Chaîne de traitements linguistiques pour l'acquisition automatique de lexique

Analyse syntaxique avec des RCG

Lionel Clément

Projet Atoll

INRIA Rocquencourt

Lionel.Clement@inria.fr

Journée RLT – le 14 juin 2002

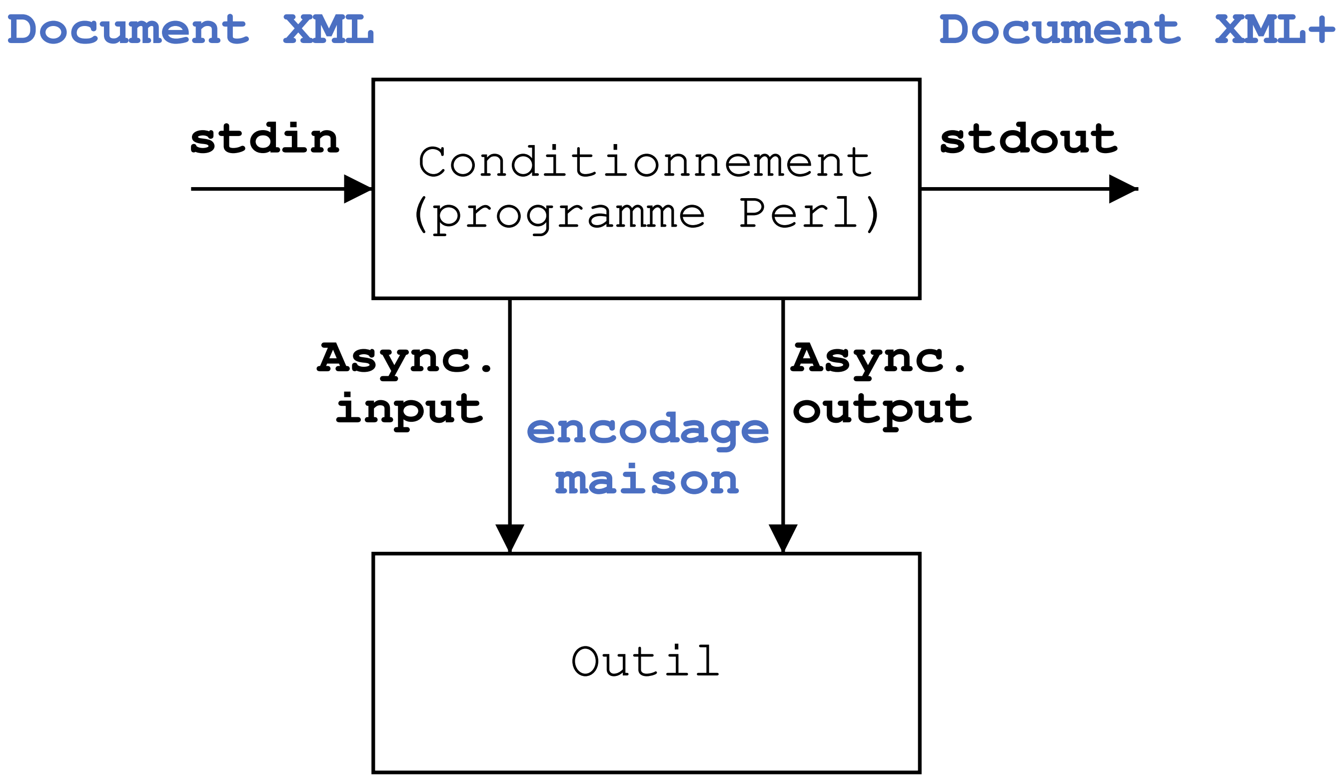
Chaîne de traitement

- Segmentation du texte en mots
- Étiquetage morphosyntaxique
- Lemmatisation
- Analyse syntaxique

RCG

- Définition
- RCG & TAG
- RCG & Syntaxe
- Proposition : ARCG

Chaîne de traitement



Segmentation en mots

- Segmenteur (**Tokenizer**) développé en **C++ / Flex**
  - Éclate les mots amalgamés (au → à le.)
  - Segmente des suites de lettres en fonction d'expressions régulières (nombres, acronymes, abréviations, mots, etc.)
  - Regroupe les mots composés et ajoute les informations morphosyntaxiques sur les mots en fonction d'un lexique (interface avec **Lexed**.)
  - Construit un DAG encodé en XML

Segmentation en mots : outils & ressources

- Outils développés réutilisables pour d'autres applications : **Lexed** & **Tokenizer**
  - Portables (vers d'autres langues, d'autres systèmes Unix)
  - Configurables
- Lexiques : aucun n'est distribué sauf ABU !
  - Multext
  - ABU
  - DELAC : Usage d'Intex (certains mots composés)
  - Lexique Paris 7 (surtout pour les mots composés)
  - Lexique Perso (verbes, noms, adjectifs, adverbes, nombres)

Segmentation – DTD

- Utilisation d'une DTD XML pour représenter des DAG
- Impératifs :
  1. Le seul contenu des balises est le texte source (PCDATA)
  2. Possibilité d'exprimer des ambiguïtés (mots simples / mots composés, homonymes)
- DTD Patrice Lopez (fsm,xml 0.1)

Segmentation – DTD

```
<fsm type="fsa">
<transition source="fsm0s0" target="fsm0s1">
<label type="in">
<token tag="D-ind-ms" form="Un" id="fsm0w0">Un</token>
<w lemma="un" tag="D a - m s - - i -" entry="un" idref="fsm0w0"/>
<w lemma="un" tag="M c m s -" entry="un" idref="fsm0w0"/>
<w lemma="un" tag="P i - m s - -" entry="un" idref="fsm0w0"/>
</label>
</transition>
<transition source="fsm0s1" target="fsm0s4">
<label type="in">
<token tag="N-C-ms" form="fer" id="fsm0w1">fer</token>
<token tag="P--" form="à" id="fsm0w2">à</token>
<token tag="N-C-ms" form="cheval" id="fsm0w3">cheval</token>
<w lemma="fer à cheval" tag="N c m s - -" entry="fer à cheval" idref="fsm0w1 fsm0w2 fsm0w3"/>
</label>
</transition>
<transition source="fsm0s1" target="fsm0s2">
<label type="in">
<w lemma="fer" tag="N c m s - -" entry="fer" idref="fsm0w1"/>
```

```
</label>
</transition>
<transition source="fsm0s2" target="fsm0s4">
<label type="in">
<w lemma="à cheval" tag="R - -" entry="à cheval" idref="fsm0w2 fsm0w3"/>
</label>
</transition>
<transition source="fsm0s2" target="fsm0s3">
<label type="in">
<w lemma="à" tag="S p a" entry="à" idref="fsm0w2"/>
</label>
</transition>
<transition source="fsm0s3" target="fsm0s4">
<label type="in">
<w lemma="cheval" tag="N c m s - -" entry="cheval" idref="fsm0w3"/>
</label>
</transition>
</fsm>
```

Étiquetage

- TreeTagger Réentraîné pour le français
  - Nouveau corpus de référence le Monde
  - Conventions différentes
  - Étiquettes riches
  - Corpus d'entraînement : 749 860 mots – 208 étiquettes
  - Corpus d'évaluation : 118 618 mots – 96,75 % précision

Lemmatisation

- **Lexed** + Lexique dico morpho
- **Flemm** (Fiametta Namer)
  - Utile seulement pour mots inconnus
  - Mais :
    - Conversion d'étiquettes
    - Portage du code (encodage des données en dur)

Parsing

- RCG + filtrage
  - TAG → RCG (Pierre Boullier 98 in TAG+4)
- Analyse TAG avec DyALog (Éric de la Clergerie)
- Filtrage :
  - Euristiques sous DyALog
  - Automate déterministe pour éliminer des familles

Range Concatenation Grammars

- Grammaires de concaténation d'intervalles (Pierre Boullier 98)
- Définit l'exacte sous-ensemble des grammaires sensibles au contexte analysables en temps polynomial.
- Construit des relations entre des occurrences de sous-chaînes et des "Prédicats".

Range Concatenation Grammars

- Une PRCG est une grammaire $G = (N, T, V, P, S)$

  P ensemble de "clauses"

  $$\psi_0 \rightarrow \psi_1 \dots \psi_m$$

  ψ_i prédicat (ou "non-terminal")

  $$A(\alpha_1, \dots, \alpha_p)$$

- Un intervalle $< i..j >$ pour une chaîne $a_1 \dots a_n$ dénote l'occurrence de la sous-chaîne $a_{i+1} \dots a_j$.
  $< i..i >$ dénote ϵ, la chaîne vide (l'ensemble des occurrences).

- Une clause instantiée, est une clause
  $A_0(\phi(\alpha_{01}), \dots, \phi(\alpha_{0p_0})) \rightarrow A_1(\phi(\alpha_{11}), \dots, \phi(\alpha_{1p_1})) \dots A_k(\phi(\alpha_{k1}), \dots, \phi(\alpha_{kp_k}))$
  où $\phi(x)$ représente une substitution de variables.

RCG

Exemple :
$A(< g..h >, < i..j >, < k..l >) \to B(< g_{+1}..h >, < i_{+1}..j_{-1} >, < k..l_{-1} >)$ est une instantiation de
$A(aX, bYc, Zd) \to B(X, Y, Z)$ relativement à la chaîne $\alpha_1 \ldots \alpha_n$ ssi
$\alpha_{g+1} = a, \alpha_{i+1} = b, \alpha_j = c$ *et* $\alpha_l = d$

- Langage engendré par une RCG
- Soit une chaîne $w = \alpha_1 \ldots \alpha_n$
  $\underset{w}{\Rightarrow}$: relation de dérivation sur des chaînes de prédicats instanciés
  $\phi(P_0) \underset{w}{\Rightarrow} \phi(P_1) \ldots \phi(P_n)$ est vérifiée ssi $\exists$ une clause $P_0 \to P_1 \ldots P_n$
- $L(G) = \{w = \alpha_1 \ldots \alpha_n \mid S(< 0..n >) \overset{+}{\underset{w}{\Rightarrow}} \epsilon\}$

RCG – exemple

- $\{www \mid w \in \{a,b\}^*\}$

$$
\begin{array}{llcl}
1) & S(XYZ) & \rightarrow & A(X,Y,Z) \\
2) & A(aX,aY,aZ) & \rightarrow & A(X,Y,Z) \\
3) & A(bX,bY,bZ) & \rightarrow & A(X,Y,Z) \\
4) & A(\epsilon,\epsilon,\epsilon) & \rightarrow & \epsilon
\end{array}
$$

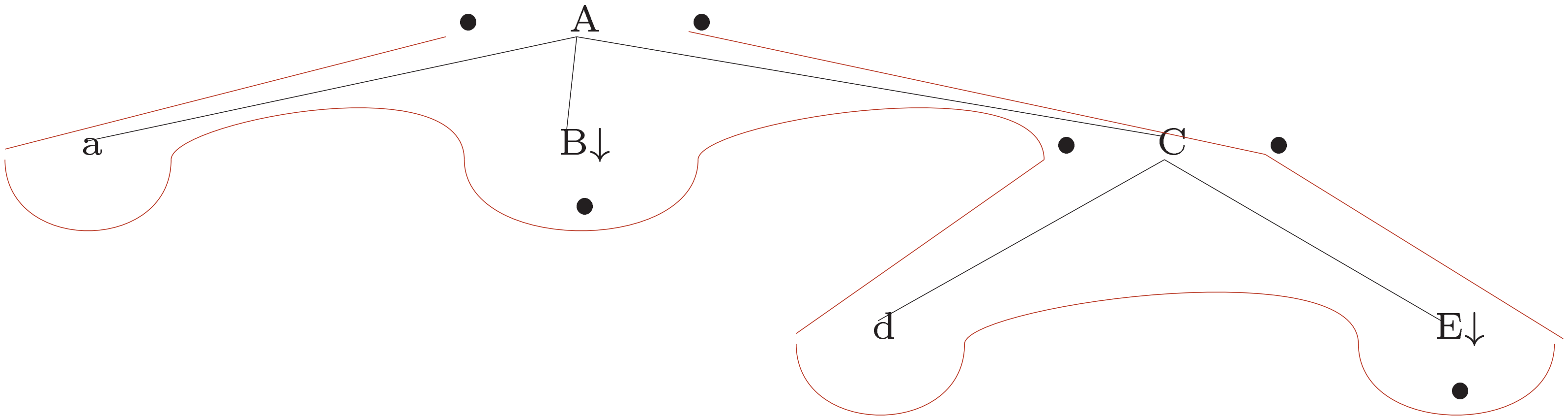
- Soit la chaîne $ababab$ ($a_1b_2a_3b_4a_5b_6$)

  $S(<0..2><2..4><4..6>) \rightarrow A(<0..2>,<2..4>,<4..6>)$ est une instantiation possible de 1
  $A(a<1..2>,a<3..4>,a<5..6>) \rightarrow A(<1..2>,<3..4>,<5..6>)$ est une instantiation possible de 2
  $A(b<2..2>,b<4..4>,b(6..6>) \rightarrow A(<2..2>,<4..4>,<6..6>)$ est une instantiation possible de 3
  $A(<2..2>,<4..4>,<6..6>) \rightarrow \epsilon$ est une instantiation possible de 4

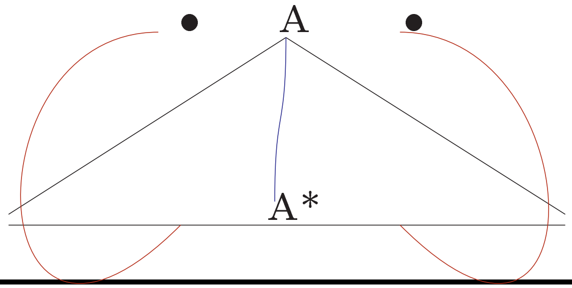
  $S(<0..2><2..4><4..6>) \underset{w}{\Rightarrow} A(<0..2>,<2..4>,<4..6>) \underset{w}{\Rightarrow} A(a<1..2>$
  $,a<3..4>,a<5..6>) \underset{w}{\Rightarrow} A(b<2..2>,b<4..4>,b(6..6>) \underset{w}{\Rightarrow} \epsilon$

RCG & TAG

- Arbre initial : Clause $A(\sigma) \rightarrow \psi_1 \ldots \psi_k$
  $A_1(LA\ a\ SB\ LC\ d\ SE\ RC\ RA) \rightarrow A_2(LA, RA)\ B_1(SB)\ E_1(SE)\ C_2(LC, RC)$



- Arbre auxiliaire : Clause $A(\sigma_1, \sigma_2) \rightarrow \psi_1 \ldots \psi_k$



RCG & Syntaxe

- Une RCG manipule des chaînes de "prédicats".
  $A(\alpha_{11}\alpha_{12}\ldots\alpha_{1k_1}, \alpha_{21}\alpha_{22}\ldots\alpha_{2k_2}, \ldots\alpha_{n1}\alpha_{n2}\ldots\alpha_{nk_n})$
  - Les positions syntaxiques sont des ensembles d'intervalles (successifs ou non) α_{ij}.
  - Les places syntaxiques sont des intervalles α_{ij}.
  - Les catégories syntaxiques sont les "prédicats".

  i.e. Lien avec TAG:
  - Subst : Un modifieur qui occupe une position syntaxique $\{\alpha\}$ est :
    - ∗ un prédicat P tel que $P(\alpha) \overset{+}{\underset{w}{\Rightarrow}} \epsilon$
  - Adj : Un modifieur qui occupe une position syntaxique $\{\alpha_1, \alpha_2\}$ est :
    - ∗ un prédicat P tel que $P(\alpha_1, \alpha_2) \overset{+}{\underset{w}{\Rightarrow}} \epsilon$