La chaîne d'acquisition de lexique

Lionel Clément

Atoll

INRIA Rocquencourt

Lionel.Clement@inria.fr

ARC RLT – le jeudi 12 décembre 2002

La chaîne d'acquisition de lexique

- Analyse "robuste" : du texte à la forêt d'analyses sur des documents volumineux bruts
- Outils et ressources réutilisables
- Expérience *jouet* méritant d'être validée
- Émergence de programmes de recherche

Les éléments de la chaîne

- Outils développés ou réutilisés
  - Lexed
  - Tokenizer
  - Wrappers
  - Spak
  - Flemm
  - TreeTagger réentraîné
  - Parser RCG
  - Parser DyaLog
  - Serveur de grammaires (en cours)
  - Outil de validation (en cours)

- Ressources produites ou réutilisées
  - Corpus (en cours)
  - Lexiques (en cours)
  - Grammaires (en cours)
- Outils de développement de ressources
  - Editeur de MG
  - Compilateur de MG
  - Parser Patrice Lopez (Azim)
  - Extraction de morphologie sur gros corpus (avec Bernard Lang)

Les éléments de la chaîne

- État des expériences actuelles
  - Chaîne relativement portable (seulement sur système unix) et utilisable pour d'autres applications
  - Éléments de la chaîne réutilisables
  - Mémoire de compétence
  - Résultats faibles en quantité mais fonctionne sur grammaire jouet
  - Manque validation
- Expérience en *vrai grandeur*
  - Tous les éléments de la chaîne sont robustes pour l'analyse de gros corpus (pipeline synchronisé, traitements optimisés)

- Manque
  - Grammaire couvrante
  - Lien avec chunking
  - Ressources robustes (développement de lexique)

- Expériences très fructueuses pour la conduite de nouvelles recherches (organisation d'une grammaire, grammaire du français, autres formalismes, etc.)
- Présentation de deux éléments de la chaîne : Lexed et Tokenizer

L'outil Lexed

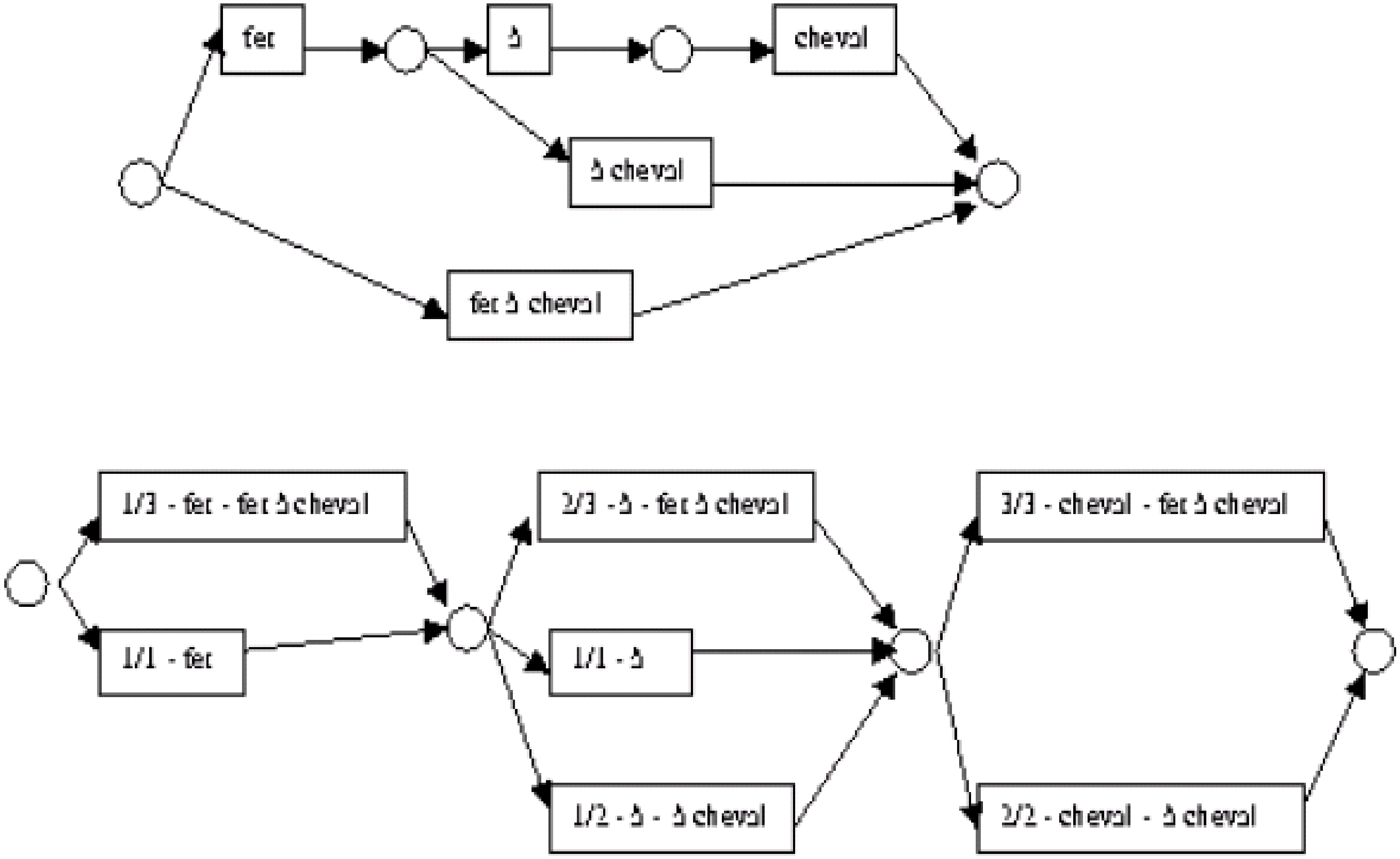
- Implémentation d'un automate pour des lexiques
- Commencé au CILSH puis à Talana. Repris à l'Inria
- Usages actuels
  - recherche l'entrée d'un lexique à partir d'une vedette (mots composés, lemmes, catégories, etc.)
  - construit des fichiers inverses (technique d'indexation de documents pour de nombreuses clefs)
  - recherche des mots à une approximation près (correction graphique de documents OCRisés)

- Limites
  - pas de gestion d'automates
  - pas de recherche sur un ensemble de vedettes
- Atouts
  - simplicité d'usage
  - rapidité
  - portabilité
- Faiblesses
  - encore peu testé et éprouvé
  - format maison (pas XML)
  - pas d'API proprement dit

- Distribution
  - Open source C++, autoconf (portage sur unixs aisé)
  - Utilisée à différents endroits de la chaîne d'acquisition RLT
  - Utilisé pour d'autres applications (CLEF, XLFG, Univ. de Lisbonne, et ailleurs)
- Fonctionnement
  - 7 tables implémentant les noeuds d'un arbre (vedettes)
  - chaque noeud pointe sur une liste chainée de pointeurs sur une table de donnée (entrées)

Tokenizer

- Principe :
  - segmente un texte brut en phrases et en mots sans connaissance syntaxique. sortie XML
  - expressions régulières et accès au lexique
  - contient un module flex appelant des fonctions prédéfinies (puissance de flex disponible)
  - outil conçu pour être adaptable sur d'autres applications et d'autres langues
- Fonctionnement : (très simple)
  - contient un module de lexed pour les recherches dans un lexique
  - construit un DAG


fer
à
cheval
à cheval
fer à cheval
1/3 - fer - fer à cheval
1/1 - fer
2/3 - à - fer à cheval
1/1 - à
1/2 - à - à cheval
3/3 - cheval - fer à cheval
2/2 - cheval - à cheval

- $Introduce()$, $Conclude()$
- $Shift(x, y, z)$
  * x : forme permettant la recherche d'une vedette après d'éventuelles corrections
  * y : forme effective
  * z : 1 si ce segment marque la fin d'un graphe
- $Test(x)$: vrai si x est une vedette du lexique (dans version en cours de développement)

- Exemple : *D'au moins*

  tokens :

  D' au moins

  shift :

  realized forms : D' au ϵ moins

  forms : D' au1:2 au2:2 moins

  entrys :

  d' au_moins (d' au_moins)

  d' au1:2 au2:2 moins (d' au moins)

  d' au1:2 au2:2_moins (d' à le_moins)

  d' à le moins (d' à le moins)

- Exemple :

  Nous simplifions un cas plus complexe : Une abréviation marque la fin d'une phrase seulement si non suivi d'un mot sans capitale.

  faire bouillir 2 mn. sans remuer

  faire bouillir en ajoutant les carottes, les oignons, etc.

  – $\{word\}./\{spaces\}{*}\{min\}\ Shift(yytext, yytext, 0);$

  – {word}./{spaces}*{cap} {
  $Shift(yytext, yytext, 0);$
  $Shift("","", 1);\}$