Raffinement du Lexique des Verbes Français

Paul Bédaride
Université de Stuttgart
paul.bedaride@gmail.com

RÉSUMÉ

Nous présentons dans cet article les améliorations apportées à la ressource « Les Verbes Français » afin de la rendre plus formelle et utilisable pour le traitement automatique des langues naturelles. Les informations syntaxiques et sémantiques ont été corrigées, restructurées, unifiées puis intégrées à la version XML de cette ressource, afin de pouvoir être utilisée par un système d'étiquetage de rôles sémantiques.

ABSTRACT

Resource Refining : « Les Verbes Français »

This paper introduce the impovements we made to the resource « Les Verbes Français » in order to make it more usable in the field of natural language processing. Syntactic and semantic information is corrected, restructured, unified and then integrated to the XML version of this resource, in order to be used by a semantic role labelling system.

MOTS-CLÉS : ressource, lexique, verbes, raffinement, étiquetage de rôles sémantiques.

KEYWORDS: resource, lexicon, verbs, refinement, semantic roles labeling.

1 Introduction

Le domaine du traitement automatique des langues naturelles nécessite à la fois des ressources représentant les particularités des langues et de leurs sémantiques, ainsi que des théories d'analyse utilisant ces ressources. Si les théories sont la plupart du temps rapidement adaptables d'une langue à l'autre, il n'en est pas de même pour les ressources. En effet, les ressources devant représenter la richesse d'une langue, il est nécessaire de réaliser une nouvelle analyse des cas problématiques ou une nouvelle annotation de corpus. Si l'on peut dire que l'anglais est une langue riche en ressources linguistiques (e.g. : PropBank, FrameNet, WordNet, ...), il n'en est pas de même pour le français. Il existe des équivalents pour certains types de ressources (e.g. : FrenchTreeBank, EuroWordNet, Wolf, ...), mais ils ont souvent une moins grande couverture et une moins bonne qualité. Si les ressources linguistiques ne sont pas très développées, c'est parce qu'elles exigent des analyses d'experts du domaine et des annotations nécessitant une quantité considérable de temps et de travail, qui engendre un fort coût de production. D'un autre coté, il existe des ressources linguistiques méconnues et sous-utilisées car nécessitant des efforts conséquents pour être adaptées au traitement automatique des langues. Plutôt que de laisser stagner ces ressources et de créer de nouvelles ressources à partir de rien, nous avons choisi d'améliorer l'une d'entre elles – « Les verbes français » (Dubois et Dubois-Charlier, 1997),

Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2: TALN, pages 155–168,
Grenoble, 4 au 8 juin 2012. ©2012 ATALA & AFCP

avec pour objectif de la rendre utilisable pour la tâche d'étiquetage de rôles sémantiques. Nous allons maintenant introduire cette ressource en évoquant ses qualités et faiblesses, ainsi que les améliorations déjà réalisées, puis nous expliquerons pourquoi cette ressource nous semble être un bon choix pour l'étiquetage de rôles sémantiques. Dans les sections suivantes nous présenterons notre objectif de ressource, les traitements réalisés pour atteindre cet objectif, ainsi qu'une évaluation de ces améliorations. Nous conclurons cet article en résumant les gains que nos travaux ont apportés à cette ressource.

2 Les Verbes Français

2.1 Historique

Le *LVF*, « *Les Verbes Français* », est une ressource lexicale réalisée par Jean Dubois et Françoise Dubois-Charlier, dont l'objectif est de fournir une description linguistique des verbes, basée sur l'adéquation entre schèmes syntaxiques et interprétation sémantique (Levin, 1993). Cette ressource a été confiée dans un premier temps à un industriel qui n'a pas su l'utiliser, ainsi qu'à un éditeur qui ne l'a pas publié (elle fut distribuée sous la forme de photocopies). Comme ces deux entreprises détenaient les droits d'utilisation, cette ressource ne pouvait pas être diffusée afin d'être largement utilisée et améliorée. Ils ont cependant accepté après un certain temps, de restituer aux auteurs le droit de diffuser leur ressource comme ils le souhaitaient.

Depuis sa libération en 2007 sous la forme d'un fichier Excel (*eLVF*), un nombre croissant de personnes se sont intéressées à cette ressource. Un des premiers constats réalisé fut que Jean Dubois et Françoise Dubois-Charlier ont développé le *eLVF* sans se soucier des problèmes que les informaticiens pourraient avoir pour utiliser leur ressource dans le traitement automatique de la langue française. Une grande partie des problèmes est cependant due aux limites d'espace de stockage existantes lors de la création du *LVF*. Un grand nombre de mots ont ainsi été tronqués ou abrégés. La représentation de la ressource sous la forme d'un tableau limite sa structuration, et les formats utilisés pour certains champs ne sont pas assez formels. Enfin, l'ouvrage « Les verbes français » est obligatoire pour comprendre tous les codes et formats utilisés dans la version Excel du *LVF*(*eLVF*).

2.2 Description

Le *eLVF* est composé de 25 609 entrées représentant les différents sens de 12 310 verbes. Il y a 4 188 verbes à plusieurs entrées et un verbe peut avoir jusqu'à 61 entrées (e.g. :pour le verbe *« passer »*). Une entrée est composée des onze champs suivants :

- MOT : entrée du verbe à l'infinitif
- DOMAINE : code donnant l'emploi principal (géologie, psychologie, . . .)
  et le niveau de langue (familier, vieux, littéraire, . . .)
- CLASSE : code définissant la classe syntactico-sémantique (appartenant à une hiérarchie)
- OPÉRATEUR : définition syntactico-semantique de l'entrée
- SENS : synonymes et définitions abrégées
- PHRASE : exemples d'utilisation de ce sens

M	DOM	CLA	OPER	SENS	PHRASE	C	CONST	DER	N	L
amasser 01	OBJ	L3b	*lc.qp qc+pl e amas*	accumuler	On a~des documents,des livres. Les preuves s'a~contre P.	1aZ	T1801 P8001	-1- -D ---- --	3*	2
amasser 02	MON	L4b	*lc.qp arg e tas*	accumuler	On a de l'argent,de l'or.. On a~sans cesse.L'argent s'a~	1aZ	T1301 P3001	--- -- ---- --	-	5
amasser 03(s)	LOC	U1a	*(qn+pl)li.simul qp*	se grouper	La foule s'a~sur la place.	1aZ	P7001	--- -- ---- --	-	5

TABLE 1 – Entrées du *eLVF* pour le verbe *« amasser »*

- CONJUGAISON : codes permettant de conjuguer le verbe
- CONSTRUCTIONS : codes pour obtenir les schèmes de construction syntaxique
- DÉRIVATIONS : codes pour produire les adjectifs verbaux et les dérivés nominaux
- NOM : code pour produire le mot dont le verbe est dérivé
- LEXIQUE : code pour obtenir le type de dictionnaire où l'entrée est répertoriée

La table 1 recense les entrées du *eLVF* représentant les différents sens du verbe *« amasser »* (i.e. : « accumuler des objets », « accumuler de l'argent », « se grouper quelque part »). Le *eLVF* donne les informations suivantes sur le premier sens de ce verbe : il est utilisé dans le DOMAINE des objets ; sa CLASSE sémantique est *mettre quelque chose quelque part, dans/sur un lieu, autour de quelque chose* ; son OPÉRATEUR signifie *être ou mettre quelque part plusieurs choses en amas* ; il a pour synonyme le verbe *« accumuler »*, il est réalisé dans les PHRASES d'exemple *« On amasse des documents. »* et *« Les preuves s'amassent contre Pierre. »* ; il fait partie des verbes du premier groupe avec un auxiliaire *« avoir »* pour le transitif et *« être »* pour le pronominal ; il se réalise dans un cadre transitif avec un sujet humain, un objet choses, et un circonstant locatif (où on est) ainsi que dans un cadre pronominal avec sujet choses et un circonstant locatif ; l'adjectif *« amassable »* et le déverbal *« amas »* en sont des dérivés ; et il provient d'un dictionnaire de base de 15 000 mots. Une bonne connaissance des encodages est clairement nécessaire pour dériver ces informations à partir de l'entrée du *eLVF*. Nous n'allons pas décrire plus précisément tous les champs[1], car cela serait trop long, mais nous allons nous focaliser sur les champs les plus utiles pour la tâche d'étiquetage de rôle sémantiques : les champs opérateur et constructions.

Le champ OPÉRATEUR interprète sémantiquement les schèmes syntaxiques. Il est composé d'un prédicat (le premier token qui n'est pas entre parenthèses), et d'un certain nombre d'arguments. Les définitions des prédicats et de certaines abréviations sont accessibles dans la version papier du *LVF*. Le sujet du prédicat, qui est optionnel, est défini entre parenthèses juste avant le prédicat. Les autres arguments suivent le prédicat et peuvent être formés d'un ou plusieurs mots. Les limites des arguments n'étant pas définies, c'est à l'utilisateur d'identifier les différents arguments à l'aide de ses connaissances de la langue et de ses capacités de raisonnement. Il existe deux types d'arguments : les contraintes syntaxico-sémantiques pouvant être réalisés syntaxiquement et les spécifications sémantiques précisant la sémantique du prédicat. Un argument est une contrainte syntaxico-sémantique s'il est le sujet du prédicat, s'il appartient à un certain ensemble d'abréviations (e.g. : *qc*, *qn*, ...) ou s'il est composé d'une préposition en capitales ; et est une spécification sémantique dans le reste des cas. Par exemple, dans la première entrée du verbe *« amasser »*, *qc+pl* représente une contrainte alors que *e amas* représente une spécification. Des opérateurs de disjonction (i.e. : */*,) permettent d'associer plusieurs contraintes à un même emplacement sémantique.

Le champ CONSTRUCTIONS liste les différents schémas syntaxiques pouvant être réalisés. Un schéma syntaxique commence par une lettre en capitale, qui définit son type et celui de ses

1. Pour une description complète de la ressource : `http://margaux.philosophie.uni-stuttgart.de/lvf/`

A	intransitif	sujet + circonstant
N	transitif indirect	sujet + complément prépositionnel
T	transitif direct	sujet + obj. direct + cpl. prép. + circonstant
P	pronominal	sujet + obj. direct + cpl. prép. + circonstant

TABLE 2 – Type de CONSTRUCTION avec arguments associés

arguments. La table 2 donne les quatre types de schèmes existant dans le *LVF* avec le type de leurs arguments. Chaque argument encode une contrainte syntaxique ou sémantique par un chiffre ou une lettre. Un sujet ou un objet ayant pour valeur *1* représente une contrainte sémantique sur le domaine de l'*humain*, un circonstant avec la valeur *1* représente une contrainte sémantique sur le domaine *locatif (où on est)* et un complément prépositionnel ayant pour valeur *b* représente une contrainte syntaxique sur un complément prépositionnel avec la préposition *« de »*. Nous donnerons uniquement la signification des codes que nous utiliserons dans cet article, et nous vous renvoyons à la version papier de *LVF* ou à notre wiki [1] si vous souhaitez étudier les autres codes.

L'espace de stockage étant nettement moins problématique de nos jours, il serait intéressant de décoder la ressource pour la rendre plus lisible et utilisable. C'est ce qu'ont commencé à faire Hadouche et Lapalme (Hadouche et Lapalme, 2010) en transformant le *eLVF* au format XML ainsi qu'une interface de consultation [2], comme nous allons le voir dans la sous-section suivante. Dans leur article présentant la version XML du *eLVF* (*xLVF*), ils ont comparé le *eLVF* avec des ressources existantes pour l'anglais (*VerbNet*(Schuler, 2005), *FrameNet*(Baker *et al.*, 1998), *WordNet*(Fellbaum, 1998)) et le français (*Dicovalence*(Mertens, 2010)). Pour résumer ce comparatif, nous pouvons dire que les approches utilisées par les ressources sont variées (pronominale pour *Dicovalence*, distributionnelle et transformationnelle pour le *LVF*, à base de cadres sémantique pour *FrameNet*, et d'ensembles de synonymes pour *WordNet*), mais qu'elles ont toutefois un certain nombre de points communs dans leur représentation de la description des unités lexicales, et de la hiérarchisation des données. D'après cette comparaison le *eLVF* est une bonne ressource linguistique car il intègre une grande partie des informations contenues dans les autres ressources, comme la hiérarchisation du sens des verbes avec les CLASSES, la description de la sémantique avec le champ OPÉRATEUR, et une liste de SYNONYMES pour chaque entrée. Il manque cependant certaines informations comme la gestion des rôles thématiques, mais le *eLVF* propose des informations que les autres ressources ne contiennent pas comme la gestion du sens figuré des verbes.

2.3 La version XML

Nous allons maintenant parler de la version XML du *eLVF* développée par Hadouche et Lapalme. Cette version du *eLVF* a pour objectif de rendre la ressource plus accessible, utilisable et extensible. Pour cela ils ont informatisé la description des différents codes contenus dans la version papier du *LVF* sous la forme de fichiers XML (voir figure 1). Ils ont ensuite généré un fichier XML représentant les données du *eLVF* décompressées. La figure 2 nous montre la version XML du verbe *« amasser »*. Les informations d'origine ont été conservées dans le fichier à l'intérieur des balises et les codes décompressés sont représentés par des attributs associés à ces balises. Toutes les informations n'ont pas été complètement décompressées, comme pour le code de CLASSE *L3b* dont nous savons qu'il représente la classe générique *Locatif* avec un sujet *non-animé propre*

2. http://rali.iro.umontreal.ca/Dubois/

```
<classes>
  <generique code="C" desc="communication">
    <semantico-syntaxique code="C1" desc="s'exprimer par un son, une parole">
      <sous-classe-syntaxique code="C1a" desc="émettre un cri, humain ou animal"/>
      <sous-classe-syntaxique code="C1b" desc="émettre un chant, humain"/>
      ...
    </semantico-syntaxique>
    <semantico-syntaxique code="C2" desc="dire/demander qc">
      <sous-classe-syntaxique code="C2a" desc="dire que, dire qc à qn"/>
      <sous-classe-syntaxique code="C2b" desc="dire que, donner un ordre à qn"/>
      ...
    </semantico-syntaxique>
    ...
  </generique>
  ...
</classes>
```

FIGURE 1 – Codes du *xLVF* pour les CLASSES

mais pas qu'il représente *mettre quelque chose quelque part, dans/sur un lieu, autour de quelque chose*. Certains champs ont aussi été restructurés comme les PHRASES, les CONSTRUCTIONS et les DÉRIVATIONS où les informations ont été séparées. Il est dommage que Hadouche et Lapalme aient uniquement intégré la description des codes au XML, car des traitements comme la dérivation des adjectifs, des adverbes et des noms auraient aussi pu être intégrés à cette nouvelle version.

2.4 Le *LVF* pour l'étiquetage de rôles sémantiques

Le *LVF* n'a pas été conçu pour l'étiquetage de rôles sémantiques, mais il contient néanmoins des informations pertinentes pour cette tâche. Les champs OPÉRATEUR, CONSTRUCTIONS et DOMAINE donnent des informations sur la syntaxe, la sémantique et l'utilisation des différentes entrées associées à un verbe. L'exploitation de ces informations devrait permettre l'identification du sens utilisé et de projeter les arguments syntaxiques sur une représentation sémantique (i.e. : le champ opérateur). Un système utilisant cette ressource serait différent de ceux existants, basés essentiellement sur de l'apprentissage automatique appliqué à de grand corpus annotés, car il utiliserait les contraintes syntaxiques et sémantiques définies manuellement par Jean Dubois et Françoise Dubois-Charlier. Le champ PHRASE pourrait être utilisé comme corpus d'exemples permettant une première évaluation d'un système d'étiquetage de rôles sémantiques. Dans un premier temps, nous allons définir notre objectif de restructuration de la ressource, puis nous décrirons les différents traitements effectués pour l'atteindre et nous terminerons sur une évaluation de la ressource obtenue.

3 Objectif

Pour qu'un système puisse utiliser le *xLVF* pour faire de l'étiquetage de rôles sémantiques, il doit être capable d'en extraire les informations nécessaires. Il est aussi important de pouvoir faire le lien entre les différents types d'information de la ressource, ainsi qu'avec des informations contenues dans d'autres ressources (e.g. : *Wolf*, *Disco*, *French TreeBank*). Il est donc nécessaire de restructurer et d'uniformiser un certain nombre d'informations du *xLVF*. De plus, il serait intéressant d'utiliser les phrases d'exemple afin de concevoir un corpus annoté, même si celui-ci n'est pas forcément représentatif. La figure 3 montre ce que l'on souhaite obtenir pour la première

```
<verbe mot="amasser" nb="3" id="amasser">
  <entree>
    <mot no="1">amasser 01</mot>
    <domaine nom="objet">OBJ</domaine>
    <classe generique="locatif" semantico-syntaxique="non-animé propre"
            construction-syntaxique="b">L3b</classe>
    <operateur>lc.qp qc+pl e amas</operateur>
    <sens>accumuler</sens>
    <phrases>
      <phrase>On <lexie-ref>a~</lexie-ref> des documents,des livres.</phrase>
      <phrase>Les preuves s'<lexie-ref>a~</lexie-ref> contre P.</phrase>
    </phrases>
    <conjugaison auxiliaire="avoir (sauf si pronominal ou entrée en être)"
   groupe="1" sous-groupe="chanter">1aZ</conjugaison>
    <construction>
      <scheme type="transitif direct" sujet="humain" objet="pluriel chose"
              circonstant="locatif (ou on est)">T1801</scheme>
      <scheme type="pronominal" sujet="pluriel chose"
              circonstant="locatif (ou on est)">P8001</scheme>
    </construction>
    <derivation der-e="positif seul"
     der-ment="il n'y a pas de nom en -ment mais il y a un déverbal">
      -1- -D ---- --
    </derivation>
    <nom nb="3">3*</nom>
    <lexique desc="dictionnaire de base" nbmots="15000">2</lexique>
  </entree>
  ...
</verbe>
```

FIGURE 2 – Entrée du *xLVF* pour le verbe *« amasser »*

entrée du verbe *« amasser »*.

Il est nécessaire d'uniformiser la ressource pour deux raisons. Premièrement pour qu'un système utilisant la ressource ne traite pas différemment une information qui aurait deux représentations distinctes, comme les abréviations *lgt* et *lgts* pour *« longtemps »* ou *poissons* et *poisson+pl* pour représenter le pluriel de *« poisson »*. La seconde raison est de rendre la ressource plus interopérable avec d'autres ressources. Pour identifier les arguments d'un prédicat, les contraintes sémantiques du *xLVF* peuvent être exploitées, mais il est nécessaire d'utiliser des ressources comme *Wolf* (Sagot et Fišer, 2008) ou *Disco* (Kolb, 2009) pour associer un type sémantique aux arguments (e.g. : humain, chose, ...). Ces ressources n'ayant pas connaissance des abréviations utilisées par le *xLVF*, il est nécessaire de remplacer les abréviations et les mots tronqués par les lemmes les représentant et de préférer la représentation du pluriel par l'utilisation de l'attribut *« +pl »*.

L'étiquetage de rôles sémantiques a pour but d'identifier les arguments du prédicat et leur associer des rôles sémantiques. L'utilisation de contraintes syntaxiques et sémantiques est une des solutions envisageables pour réaliser cette tâche. Pour cela, il est important d'avoir des informations syntaxiques et sémantiques structurées et inter-connectées pour chaque sens de chaque verbe. Ces informations doivent permettre de projeter la syntaxe sur la sémantique et inversement. Il est aussi important de bien séparer les informations syntaxiques et sémantiques, afin de permettre une meilleure cohérence de la ressource. Dans le *xLVF*, ces informations sont mélangées, et le même type d'information peut se retrouver à différents endroits, ce qui peut mener à des incohérences. Le champ OPÉRATEUR sera utilisé comme base pour la sémantique et le champ CONSTRUCTEUR comme base pour la syntaxe.

Le cadre sémantique est défini comme un prédicat auquel sont associés des arguments et des contraintes sémantiques. Les contraintes sémantiques ont des identifiants permettant d'associer

ses arguments à ceux des cadres syntaxiques. Le prédicat a pour valeur un des différents prédicats du champ OPÉRATEUR (e.g. : *r.d* : rendre/devenir tel). Les arguments sémantiques précisent la sémantique du prédicat et ne peuvent pas être réalisés syntaxiquement. Les contraintes sémantiques permettront de définir le type des différents arguments du prédicat (e.g. : un humain, une chose, un animal) qui pourront être réalisés syntaxiquement.

Les cadres syntaxiques sont ceux utilisés par le champ CONSCRUCTIONS (i.e. : intransitif, transitif indirect, transitif direct, pronominal). Les arguments sont définis par un type, un complément selon le cas (e.g. : une préposition) ainsi que des liens vers les arguments sémantiques réalisés. Un argument syntaxique peut réaliser plusieurs arguments sémantiques. Ainsi, dans la phrase « Marie se maquille », on a une variation syntaxique pronominale réfléchie et *Marie* endosse les rôles d'agent et d'expérient.

Le corpus d'exemples sera composé de phrases simples mettant en avant les différentes façons de réaliser les différents sens de chaque verbe. Ces phrases seront analysées en dépendances, et posséderont des annotations permettant de savoir quel cadre syntaxique leur est associé et à quels arguments sémantiques correspondent leurs arguments syntaxiques.

4 Transformation

Nous allons maintenant décrire les différents traitements appliqués au *xLVF* pour atteindre notre objectif. La production de cette nouvelle ressource est composée de sept étapes : l'uniformisation du champ OPÉRATEUR, sa structuration, la récupération de données à partir de la version papier du *LVF*, l'alignement des CONSTRUCTIONS, la liaison des champs OPÉRATEUR et CONSTRUCTIONS, la répartition de l'information et enfin la construction du corpus d'exemples.

L'étape d'uniformisation du champ OPÉRATEUR permet de corriger les abréviations et les mots tronqués. Un ensemble de mots suspects n'apparaissant pas dans les noms et adjectifs de *Morphalou* (Romary *et al.*, 2004) a été récupéré automatiquement (947 mots). Les occurrences de chaque mot ont été examinées manuellement afin d'identifier s'il s'agissait d'une abréviation, d'un mot tronqué ou mal orthographié, ou encore d'un mot technique n'existant pas dans *Morphalou*. Une définition ou une correction a ensuite été associée à chaque mot. Les différentes orthographes d'un mot ont été homogénéisées dans la définition afin qu'il y ait une représentation unique par sens. La définition *longtemps* a ainsi été associée aux abréviations *lgt* et *lgts*. Les mots ayant plusieurs orthographes (e.g. : acuponcture, acupuncture) ont été unifiés à l'aide de *Morphalou* et du *Wiktionnaire*, pour que les verbes ayant des orthographes différentes aient là même sémantique (e.g. : « *acuponcturer* » et « *acupuncturer* »). La gestion du pluriel a été uniformisée en prenant le singulier des mots et en leur ajoutant l'attribut *+pl*. Le mot *plantes* est ainsi devenu *plante+pl*.

La seconde étape consiste à traiter le champ OPÉRATEUR afin d'obtenir une structure proche de celle du cadre sémantique défini précédemment. Pour cela, nous avons dû identifier le prédicat et ses arguments, déterminer pour chaque argument s'il représentait une contrainte syntaxico-sémantique ou une spécification sémantique, et discerner la portée des disjonctions. Nous avons choisi d'utiliser des méthodes d'analyse de surface car elles sont robuste et facilement maintenable. Ces aspects sont importants, car le champ OPÉRATEUR comporte un grand nombre de cas particuliers que nous avons dû gérer au fur et à mesure de leur rencontre. Dans un premier

```
<CadreSemantique>
  <Contrainte cidx="0" desc="humain" />
  <Predicat desc="être ou mettre quelque part" />
  <Contrainte cidx="1" pluriel="True" desc="chose" />
  <Semantique sidx="0" prep="en" desc="amas" />
  <Contrainte cidx="2" desc="locatif (où on est)" />
</CadreSemantique>

<CadresSyntaxiques>
  <CadreSyntaxique type="transitif direct">
    <Argument type="sujet">
      <LienSemantique cidx="0" />
    </Argument>
    <Argument type="objet">
      <LienSemantique cidx="1" />
    </Argument>
    <Argument type="circonstant">
      <LienSemantique cidx="2" />
    </Argument>
  </CadreSyntaxique>
  <CadreSyntaxique type="pronominal">
    <Argument type="sujet">
      <LienSemantique cidx="1" />
    </Argument>
    <Argument type="circonstant">
      <LienSemantique cidx="2" />
    </Argument>
  </CadreSyntaxique>
</CadresSyntaxiques>

<Phrases>
  <Phrase text="On  a~ des documents.">
    <Dep dep="root" wid="2" form="a~" lemma="a~" pos="V" srl="transitif direct" >
      <Dep dep="suj" wid="1" form="On" lemma="on" pos="CL" cidx="0" />
      <Dep dep="obj" wid="4" form="documents" lemma="document" pos="N" cidx="1">
        <Dep dep="det" wid="3" form="des" lemma="un" pos="D" />
      </Dep>
      <Dep dep="ponct" wid="5" form="." lemma="." pos="PONCT" />
    </Dep>
  </Phrase>
  <Phrase text="On  a~ des livres.">
    <Dep dep="root" wid="2" form="a~" lemma="a~" pos="V" srl="transitif direct">
      <Dep dep="suj" wid="1" form="On" lemma="on" pos="CL" cidx="0"/>
      <Dep dep="obj" wid="4" form="livres" lemma="livre" pos="N" cidx="1">
        <Dep dep="det" wid="3" form="des" lemma="un" pos="D" />
      </Dep>
      <Dep dep="ponct" wid="5" form="." lemma="." pos="PONCT" />
    </Dep>
  </Phrase>
  <Phrase text="Les preuves s' a~ contre P.">
    <Dep dep="root" wid="4" form="a~" lemma="a~" pos="V" srl="pronominal">
      <Dep dep="suj" wid="2" form="preuves" lemma="preuve" pos="N" cidx="1">
        <Dep dep="det" wid="1" form="Les" lemma="le" pos="D" />
      </Dep>
      <Dep dep="aff" wid="3" form="s'" lemma="il" pos="CL" />
      <Dep dep="mod" wid="5" form="contre" lemma="contre" pos="P" cidx="0">
        <Dep dep="obj" wid="6" form="Pierre" lemma="pierre" pos="N" />
      </Dep>
      <Dep dep="ponct" wid="7" form="." lemma="." pos="PONCT" />
    </Dep>
  </Phrase>
</Phrases>
```

FIGURE 3 – Objectif d'amélioration

1	f.ire/PRD abs/ABR SR/PP	faire aller quelque chose d'abstrait sur	focaliser l'attention sur
2	abda/PRD chemin/MOT abs/ABR A/PP qn/ABR	enlever chemin abstrait à quelqu'un	couper la route du succès
3	ict/PRD total/MOT soi/ABR abs/ABR	frapper totalement soi abstrait	se suicider
4	abda/PRD pr/PP soi/ABR abs/ABR	obtenir pour soi quelque chose d'abstrait	acquérir de l'expérience
5	loq/PRD AV/PP qn/ABR D/PP //DIS SR/PP prix/	parler avec quelqu'un de prix, sur le prix	marchander
6	dat/PRD A/PP qc/ABR ,/DIS A/PP +inf/ATT	donner à quelque chose, à (infinitif)	contribuer
	contrainte syntaxico-sémantique argument sémantique disjonction		

TABLE 3 – Structuration du champ OPÉRATEUR

temps nous avons étiqueté les tokens pour abstraire les règles de l'analyse de surface. L'étiquette utilisée par défaut représente un mot (MOT). Les autres étiquettes définissent les prédicats (PRD), les prépositions (PP), les abréviations (ABR), les attributs (ATT), et les différents symboles (e.g : (),/-). L'étiquetage a été réalisé grâce à des lexiques définis manuellement et en fonction de l'emplacement des tokens. L'analyse de surface a ensuite été réalisée à partir d'un ensemble de règles générales, basées sur les étiquettes, et d'un ensemble de règles spécifiques, basées sur les mots. Les trois gros problèmes de la structuration de ce champ furent le regroupement de tokens pour former les arguments, la gestion de la portée des disjonctions, et le typage des arguments. Des règles générales permettent de regrouper des suites de mots et des prépositions avec le groupe à leur droite. Il existe cependant des cas où les abréviations et les mots peuvent être regroupés entre eux comme les exemples 2 et 3 de la table 3. La tâche n'est pas triviale car le regroupement de certains tokens est dépendant du contexte. L'abréviation *abs* peut être reconnue comme un argument canonique désignant un concept abstrait (e.g. : exemple 1, *« une idée »*), où être associée à un autre token pour l'abstraire (e.g. : exemple 2, *« le chemin du succès »*. Son association dépend de son emplacement, du prédicat de l'OPÉRATEUR, et du token auquel elle peut se lier. Des règles spécifiques utilisant des lexiques ont été mises en œuvre pour gérer ces cas particuliers. Les exemples 5 et 6 montrent que les disjonctions peuvent avoir des portées plus ou moins grandes. Le choix de la portée des arguments se fait en fonction des étiquettes des groupes adjacents à la disjonction. La portée courte se fait en priorité sur les disjonctions ayant des tokens adjacents avec les mêmes étiquettes (e.g. : exemple 5). La portée longue se fait à la fin en prenant les groupes adjacent à la disjonction (e.g. : exemple 6). Pour la dernière étape, consistant à typer les arguments, nous avons utilisé la casse des préposition, les étiquettes des tokens, ainsi que le prédicat et l'emplacement de l'argument par rapport à celui-ci. La table 3 nous montre différents exemples d'identification du type des arguments.

Un premier essai d'alignement des CONSTRUCTIONS nous a révélé que les informations contenues dans le *xLVF* n'étaient pas suffisantes. Il est nécessaire, entre autre, de savoir si le verbe est factitif et de connaitre le type des constructions pronominales (i.e. : subjectif, réfléchi, réciproque, passif) pour lever les ambigüités existantes. Ces informations apparaissant dans la version papier du *LVF*, nous avons entrepris de les extraire à partir d'une version PDF du *LVF*. Le PDF a tout d'abord été converti en HTML[3] pour avoir un format plus facilement analysable. Des expression régulières basées sur les balises HTML et sur différents mots-clés ont permis d'identifier les données utiles. Une analyse de surface a été effectuée pour donner du volume à cette suite d'éléments afin de générer un fichier XML (le *xoLVF*, voir figure 4) contenant les informations de la version papier du *LVF* dans un format univoque et structuré. En plus de ces informations, la description complète de la classification des verbes a été récupérée, ajoutant deux nouveaux niveaux de classification.

3. grâce à pdftohtml

```
<LVF>
  <ClasseGenerique nom="C" nombre="2039" desc="de communication">
    <ClasseSemantique nom="C1" nombre="1059" desc="exprimer par cri, parole, son">
      <Classe nom="C1a" nombre="232" desc="émettre un cri, humain ou animal">
        <SousClasse nom="1" desc="émettre le cri spécifique de l'espèce animale">
          <Const nom="A20" desc="intransitifs" intran="True">
            <Entree nom="aboyer 01" oper="(canis)f.cri espèce"
                    sens="émettre aboiement" phrase=" Le chien a~ ."
                    deriv="aboi,-">
                  ...
          </Const>
          ...
        </SousClasse>
        <SousClasse nom="2" desc="émettre une des diverses formes de parler ou
                                  d'écrit spécifiques à l'humain">
          <Const nom="A16" desc="intransitifs" intran="True">...</Const>
          <Const nom="P1006" desc="pronominaux" prono="subjectif">...</Const>
          ...
        </SousClasse>
        ...
      </Classe>
      ...
    </ClasseSemantique>
    ...
  </ClasseGenerique>
  ...
</LVF>
```

FIGURE 4 – Extrait du *xoLVF*, la version structurée de l'ouvrage *LVF* accessible sur notre wiki[1]

À l'aide des informations complémentaires extraites du *LVF*, l'alignement des CONSTRUCTIONS associées à une entrée a pu être effectué. Les contraintes syntaxico-sémantiques des constructions ont été utilisées afin d'identifier les arguments compatibles. Un ensemble de contraintes limitant les associations possibles entre les arguments des différentes constructions a été défini. Une première contrainte, gérant l'identité, permet d'associer des arguments ayant des codes identiques (e.g. : sujet humain et objet humain). Une autre contrainte permet d'aligner un argument pluriel avec deux arguments singuliers du même type (e.g. : sujet humain pluriel). Les informations extraites du *LVF* interdisent les associations entre certains emplacements (e.g. : le sujet d'un transitif et le sujet d'un pronominal passif), empêchent l'alignement de certains emplacements (e.g. : sujet des factitifs), et permettent de lier un élément à plusieurs du même type (e.g. : pronominaux réciproques et réfléchis). D'autres contraintes plus spécifiques ont été rajoutées après une analyse des premiers résultats, comme celle permettant de lier un argument de type humain pluriel avec un argument de type humain et un argument prépositionnel en « à ». Cet ensemble de contraintes ne permettant pas de lever toutes les ambigüités, l'alignement donne la priorité aux premiers arguments. Ainsi, pour les CONSTRUCTIONS T1100 et A10 (« applaudir »), le sujet de l'intransitif est lié au sujet du transitif (et pas à son objet direct).

La liaison des champs OPÉRATEUR et CONSTRUCTIONS peut être accomplie maintenant qu'ils ont été uniformisés et structurés. Pour cela, la redondance des informations contenue dans ces deux champs est utilisée. Les arguments des différentes CONSTRUCTIONS sont liées aux arguments de l'OPÉRATEUR en prenant soin de lier les arguments des CONSTRUCTIONS ayant été associés précédemment, au même argument de l'OPÉRATEUR. L'opération est similaire à l'étape de liaison des CONSTRUCTEURS, mais est plus complexe car l'OPÉRATEUR a un vocabulaire plus varié. Comme précédemment, des règles avec différentes priorités permettant de lier les éléments entre eux ont été utilisées. Des règles basées sur les informations syntaxiques permettent de lier les contraintes similaires, comme celles ayant des prépositions identiques. Des règles basées sur l'emplacement

Verbe	Sens	Const	Phrase
expirer 04	« faire sortir de l'air de soi »	A 1^0 0	« Pierre expire »
		T 1^0 3^1 0 0	« Pierre expire de l'air »
préoccuper 03	« être inquiet »	A 1^0 0	« Pierre est préoccupé »
		T 3^1 1^0 0 0	« L'avenir de son fils préoccupe Pierre »
accoucher 01	« enfanter »	N 1^0 b^1	« Marie accouche de Pierre »
		A 1^0 0	« Marie accouche »
		T† 1^2 1^0 0 8^3	« Le médecin accouche Marie »
libérer 04	« affranchir »	T 1^0 1^1 b^2 0	« On libère Pierre de l'emprise de sa mère »
		P‡ $1^{0,1}$ 0 b^2 0	« On se libère de l'influence de Pierre »

0 :rien, 1 :humain, 3 :chose, 8 :instrumental/moyen, b :préposition « de »
† : factitif, ‡ : pronominal réflechi

TABLE 4 – Exemples de CONSTRUCTIONS

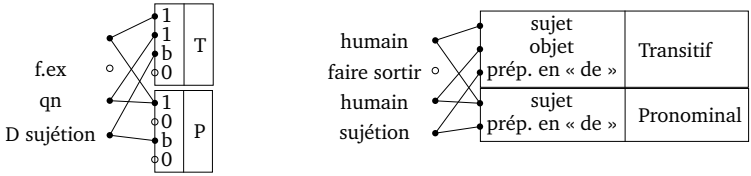

FIGURE 5 – Alignement et redistribution des champs OPÉRATEUR et CONSTRUCTIONS de *« libérer 04 »*

permettent d'associer le sujet de l'OPÉRATEUR avec le sujet de la première CONSTRUCTION. Enfin, le dernier type de règle est basé sur la sémantique et permet à des contraintes sémantiques similaires d'être liées. Si un argument d'une CONSTRUCTION ne peut être lié à aucun argument de l'OPÉRATEUR (e.g. : si l'OPÉRATEUR n'a pas de sujet), un argument factice est créé. La partie gauche de la figure 5 nous donne un exemple d'alignement, montrant les projections des constructions syntaxiques transitive et pronominale réfléchie de l'entrée *« libérer 04 »* sur son opérateur.

L'étape finale consiste à redistribuer les informations sémantiques et syntaxiques afin d'atteindre l'objectif défini dans la section 3. L'OPÉRATEUR est utilisé comme base pour le cadre sémantique et les CONSTRUCTIONS comme base pour les cadres syntaxiques. Les informations syntaxiques des contraintes de l'OPÉRATEUR ont été transférées au niveau des CONSTRUCTIONS et les contraintes sémantiques des CONSTRUCTIONS ont été transférées au niveau de l'OPÉRATEUR. Les arguments sémantiques de l'OPÉRATEUR ont été laissés tel quel. La partie droite de la figure 5 nous montre la redistribution et la transformation des codes pour l'entrée *« libérer 04 »*. Le résultat final correspond bien à l'objectif de la figure 3.

Pour la création du corpus d'exemples, nous avons utilisé les PHRASES associées aux différentes entrées. Le premier problème est que ces exemples peuvent représenter plusieurs phrases (e.g. : « On a∼des documents,des livres. », « amasser »). Nous avons donc dû décomposer ces exemples en phrases canonique à l'aide d'outils d'analyse de surface. La décomposition n'est pas toujours possible comme pour certains verbes avec des sujets pluriel et singulier (e.g. : *« Ses forces, la chance ont a∼ Pierre . »*, « abandonner »). Le second problème est que le verbe est représenté par son initiale suivie du symbole ∼, et donc un analyseur syntaxique normal ne pourra pas analyser

correctement ces exemples. Pour résoudre ce problème, nous avons entrainé un analyseur syntaxique (Bohnet *et al.*, 2010) sur une version modifiée du FrenchTreeBank(Abeillé *et al.*, 2000). Nous avons remplacé tous les verbes en tête de phrase par leur initiale suivie du symbole ∼. Nous n'avons pas remplacé tous les verbes car il existe des entrées avec des compléments syntagmatiques où le verbe est conjugué (i.e. : « On v∼qu'il soit heureux. », « vouloir »). L'autre intérêt d'avoir uniquement transformé les verbes en tête de phrase est que cela pousse l'analyseur syntaxique à prendre le verbe abrégé comme tête de la phrase. L'association des rôles sémantiques aux arguments syntaxiques a été effectuée à l'aide de règles de réécriture permettant d'identifier les types de constructions syntaxiques utilisées. Les informations sémantiques n'ont pas été nécessaires car nous connaissions le sens du verbe et les réalisations syntaxiques utilisées dans le *xLVF* sont assez limitées.

5 Évaluation

L'évaluation de la qualité des améliorations réalisées a été effectuée à l'aide de scripts permettant de vérifier la cohérence de la ressource obtenue, ainsi que par une analyse manuelle d'un échantillon représentatif.

La vérification automatisée de la cohérence de la ressource est importante car elle est peu couteuse et permet d'éviter nombre d'erreurs. Ce contrôle a été fait à l'aide de scripts vérifiant que :
– un OPÉRATEUR a un sujet, un prédicat et un certain nombre d'arguments complémentaires,
– le *xoLVF* contient le bon nombre d'entrées pour chaque classe et sous-classe,
– les arguments du CONSTRUCTEUR sont tous liés à ceux de l'OPÉRATEUR
Pour l'évaluation manuelle, un échantillon représentatif de 100 entrées a été extrait puis examiné afin de vérifier si le résultat obtenu était celui souhaité. Chaque entrée a été contrôlée sur la correction des abréviations, la structuration de l'OPÉRATEUR, la liaison des CONSTRUCTEURS, et la liaison de l'OPÉRATEUR avec les CONSTRUCTEURS. Sur les 100 entrées analysées 84 sont bonnes, 13 ont des erreurs dues à notre analyses et 3 ont des erreurs dues au *xLVF*. Parmi les erreurs de nos analyses, il y en a 3 de structuration de l'OPÉRATEUR, 1 de liaison des CONSTRUCTEURS et 9 de liaison OPÉRATEUR-CONSTRUCTEUR. Les erreurs sont dues à certaines entrées manquantes dans les lexiques. Par exemple, *org mvs* (i.e. : un mauvais organe) n'est pas regroupé en un seul terme lors de la structuration de l'OPÉRATEUR, et les contraintes du CONTRUCTEUR sur le domaine des *choses* ne sont pas liées avec les arguments *coup* et *viande* des OPÉRATEURS (car ces deux mots ne sont pas considérés comme des choses). La plupart des erreurs sont donc facilement corrigibles en ajoutant des entrées aux lexiques. Les erreurs dues au *xLVF* sont dues à certaines prépositions qui ne sont pas mises en majuscules et qui sont donc considérés comme des arguments uniquement sémantiques, comme pour l'entrée *« ressaisir 01 »* où l'argument P SOM de l'OPÉRATEUR GRP+RE QN P SOM () devrait être associé au circonstant de manière du CONSTRUCTEUR P1108.

Nous avons analysé 100 phrases annotées afin de vérifier si les différentes étapes de l'annotation se sont bien déroulées. Cette analyse indique que 70 phrases ont été bien annotées, et possèdent les bons rôles sémantiques. Les problèmes rencontrés dans les 30 autres phrases sont de natures diverses. Le problème le plus apparent est le mauvais filtrage des structures en dépendances (15 phrases). En effet, les tournures ne correspondant pas à celles que nous avions identifié n'ont pas été étiquetées. Cela est toutefois facilement corrigible en intégrant ces nouvelles tournures à

nos règles de filtrage. Le second problème rencontré est la mauvaise analyse des exemples par l'analyseur syntaxique (6 phrases). Par exemple, il arrive que des groupes prépositionnels soient rattachés à un argument plutôt qu'au verbe. Un autre problème provient de la mauvaise liaison entre les constructions et les opérateurs. Nous nous retrouvons donc à associer les arguments syntaxiques aux mauvais arguments sémantiques. Le dernier problème est la mauvaise séparation des différents exemples (3 phrases). Il arrive ainsi d'avoir une phrase avec plusieurs occurrences du verbe ou d'un de ses arguments. Les résultats de cette analyse sont donc positifs étant donné que la plupart des phrases ont été bien annotées et qu'une grande parties des erreurs est corrigible.

6 Conclusion

Nous avons présenté des améliorations du *xLVF* qui donnent à cette ressource une nouvelle dimension. Les informations son structurées, moins ambigües et plus uniformes, permettant ainsi l'utilisation du *xLVF* pour faire de l'étiquetage de rôles sémantiques. Ces améliorations vont aussi permettre de nouvelles améliorations du *xLVF* comme identifier quelles sont les entrées d'un verbe qui sont SYNONYMES d'une entrée (pour le moment les SYNONYMES sont des verbes et non des entrées). Ainsi, on sait que l'entrée *« humilier 01 »* a pour SYNONYME le verbe *« abaisser »*, mais on ne sait pas si c'est l'entrée *« abaisser 01 »* (« On abaisse le rideau. ») ou l'entrée *« abaisser 06 »* (« On abaisse Pierre. ») qui est SYNONYME. Cette nouvelle version pourra aussi être utilisée pour effectuer des recherches plus précises sur le *xLVF*, permettant par exemple à des linguistes d'identifier des verbes ayant certaines caractéristiques syntaxiques et sémantiques. Les deux fichiers XML obtenus sont en libre accès sur notre wiki `http://margaux.philosophie.uni-stuttgart.de/lvf/`. Nous pensons prochainement associer cette ressource avec *Disco* et *Wolf* pour annoter le French TreeBank.

Références

ABEILLÉ, A., CLÉMENT, L. et KINYON, A. (2000). Building a treebank for french. *In In Proceedings of the LREC 2000*.

BAKER, C. F., FILLMORE, C. J. et LOWE, J. B. (1998). The berkeley framenet project. *In Proceedings of the COLING-ACL*, Montreal, Canada.

BOHNET, B., WANNER, L., MILLE, S. et BURGA, A. (2010). Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. *In Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 98–106, Stroudsburg, PA, USA. Association for Computational Linguistics.

DUBOIS, J. et DUBOIS-CHARLIER, F. (1997). *Les verbes français*. Larousse-Bordas.

FELLBAUM, C., éditeur (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London.

FRANÇOIS, J., PESANT, D. et LEEMAN, D. (2007). *Le classement syntactico-sémantique des verbes français*. Langue française. Larousse.

HADOUCHE, F. et LAPALME, G. (2010). Une version électronique du LVF comparée avec d'autres ressources lexicales. *Langages*, 10(179-180):193–220. Mise en page différente que celle parue dans la revue.

KINGSBURY, P. et PALMER, M. (2003). Propbank : The next level of treebank. *In Proceedings of Treebanks and Lexical Theories*, Växjö, Sweden.

KOLB, P. (May 2009). Experiments on the difference between semantic similarity and relatedness. *In Proceedings of the 17th Nordic Conference on Computational Linguistics - NODALIDA '09*, Odense, Denmark.

LEVIN, B. (1993). *English verb classes and alternations : a preliminary investigation*. University of Chicago Press.

MERTENS, P. (2010). Restrictions de sélection et réalisations syntagmatiques dans dicovalence. *In Actes TALN 2010*, Montreal, Canada.

ROMARY, L., SALMON-ALT, S. et FRANCOPOULO, G. (2004). Standards going concrete : from LMF to Morphalou. *In Workshop Enhancing and Using Electronic Dictionaries*, page 7 p, Geneva, Switzerland. none. Colloque avec actes et comité de lecture. internationale.

SAGOT, B. et FIŠER, D. (2008). Building a free French wordnet from multilingual resources. *In OntoLex*, Marrakech, Morocco.

SCHULER, K. K. (2005). *Verbnet : a broad-coverage, comprehensive verb lexicon*. Thèse de doctorat, , Philadelphia, PA, USA. AAI3179808.