005), "A Large Semantic Lexicon for Corpus Annotation", *Proceedings of the Corpus Linguistics 2005 conference*, July 14-17, Birmingham, UK. Proceedings from the Corpus Linguistics Conference Series on-line e-journal, Vol. 1, no. 1, ISSN 1747-9398.

SILBERZTEIN M. (2004), NooJ : A Cooperative, Object-Oriented Architecture for NLP, INTEX pour la Linguistique et le traitement automatique des langues. Cahiers de la MSH Ledoux, Presses Universitaires de Franche-Comté .

SILBERZTEIN M. (2005), "NooJ's Dictionaries", *The Proceedings of LTC 2005*, Poznan University.

WIDLÖCHER A. & BILHAUT F. (2006), "La plate-forme LinguaStream", *Actes du Colloque International des Etudiants Chercheurs en Didactique des Langues et en Linguistique*, Grenoble, France.