Apprentissage de langages non réguliers

Soutenance de stage

Jean-Marie Madiot

LIF de Marseille, ENS Lyon

9 septembre 2009

Plan

Apprentissage de langages naturels

- Heuristiques
  Peu de formalisme (garantie)
- Classes existantes apprenables (réguliers, very simple)
  Trop petites
- Classes puissantes (context-free, mildly context-sensitive)
  Difficilement apprenables

Approche
Relier apprenabilité et formalisme

Apprentissage : basé sur l'observable

- distribution
- contextes
- ...

Idée : formaliser à l'aide des contextes

une CBFG G est la donnée de :

- Σ, alphabet
- F, ensemble fini de contextes
- P, ensemble de règles $A \longrightarrow B \cdot C$ $(A, B, C \subseteq F)$
- P_L, ensemble de règles $A \longrightarrow a$ $(A \subseteq F,\ a \in \Sigma)$

Principe : $A \longrightarrow B \cdot C$ signifie : si u, v est **associé** à tous les contextes de B, C, on associe à uv tous les contextes de A.

« on associe (a, b) à u » : $(a, b) \in f_G(u)$.
« u est dans le langage » : $(\lambda, \lambda) \in f_G(u)$

CBFG
Contextual Binary Feature Grammar

$$f_G(\lambda) = \emptyset \tag{1}$$

$$f_G(a) = \bigcup_{x \to a} x \qquad \text{si } |a| = 1 \tag{2}$$

$$f_G(w) = \bigcup_{u,v : uv = w} \; \bigcup_{\substack{x \to yz : \\ y \subseteq f_G(u) \wedge \\ z \subseteq f_G(v)}} x \qquad \text{si } |w| > 1. \tag{3}$$

Calcul de f_G sur les facteurs de w.

- $|w| = 1$, $w = a$:

$$f_G(a) = \bigcup_{x \longrightarrow a} x$$

- $|w|$ quelconque : pour tous u, v, $w = uv$, on inclut les x tels que $x \longrightarrow yz$, et tels que $f_G(u)$ contienne y (et $f_G(v)$ z)

Différence avec les grammaires hors-contexte

Hors-contextes : génératifs
CBFG : contraintes

$A, C, D \subseteq F$, ensemble de contextes

- A $(a^* b^n c^n)$
- C $(a^n b^n c^*)$
- D (d)

$$S \longrightarrow A \cup C \cdot D$$

correspond à $a^n b^n c^n d$. (contextuel)

CBFG exacte

F est abstrait :
les contextes de F n'ont pas de sens par rapport au langage
la **structure** du langage ne correspond pas à son **apparence**

Intérêt d'apprentissage : quand les contextes de F coïncident avec ceux du langage

$(a, \lambda) \in F$ peut être un contexte de b, sans que $f_G(b)$ contienne (a, λ).

CBFG exacte : quand $f_G(u)$ contient les contextes de F qui apparaissent autour de u dans le langage

Un langage qui n'admet pas de CBFG exacte

$$L = \{a^n b^m,\ n > m > 0\} \cup \{a^n c,\ n > 0\}$$

Si L admet une CBFG exacte G :

$f_G(c) \subset f_G(b^k)$ car b^k a les contextes de c dans L.

$ac \in L$ donc il existe une règle :

$$\{..., (\lambda, \lambda), ...\} \to D \cdot E$$

avec $D \subseteq f_G(a)$, $E \subseteq f_G(c) = f_G(b^k)$.

Donc $(\lambda, \lambda) \in f_G(ab^k)$.

Quelles classes de langages ?

La classe des CBFG :

- $\supset$ hors-contextes
- $\supset$ substituables
- $=$ grammaires conjonctives

La classe apprenable (CBFG **exactes**) :

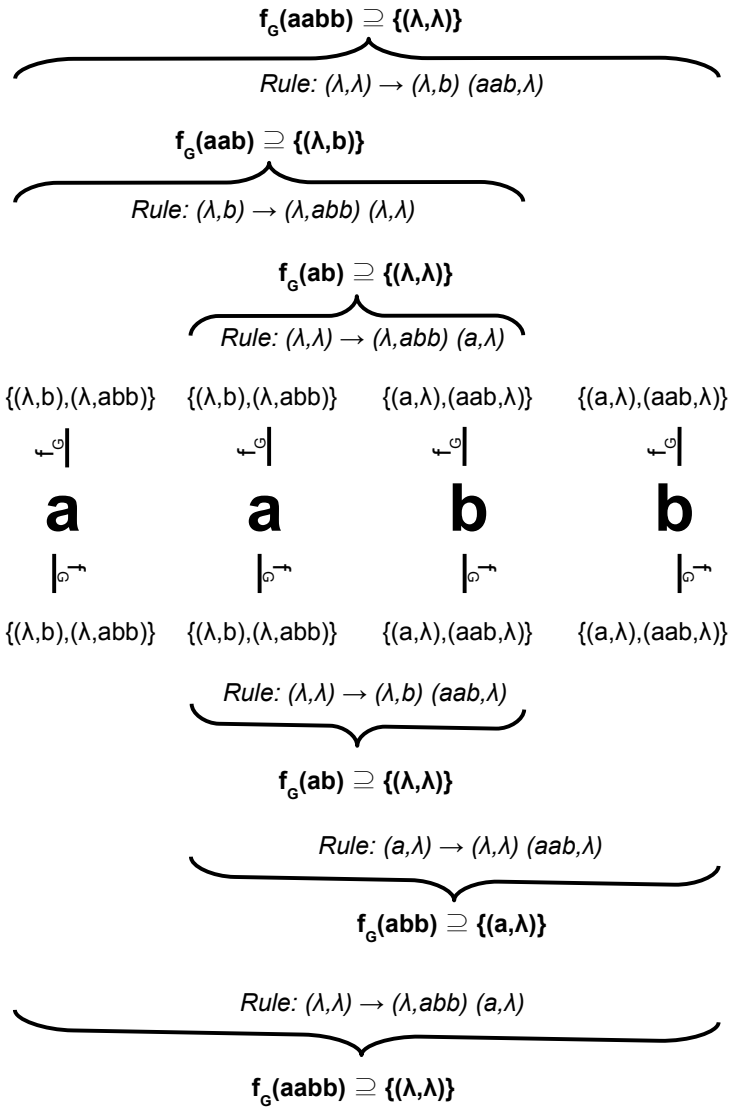
- $\supset$ réguliers
- $\supset$ subtituables, *very simple*
- $\not\supseteq\not\subseteq$ hors-contextes
- $\subset$ grammaires conjonctives

Exemple : $a^n b^n$
Règles

$$F = \{(\lambda, \lambda), (a, \lambda), (\lambda, b), (aab, \lambda), (\lambda, abb)\}$$

- $\{(\lambda, b), (\lambda, abb)\} \rightarrow a$
- $\{(a, \lambda), (aab, \lambda)\} \rightarrow b$
- $\{(\lambda, \lambda)\} \rightarrow \{(\lambda, b)\}\{(aab, \lambda)\}$
- $\{(\lambda, \lambda)\} \rightarrow \{(\lambda, abb)\}\{(a, \lambda)\}$
- $\{(\lambda, b)\} \rightarrow \{(\lambda, abb)\}\{(\lambda, \lambda)\}$
- $\{(a, \lambda)\} \rightarrow \{(\lambda, \lambda)\}\{(\lambda, aab)\}$

Exemple : $a^n b^n$
Parsing



Construction d'une grammaire

K, ensemble de mots
F, ensemble de contextes
L, langage

$F_L(u)$: ensemble des contextes de F qui entourent u dans L

Construction d'une grammaire

K, ensemble de mots
F, ensemble de contextes
L, langage

P_L est l'ensemble des $F_L(a) \longrightarrow a$
P est l'ensemble des $F_L(uv) \longrightarrow F_L(u) \cdot F_L(v)$

on définit $G_0(K, L, F) = \langle F, P, P_L, \Sigma \rangle$

Construction d'une grammaire

$G_0(K, L, F) = \langle F, P, P_L, \Sigma \rangle$

Apprentissage d'un langage L donné :
F et de K donnent une grammaire $G_0(K, L, F)$.

Choix de F et K :

- $G_0(K, L, F)$ est croissant selon K
- $G_0(K, L, F)$ est décroissant selon F

Cet algorithme apprend les CBFG exactes.

Algorithme d'apprentissage

Utilisation d'un **oracle** : « Membership queries »

Principe :
$S =$ les mots (positifs) en entrée
$K \subset$ les facteurs de S (*kernel*)
$F \subset$ tous les contextes de S

Combinaison de F et K et soumission à l'oracle ;
s'il existe un faux positif, on augmente F
s'il existe un faux négatif, on augmente K et F

Algorithme d'apprentissage

Algorithme 1 : Algorithme d'apprentissage de CBFG

Données : des mots, $\{w_1, w_2 \ldots, \}$, un oracle $\mathcal{O}$
Résultat : Une suite de CBFG $G_1, G_2, \ldots$
$K \leftarrow \emptyset$ $F \leftarrow \{(\lambda, \lambda)\}$ $D \leftarrow \emptyset$ $G_0 = G_0(K, \mathcal{O}, F)$;
pour w_i **faire**
  $D \leftarrow D \cup \{w_i\}$;
  $S \leftarrow Con(D) \odot Sub(D)$;
  si $\exists w \in S$ *tel que* $w \in L(G_{i-1}) \setminus L$ **alors**
    $F \leftarrow Con(D)$;
  fin
  si $\exists w \in S$ *tel que* $w \in L \setminus L(G_{i-1})$ **alors**
    $K \leftarrow Sub(D)$;
    $F \leftarrow Con(D)$;
  fin
  Renvoyer $G_i = G_0(K, \mathcal{O}, F)$;
fin

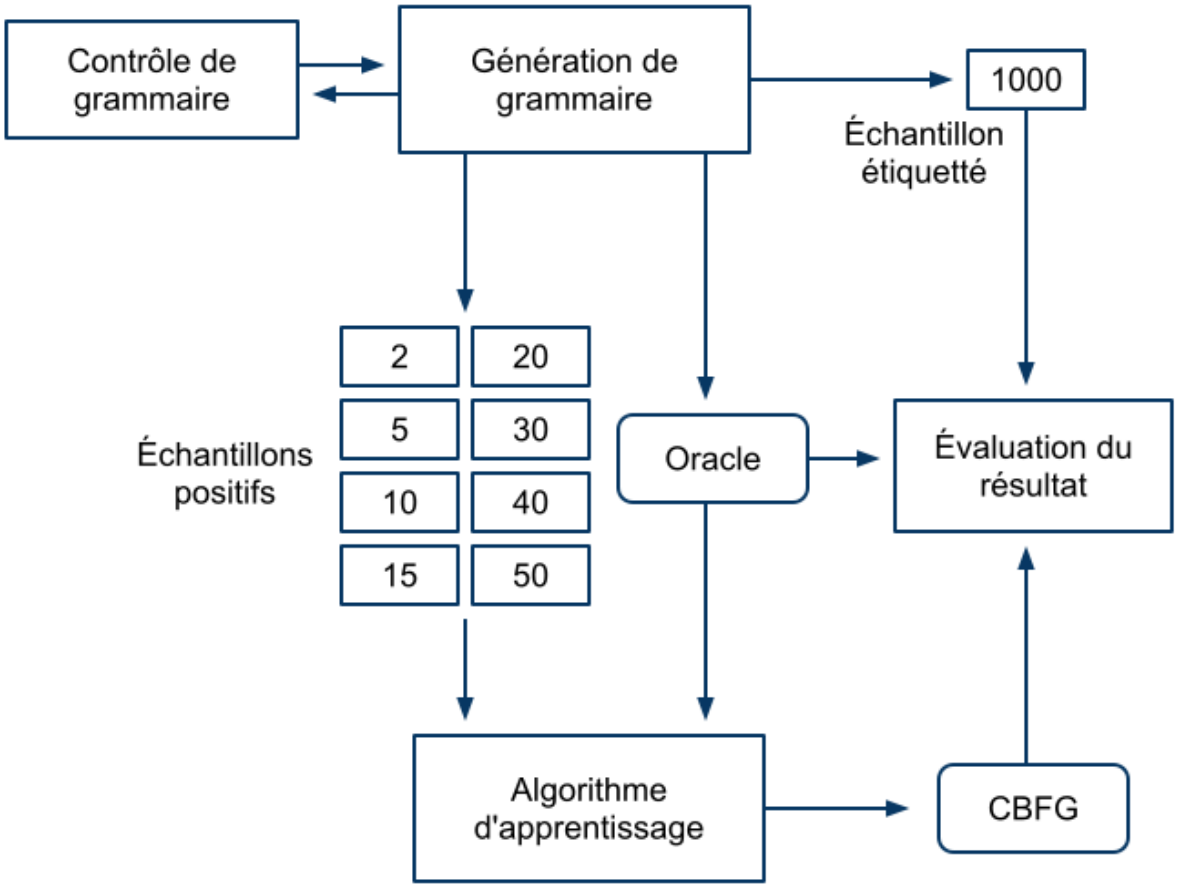
Expérimentations : but

Convergence : prouvée
Algorithme : polynomial

Comportement ?

- temps raisonnable ?
- approche progressive/soudaine ?
- sur/sous-généralisation ?

Protocole expérimental



Choix de la classe de test

CBFG

- CBFG : non génératif, pas d'ESC
- exactitude indécidable

Mais les hors-contexte

- toujours un défi
- bonne variété parmi les CBFG
- connus
  (comparaisons, résultats existants, notoriété)

Génération de grammaire
Hors contexte

- T : terminaux
- N : non-terminaux
- R : règles

$\#T$, $\#N$ et $\#R$ sont fixés.

Graphe de sommets T et N
Une règle : deux arêtes

On relie aléatoirement les sommets à l'aide d'arêtes.
On garde la composante connexe du graphe qui contient S.

Analyse de grammaire

Exploration du graphe à partir des terminaux

Calculable :

- L est fini (ou $L = \emptyset$)
  $\longrightarrow$ génération d'une nouvelle grammaire
- des éléments sont inaccessibles
  $\longrightarrow$ élimination des éléments

Incalculable :

- $L = \Sigma^*$
- L est régulier

Ensemble structurellement complet

Ensemble structurellement complet :
Ensemble de mots S tq :

$$\forall r \in R,\ \exists w \in L,\ w \text{ se dérive avec } r$$

Construction : exploration à partir des terminaux.

Idée de l'algorithme :
Calcul de l'ensemble des couples (règles accessibles, mots) pour chaque sommet
Fusions successives de ces ensembles
Détections des boucles (le langage est alors infini)

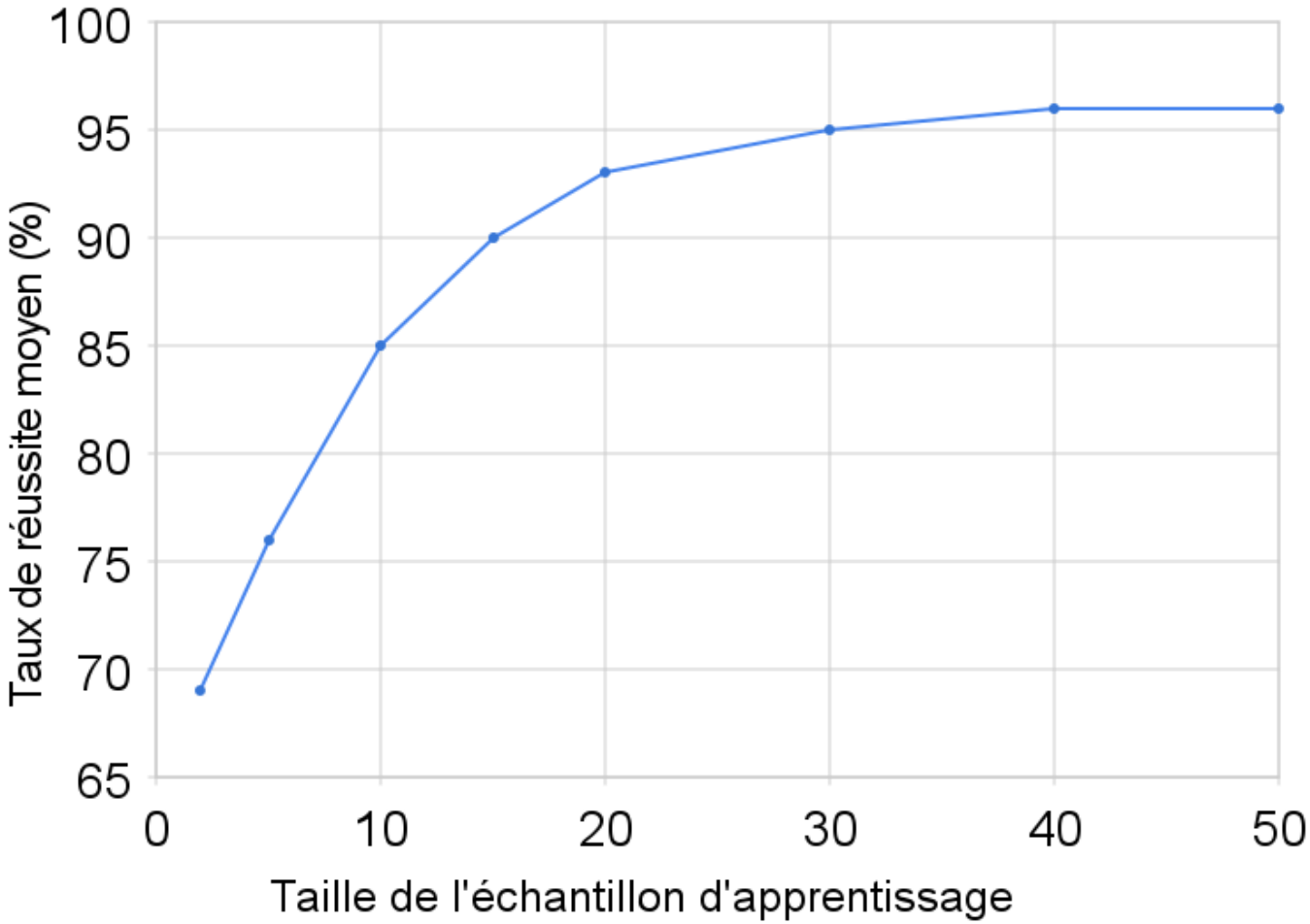
Génération de l'échantillon

Mots : *parsing* d'un mot aléatoire

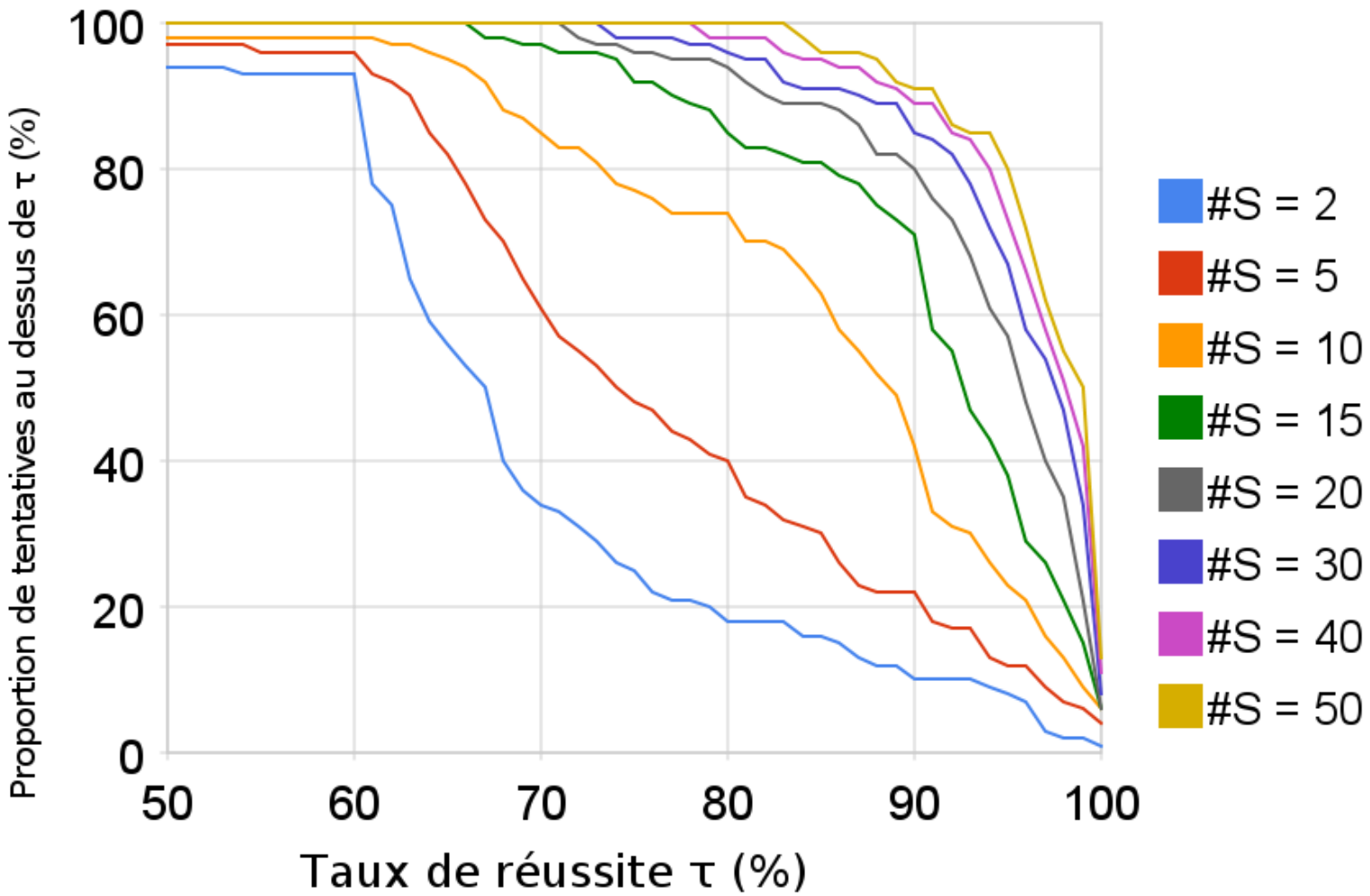
- mots positifs : apprentissage
- mots étiquetés : test (minimum 40% de chaque)
- oracle : sous la forme d'une grammaire hors-contexte

Tailles d'apprentissage : 2, 5, 10, 15, 20, 30, 40, 50
Même échantillon de test, de taille 1000
Passage à l'échelle : 10000

Résultats
Taux de réussite moyen



Résultats
Proportion au dessus d'un score donné



Résultats
Répartition des erreurs

$\#S$	faux positifs	faux négatifs
2	2.3 %	28.1 %
5	3.3 %	20.0 %
10	2.4 %	12.1 %
15	2.6 %	6.5 %
20	1.6 %	4.6 %
30	1.1 %	3.6 %
40	0.69 %	3.0 %
50	0.55 %	2.5 %

Sous-généralisation (recherchée)

Conclusion

Problèmes

- ensemble structurellement complet
- nombre important de requêtes (naïf)

Justification de l'oracle

- langue maternelle
- « Zulu, interactive learning competition »

Questions ouvertes

- remplacer l'oracle
- préciser la classe des CBFG exactes
- sémantique d'une dérivation

Des questions ?