# Exploitation des tables du Lexique-Grammaire pour l'analyse syntaxique automatique

Benoît Sagot[1] & Elsa Tolone[2]

1. Alpage, INRIA Paris-Rocquencourt & Université Paris 7 (France)
2. IGM, Université Paris-Est (France)

Colloque Lexique Grammaire – Bergen, Norvège
3 octobre 2009

# Contexte

- Les tables du Lexique-Grammaire sont une source très riche d'informations lexicales
- Ces informations devraient être très utiles pour l'analyse syntaxique automatique
- Mais les tables ne sont pas exploitables telles quelles dans un analyseur syntaxique
  - les propriétés définitoires ne sont pas représentées
  - le format de représentation des informations lexicales n'est pas formalisé
  - il faut interfacer ces informations avec un analyseur syntaxique réel
- Nous nous sommes restreint pour l'instant aux tables de verbes simples

# Objectifs

Triple objectif :

- convertir les tables du Lexique-Grammaire en un format TAL
- coupler le lexique syntaxique obtenu, nommé $lglex_{\text{Lefff}}$, avec un analyseur syntaxique
- évaluer l'analyseur obtenu

Outils TAL retenus :

- analyseur syntaxique : FRMG [Thomasset et de La Clergerie 2005]
- formalisme lexical : Alexina, format du lexique Lefff [Sagot *et al.* 2006] utilisé par FRMG

→ ceci permet une comparaison entre $\text{FRMG}_{\text{Lefff}}$ et $\text{FRMG}_{lglex}$

# 1. Le lexique verbal *lglex*

# Tables des classes

Propriétés définitoires = pas représentées dans les tables
→ à décrire dans des **tables des classes** (1 par catégorie) :

- colonnes = toutes les propriétés syntaxiques répertoriées pour la catégorie concernée (après normalisation)
- lignes = classes définies pour cette catégorie
- intersection ligne/colonne :
  - $+$ (resp. $-$) = la propriété correspondante est vérifiée (resp. non vérifiée) par tous les éléments de la classe
  - $o$ = propriété explicitement codée dans la table concernée
  - ? = non encore renseigné

En cours de réalisation à l'IGM, presque terminé pour les verbes simples [Constant & Tolone 2008]

# Tables des classes : un exemple

| table | N0 =: Nhum | N0 =: N-hum | N0 =: Nnr | N0 =: V1-inf W | <ENT> | Ppv =: se figé | N0 V | N0 V N1 | zone 1 | N0 V à N1 | N1 =: Nhum | N1 =: N-hum | N0 V Prep N1 V0-inf W | N0 V N1 V0-inf W | N0 V V0-inf W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V_2 | + | - | - | - | o | o | - | - | - | - | - | + | o | o | + |
| V_4 | - | - | + | + | o | - | o | + | - | - | o | o | - | - | - |
| V_31R | o | o | - | - | o | o | + | - | - | - | - | - | - | - | - |
| V_31H | + | - | - | - | o | o | + | - | - | - | - | - | - | - | - |
| V_33 | o | o | o | - | o | o | o | - | - | + | o | o | - | - | - |
| V_32H | o | - | o | - | o | o | - | + | - | - | + | - | - | - | - |

# *lglex*

Table des classes des verbes → possibilité d'extraire un **lexique syntaxique** des verbes simples à partir des tables [Constant & Tolone 2008] :

- format textuel ou XML
- nommé lexique **lglex**
- conversion depuis les tables Excel grâce à l'outil *LGExtract*

*lglex* est le point de départ du processus de conversion vers le format Alexina

# *lglex* : un exemple

*ID=V_35L_242*
*lexical-info=[locs=(loc=[id="1",list=()],loc=[id="2",list=()]),cat="verb",verb=[***lemma="ruisseler"***],*
*aux-list=(),prepositions=()]*
*args=(*
*const=[dist=(comp=[cat="NP",source="true",introd-prep=(),origine=(orig="Loc N1 =: de N1 source"),*
*introd-loc=(prep="de")]),pos="1"],*
*const=[dist=(comp=[***cat="NP"***,introd-prep=(),origine=(orig="Loc N2 =: vers N2 destination",*
*orig="Loc N2 =: dans N2 destination"),***introd-loc=(prep="vers",prep="dans")***,destination="true"]),***pos="2"***],*
*const=[pos="0",dist=(comp=[cat="NP",introd-prep=(),nothum="true",origine=(orig="N0 =: N-hum"),*
*introd-loc=()])])*
*all-constructions=[absolute=(***construction="o::N0 V Loc N1 source Loc N2 destination"***,construction="o::N0 V",*
*construction="o::N0 être V-ant",construction="true::N0 V Loc N1"),*
*relative=(construction="Ppv =: y",construction="Ppv =: en",construction="[extrap]")]*
*example=[example="L'eau ruisselle de la gouttière sur les passants"]*

# 2. Le Lefff et le format Alexina

# Le Lefff

- Le Lefff (Lexique des Formes Fléchies du Français) est un lexique morphologique et syntaxique pour le français
  - à large couverture
  - librement distribué
- Il repose sur l'architecture **Alexina** d'acquisition et de modélisation de lexiques morphologiques et syntaxiques.

# Alexina

Architecture à deux niveaux

- Le lexique **intensionnel**
  - associe à chaque entrée (emploi d'un lemme) un cadre de sous-catégorisation canonique
  - liste les redistributions possibles à partir de ce cadre
- Le processus de **compilation** du lexique intensionnel en lexique **extensionnel** construit différentes entrées pour chaque forme fléchie du lemme et chaque redistribution possible.

# Alexina sur un exemple

- Exemple d'entrée intentionnelle :

  *clarifier*$_1$ *v-er:std*
  *Lemma;v;*
  *<***Suj***:cln|scompl|sinf|sn,***Obj***:(cla|scompl|sn)>;*
  *%ppp_employé_comme_adj,%actif,%se_moyen_impersonnel,*
  *%passif_impersonnel,%passif*

# Fonctions, réalisations et redistributions

- **Fonctions syntaxiques** (cf. Dicovalence) : Suj, Obj, Objà, Objde, Loc, Dloc, Att, Obl/Obl2
- **Réalisations** : directes (sn, sa, sinf, scompl, qcompl) ; clitiques (cln, cla, cld, y, en) ; prépositionnelles (prep+directe, p.ex. par-sn, à-sinf, de-scompl)
- **Redistributions** : %actif, %passif, %se_neutre, %actif_impersonnel...

# 3. Conversion de *lglex* en un lexique au format Alexina

# Différents types de constructions

Chaque entrée de *lglex* est associée à des constructions dont on peut distinguer plusieurs types :

1. la construction **de base**, définitoire de la classe de l'entrée
2. les constructions **de base étendues**, obtenues par adjonction d'arguments à la construction de base
   - en pratique, ces constructions sont toutes des intermédiaires entre la construction de base et une construction dite **de base maximalement étendue**
3. les **variantes** de la construction de base, obtenues par effacement d'un ou de plusieurs arguments ou par changement de type de réalisation
4. les constructions qui sont des **redistributions**
5. les constructions dont il semble qu'elles auraient dû conduire à des entrées distinctes, dites **entrées secondaires**

# Identification des entrées à construire

- Nous avons développé une méthode permettant d'*aligner* deux constructions
- $\rightarrow$ identification de la construction de base maximalement étendue et de ses variantes
  - ces constructions sont rassemblées en une seule entrée dite **entrée canonique**
- Autres constructions :
  - certaines correspondent à des **redistributions standard** ([passif par], [extrap]...) $\rightarrow$ ajout à l'entrée canonique de cette redistribution
  - les autres induisent la création d'**entrées distinctes**

# Construction d'une entrée

Pour chaque entrée à construire :

- on extrait la **sous-catégorisation du cadre maximal**
  - les **fonctions syntaxiques** sont calculées à l'aide d'heuristiques utilisant la position, la nature et certaines propriétés des arguments
  - les **réalisations** sont obtenues soit directement, soit par l'intermédiaire des champs *prep* et *loc* lorsque la construction mentionne un argument en *Prép X*, soit grâce à des propriétés spécifiques (par exemple, *à* $N_1 = Ppv = : le$)
- on prend en compte les différentes **variantes** du cadre maximal en rendant **facultatifs** les arguments appropriés
- on ajoute des informations complémentaires lorsqu'on peut les extraire de propriétés
  - informations de contrôle, de mode pour les complétives, etc.

# Résultat de la conversion sur l'exemple précédent

$ruisseler^{35\mathrm{L}}_{242}$ *v-er:std*
*100;Lemma;v;*
*<***Suj***:cln|sn,***Dloc***:(de-sn|en),***Loc***:(vers-sn|dans-sn|y)>;*
*cat=v;*
*%actif*

# Lexique obtenu : $lglex_{\text{Lefff}}$

Le lexique verbal obtenu, $lglex_{\text{Lefff}}$, contient 16 903 entrées pour 5 694 lemmes verbaux différents (2,96 entrées/lemme).

- À titre de comparaison, le Lefff contient seulement 7 072 entrées verbales pour 6 818 lemmes verbaux distincts (1,04 entrées/lemme)

Au niveau extensionnel, le Lefff contient 361 268 entrées, alors que le lexique extrait de *lglex* en contient 763 555.

# 4. Intégration dans l'analyseur syntaxique FRMG

# Intégration dans l'analyseur syntaxique FRMG

- L'intégration de *lglex*$_{\text{Lefff}}$ dans l'analyseur FRMG est rapide
- Le *lexeur* de l'analyseur fait normalement appel à une base de données lexicales construite à partir du Lefff
- Nous allons donc construire un autre lexique, qui repose sur *lglex*$_{\text{Lefff}}$, que nous demanderons au lexeur d'utiliser à la place du Lefff

# Intégration dans l'analyseur syntaxique FRMG

Il faut donc :

- remplacer les entrées verbales du Lefff par $lglex_{\text{Lefff}}$
- conserver les autres entrées du Lefff
- compléter le résultat par diverses entrées verbales venant du Lefff, qui ne font pas partie du lexique *lglex*
  - entrées pour les auxiliaires et semi-auxiliaires
  - certains verbes à montée
  - les verbes impersonnels
  - les entrées pour les têtes syntaxiques des constructions à verbes support
- construire la base de données lexicales correspondantes
- spécifier à FRMG d'utiliser cette dernière

Le résultat est une variante de l'analyseur FRMG, que nous noterons $\text{FRMG}_{lglex}$, par opposition à la variante standard notée $\text{FRMG}_{\text{Lefff}}$.

# 5. Évaluation et discussion

# Protocole utilisé

- Nous avons évalué $\text{FRMG}_{\text{Lefff}}$ et $\text{FRMG}_{lglex}$ en analysant la partie annotée manuellement du corpus EASy [Paroubek *et al.* 2005]
  - 4 306 phrases de styles variés (journalistique, médical, oral, questions, littéraire...)
- métriques utilisées : celles de la première campagne EASy d'évaluation des analyseurs syntaxiques, qui a eu lieu fin 2005 [Paroubek *et al.* 2006]
  - évaluation en *chunks* et en *relations* ($\sim$ dépendances entre mots pleins)

# Précautions

Les résultats de FRMG$_{lglex}$ seront à interpréter en gardant à l'esprit plusieurs points

- FRMG utilise les entrées verbales converties à partir des tables, et non pas les entrées telles qu'elles sont dans les tables
  $\longrightarrow$ le processus de conversion est sûrement entâché d'erreurs
- le Lefff a été développé en parallèle aux campagnes EASy, contrairement aux tables
- *lglex*$_{\text{Lefff}}$ ne contient pas toutes les entrées verbales nécessaires, et il a fallu en rajouter
  $\longrightarrow$ il se peut que certaines entrées manquent encore

# Résultats

Résultats comparatifs EASy de $\text{FRMG}_{\text{Lefff}}$ et $\text{FRMG}_{\textit{lglex}}$ (pourcentages de f-mesure) :

| | Chunks | | Relations | |
|---|---|---|---|---|
| Sous-corpus | $\text{FRMG}_{\text{Lefff}}$ | $\text{FRMG}_{\textit{lglex}}$ | $\text{FRMG}_{\text{Lefff}}$ | $\text{FRMG}_{\textit{lglex}}$ |
| `general_lemonde` | **86.8%** | 82.8% | **59.8%** | 56.9% |
| `general_senat` | 82.7% | **83.1%** | **56.7%** | 54.9% |
| `litteraire_2` | **84.7%** | 81.5% | **59.2%** | 56.3% |
| `medical_2` | 85.4% | **89.2%** | **62.4%** | 58.6% |
| `oral_delic_8` | **74.1%** | 73.6% | 47.2% | **48.5%** |
| `questions_amaryllis` | 90.5% | **90.6%** | **65.6%** | 63.2% |
| *total* | **84.4%** | *82.3%* | **59.9%** | *56.6%* |

Temps d'analyse plus élevés avec $\text{FRMG}_{\textit{lglex}}$ qu'avec $\text{FRMG}_{\text{Lefff}}$ : temps médian par phrase de 0,62 s contre 0,26 s

- ceci provient du nombre d'entrées par lemme 3 plus élevé dans *lglex* que dans le Lefff

- FRMG$_{lglex}$ donne de meilleurs résultats que FRMG$_{Lefff}$ pour certaines relations
  - relations « classiques » MOD-A et MOD-R
  - relations « difficiles » MOD-P et APP
- la relation ATB-SO (attribut du sujet ou de l'objet) est celle pour lequel la différence en rappel est la plus importante (34,0% contre 58,4%) ;

- l'ambiguïté lexicale plus élevée dans FRMG$_{lglex}$ induit
  - une ambiguïté plus élevée dans l'analyseur
  - et donc d'autant plus de risque de se tromper au moment de la désambiguïsation
- exemple :
  - *[...] on estime que cette décision [ferait] dérailler le processus de paix*
  - FRMG utilise l'heuristique habituelle « on préfère les arguments aux modifieurs »
  - FRMG$_{lglex}$ fait de *de paix* un argument de *estimer* (*estimer qqch de qqn*)
  - FRMG$_{Lefff}$ ne se trompe pas car dans le *Lefff*, *estimer* n'a pas d'Objde

# Conclusions et perspectives

# À court terme

- De nombreuses phrases reçoivent une analyse complète par $\text{FRMG}_{lglex}$ mais pas par $\text{FRMG}_{\text{Lefff}}$, et inversement
  - par exemple, sur le sous-corpus `general_lemonde`, 177 phrases sont entièrement reconnues par les deux analyseurs, 85 seulement par $\text{FRMG}_{\text{Lefff}}$, 76 seulement par $\text{FRMG}_{lglex}$, et 111 par aucun des deux
- Les analyses complètes sont souvent bien meilleures que les analyses partielles
- Possibilité d'utiliser les deux analyseurs en même temps, et de ne conserver le résultat que de celui qui a donné une analyse complète

# À long terme (1/2)

Exploiter la complémentarité des deux ressources pour y détecter des erreurs et les améliorer mutuellement

- étudier les différences entre les erreurs faites par chacun des deux analyseurs
- utiliser des techniques automatiques de fouille d'erreurs pour extraire automatiquement des entrées probablement erronées [Sagot et de La Clergerie 2008]
  - pour $lglex_{\text{Lefff}}$, de nombreuses erreurs que l'on trouverait viendraient probablement du processus de conversion
  - certaines erreurs détectées pourraient provenir d'erreurs dans les tables, que l'on pourrait ainsi corriger

# À long terme (2/2)

L'objectif reste néamoins d'exploiter au mieux les informations lexicales des tables du Lexique-Grammaire pour construire un analyseur syntaxique qui soit aussi bon que possible

- améliorer le processus de conversion
- prendre en compte, au fur et à mesure de la construction des tables des classes, les autres catégories