Concevoir, améliorer et exploiter une méta-grammaire factorisée du français

http://alpage.inria.fr

Éric de la Clergerie
<Eric.De_La_Clergerie@inria.fr>

INRIA Paris-Rocquencourt / Univ. Paris Diderot

Séminaire de recherche en Linguistique
Université de Genève – 9 Novembre 2010

L'équipe ALPAGE

Analyse Linguistique Profonde À Grande Échelle

- Équipe-projet INRIA fondée en 2007
  par fusion du projet ATOLL et du groupe TALaNa (Univ. Paris 7)
  dirigée par Laurence Danlos (Paris 7)
  environ 15-20 people (9 permanents)

- **Thématique**: TAL

- **Objectifs**:
  - marier expertise linguistique (Profond) et informatique (Grande Échelle)
  - développer des outils et ressources linguistiques, pour le français et d'autres langues
  - les utiliser dans des applications de traitement de grands corpus (efficacité, robustesse, précision)
    $\Longrightarrow$ acquisition de connaissances / extraction d'information

Les origines

- Développement du système DyALog
  - environnement de programmation en logique (Prolog)
  - partage de calculs (tabulation $\Longrightarrow$ chart parsing)
  - plusieurs formalismes de grammaires d'unification dont Grammaires d'Arbres Adjoints (TAG – Joshi)

- Participation à la campagne d'évaluation EASy (en 2004)
  $\Longrightarrow$ nécessité de développer (rapidement):
  - lexique syntaxique Lefff (Sagot & Clément)
  - segmentation & entités nommées SxPipe (Sagot & Boullier)
  - grammaire TAG FRMG

- Processus d'amélioration et d'exploitation
  - la chaîne de traitement ALPAGE (dont FRMG) s'améliore par son utilisation sur corpus + *feedback*

Outline

1. Concevoir
2. Utiliser
3. Améliorer
4. Exploiter

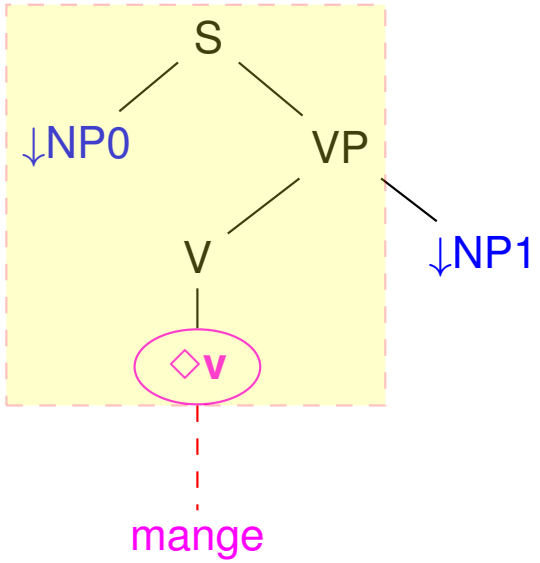
Problèmes avec les grammaires TAGs

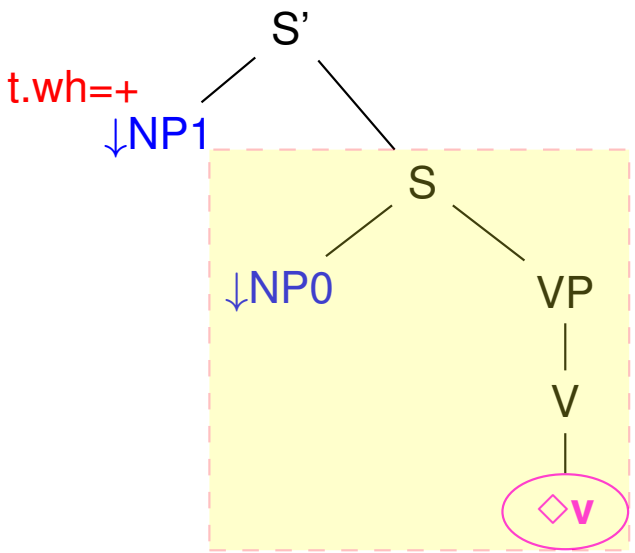
Le domaine étendu de localité des arbres TAG est un avantage

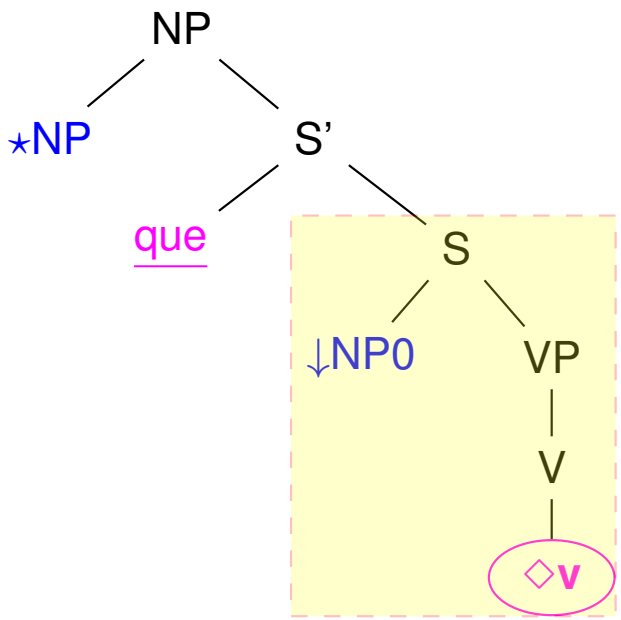
- cadres de sous-catégorisation
- dépendances à longue distance (extractions/mouvements)

mais entraîne

- une explosion du nombre d'arbres: plusieurs (dizaines de) milliers
  $\Longrightarrow$ problème d'efficacité de l'analyse
- beaucoup de sous-arbres en commun
  $\Longrightarrow$ problème de développement et maintenance







Solutions

Pour FRMG, le choix est:

- utilisation d'opérateurs de factorisation de DyALog dans les arbres TAG.
  mais difficile d'écrire directement des arbres factorisés complexes

- utilisation d'une Métagrammaire
  description modulaire et factorisée de phénomènes syntaxiques
  $\rightsquigarrow$ génération des arbres factorisés à partir des descriptions

Factorisation d'arbres

Principe : combiner plusieurs arbres en un seul, pour partager des parties communes

- plusieurs traversées possibles par arbre (Harbush)
- ou utiliser des opérateurs réguliers dans les arbres (DyALog):

  disjonctions $T[t_1;t_2] \equiv T[t_1] \cup T[t_2]$

  répétitions (Kleene Stars) $T[t@*] \equiv \{T[\epsilon], T[t], T[(t,t)], \ldots\}$

  entrelacements (ordre libre entre séquences)
  $(t_1, t_2)\#\#t_3 \equiv (t_1, t_2, t_3; t_1, t_3, t_2; t_3, t_1, t_2)$

  optionalité $t? \equiv (t; \epsilon)$

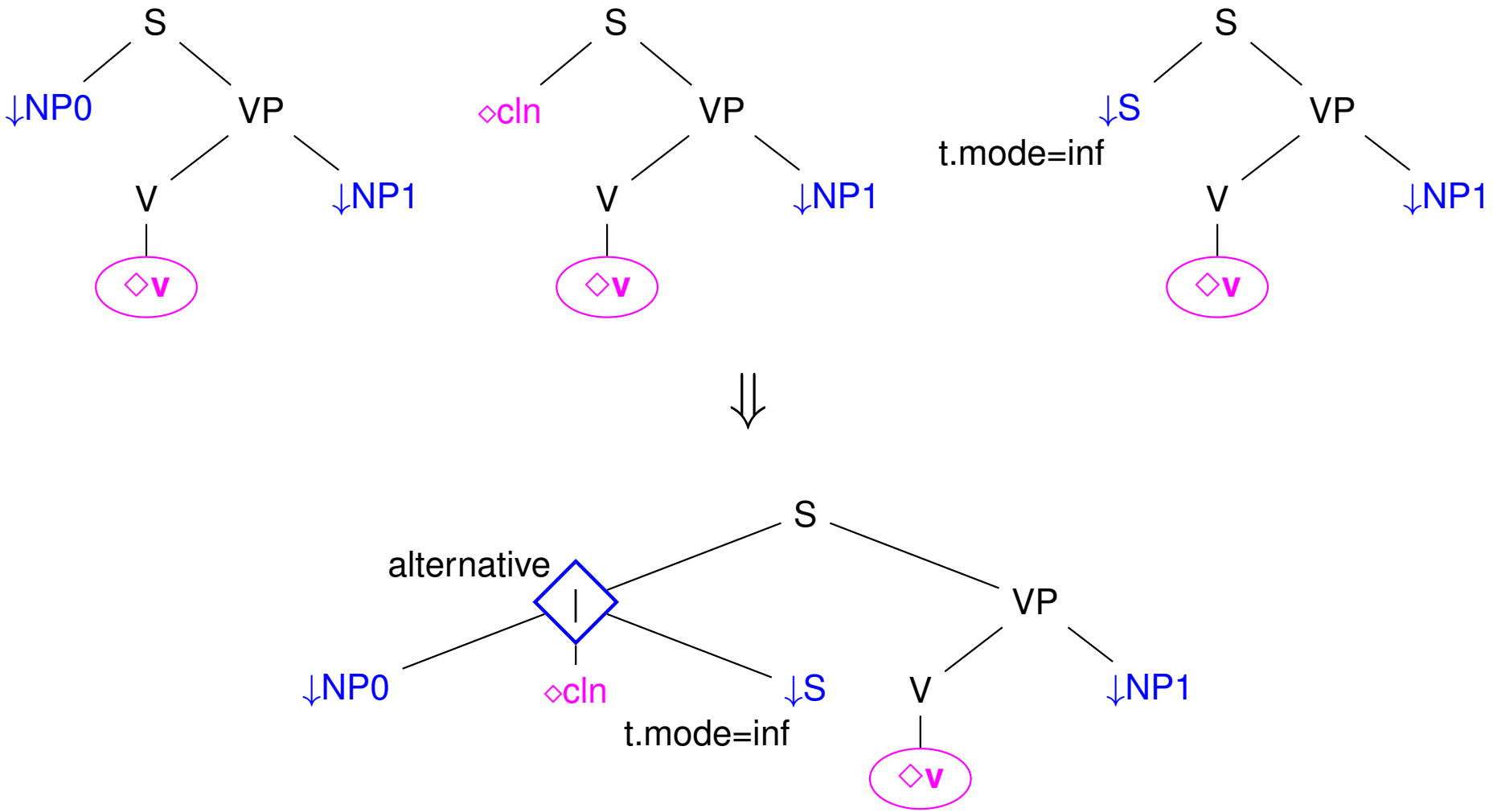
  gardes (noeuds avec gardes) $T[G_+, t; G_-] \equiv T[t].\sigma_+ \cup T[\epsilon].\sigma_-$
  gardes: formules booléennes sur des équations entre valeurs de traits

La factorisation

- ne change ni le pouvoir expressif, ni la complexité des TAGs
- mais élimination factorisation $\Longrightarrow$ taille exponentielle de la grammaire
- utilisation sans élimination dans DyALog

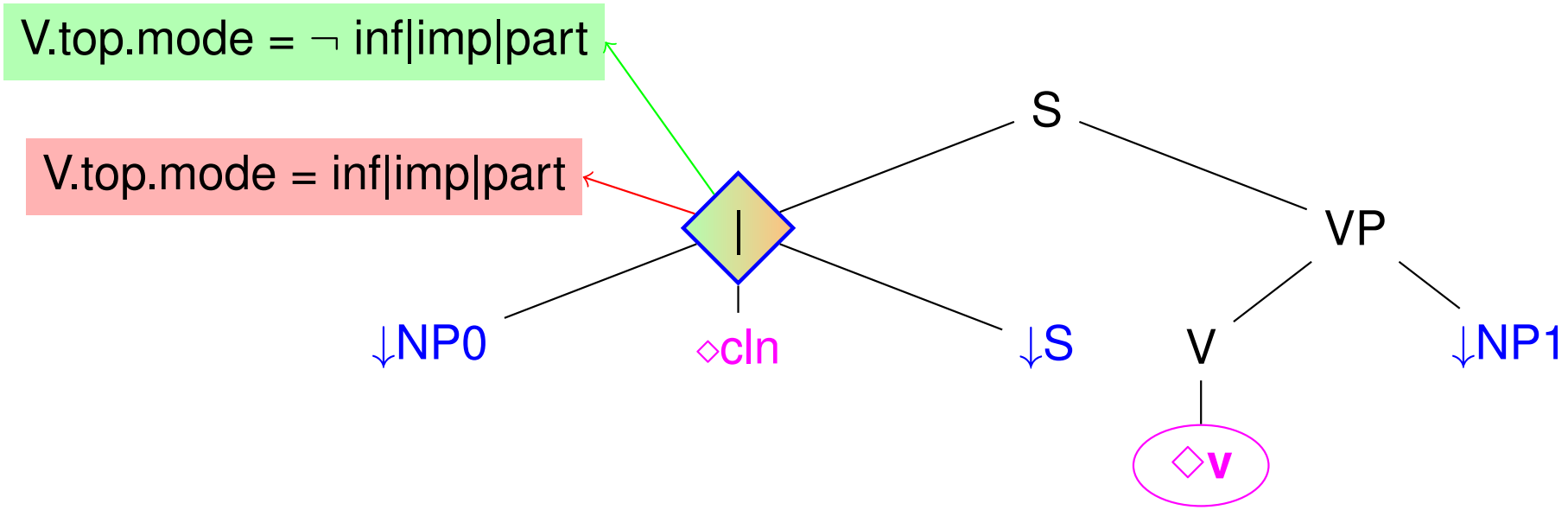
Disjonction

Plusieurs réalisations possibles pour le sujet (NP, cln, S, . . .)



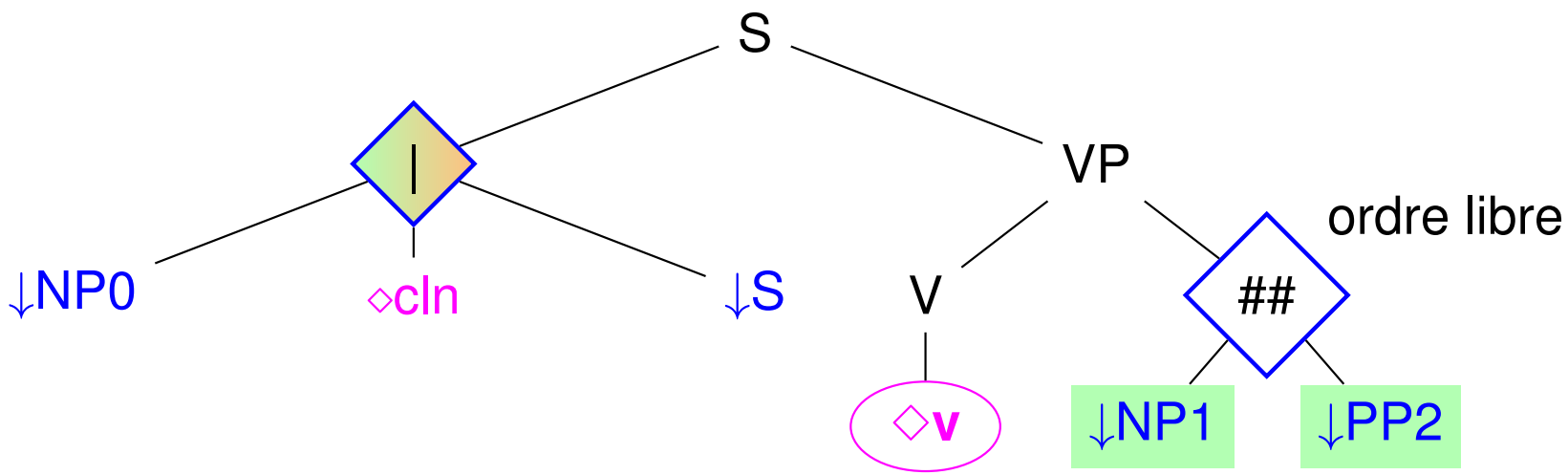
Gardes

Le sujet est absent sous certaines conditions



Ordre libre

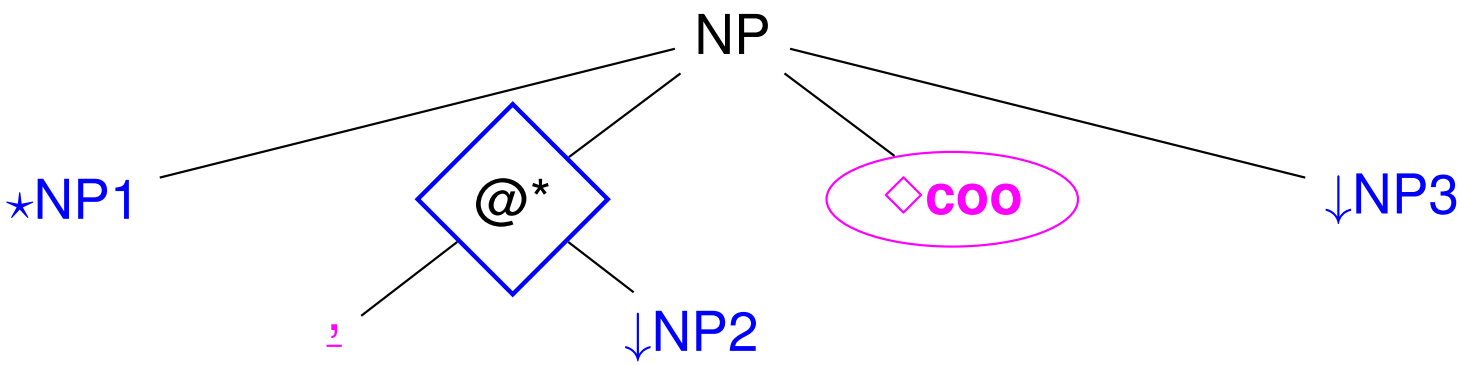
Les arguments verbaux ne sont pas ordonnés



Défactorisation: $(1_{\text{no subj}} + 3_{\text{subj}}) * (1_{\text{no arg}} + 2_{\text{1 arg}} + 2_{\text{2 args}}) = 20$ arbres

Répétition (étoile de Kleene)

Écriture naturelle de la coordination avec une répétition



Les Méta-Grammaires

Definition (Méta-Grammaire)

Description modulaire par classes regroupant des contraintes, avec héritage

Les Méta-Grammaires

Definition (Méta-Grammaire)

Description modulaire par classes regroupant des contraintes, avec héritage

```
class collect_real_subject_canonical {
 <: collect_real_subject;
 $arg.extracted = value(~cleft);
 S >> VSubj; V >> psubj;
 VSubj < V; VMod < psubj;
 node psubj: [cat:N2, id:subject,
              top:[wh:-,sat:+]];
 - psubj::agreement; psubj = psubj::N;
 psubj =>
  node(Infl).bot.inv = value(+),
  $arg.extracted = value(-),
  $arg.real = value(N2),
  desc.extraction = value(~-),
  node(V).top.mode= value(~inf|imp|...);
 ~psubj=> node(Infl).bot.inv = value(~+);
}
```

- Héritage (<:)
- Contraintes
  - dominance (>> et >>+)
  - précédence (<)
  - égalité (=)
  - Décorations (FS)
    - ★ noeuds
    - ★ classe
  - Éq. entre chemins (.)
    - ★ noeuds (node psubj)
    - ★ classe (desc)
    - ★ variable ($arg)
- Ressources + / Besoins -
  - Espace de noms (::)
- Gardes (=>)

SMG format (suite)

```
%% Macro for Finite Sets
template @pcas_set = -|+|à|de|dans|sur|vers|...

%% Macro for paths
path @arg0 = .ht.arg0
path @real0 = .@arg0.real

class adj_on_noun { %% Adjectives on nouns
 <: adj_as_modifier;
 desc.@real0 = value(N2); %% = desc.ht.arg0.real
 node Root : [cat: N];
}

%% Disabling a class and all its descendants
 disable verb_categorization_passive
 disable verb_extraction_wh
```

Namespaces: illustration

Possibilité d'importer plusieurs fois les contraintes d'une classe *C*, avec renommage des noeuds, des variables et des ressources.

```
class agreement {
    %% Generic class to add agreement equations
    + agreement;
    father(N).bot.number = node(N).top.number;
    father(N).bot.gender = node(N).top.gender;
    father(N).bot.person = node(N).top.person;
}

%% le garçon le plus grand / la fille la plus grande
class superlative_as_adj_mod {
    <: superlative_as_mod;
    node(Foot).cat = value(adj);
    - det::agreement; det = det::N;
    - adj::agreement; Foot = adj::agreement;
}
```

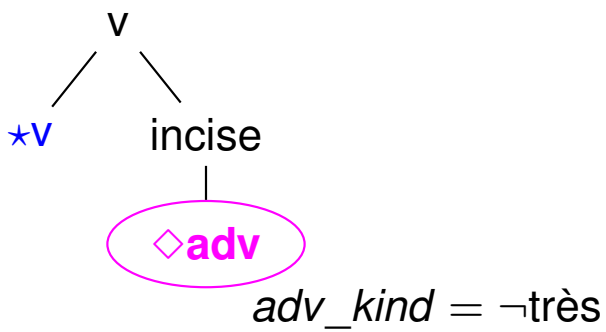
Exemple d'héritage (pour les adverbes)

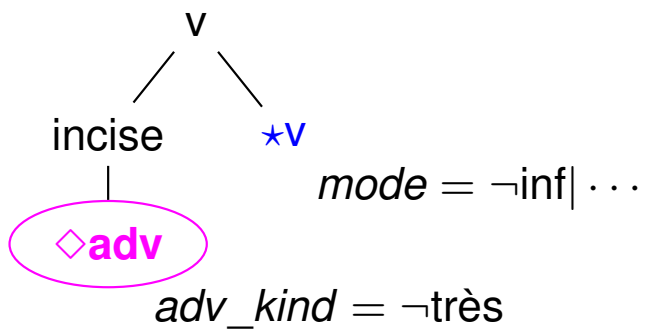
```
class categories { %% base for anchored tree
    node Anchor : [type:anchor];
    desc.@htcat = node(Anchor).cat;
    node(Anchor).id = node(Anchor).cat;
    desc([ht:@ht_fs]); }

class adv { %% Adverbs
    <: categories;
    node Adv : [cat: adv, bot: [degree: -]]; Adv=Anchor;
    desc.ht = value([...]); }

class adv_modifier {   %% Adverbs as modifier
    <: adv;
    - shallow_auxiliary;
    Root >> Incise;
    Incise >> Adv;
    node Incise : [cat:incise,id:incise, type:std];
    node(Root).bot = node(Foot).top;  }
```

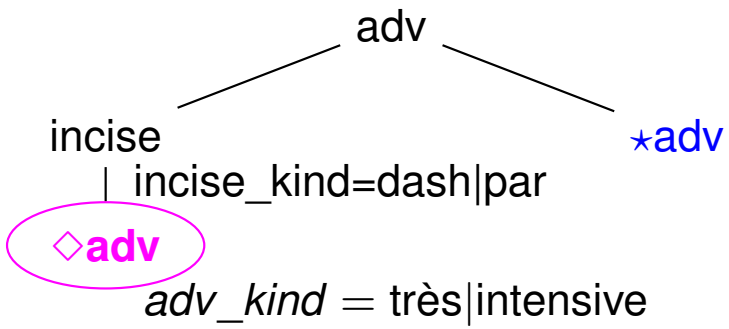
Exemple pour adv (suite)

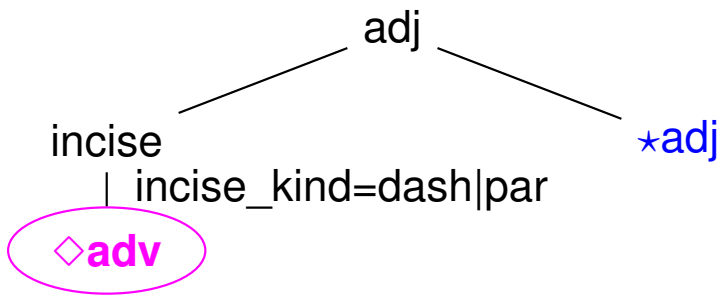
```
class adv_v { %% Adverbs on verbs: may only occur after verb
    <: adv_modifier;
    node Root : [cat: v];  Foot < Anchor;
    node(Adv).bot.adv_kind = value(~très); }

class adv_before_v { %% Adverbs on non tensed verbs
  <: adv_modifier;
  node Root : [cat: v];  Anchor < Foot;
  node(Adv).bot.adv_kind = value(~très);
  node(Foot).top.mode = value(infinitive|participle|gerundive);
  desc.@kind0 = value(-); }

class adv_adv { %% Adverbs on adverbs: très très petit
    <: adv_modifier;
    node Root : [cat: adv];  Adv < Foot;
    node(Incise).bot.incise_kind = value(dash|par);
    node(Adv).bot.adv_kind = value(très|intensive); }

class adv_adj { %%  Adverbs on adjectives: très petit
    <: adv_modifier;
    node Root : [cat: adj]; Adv < Foot;
    node(Incise).bot.incise_kind = value(dash|par); }
```

Adv: quelques arbres





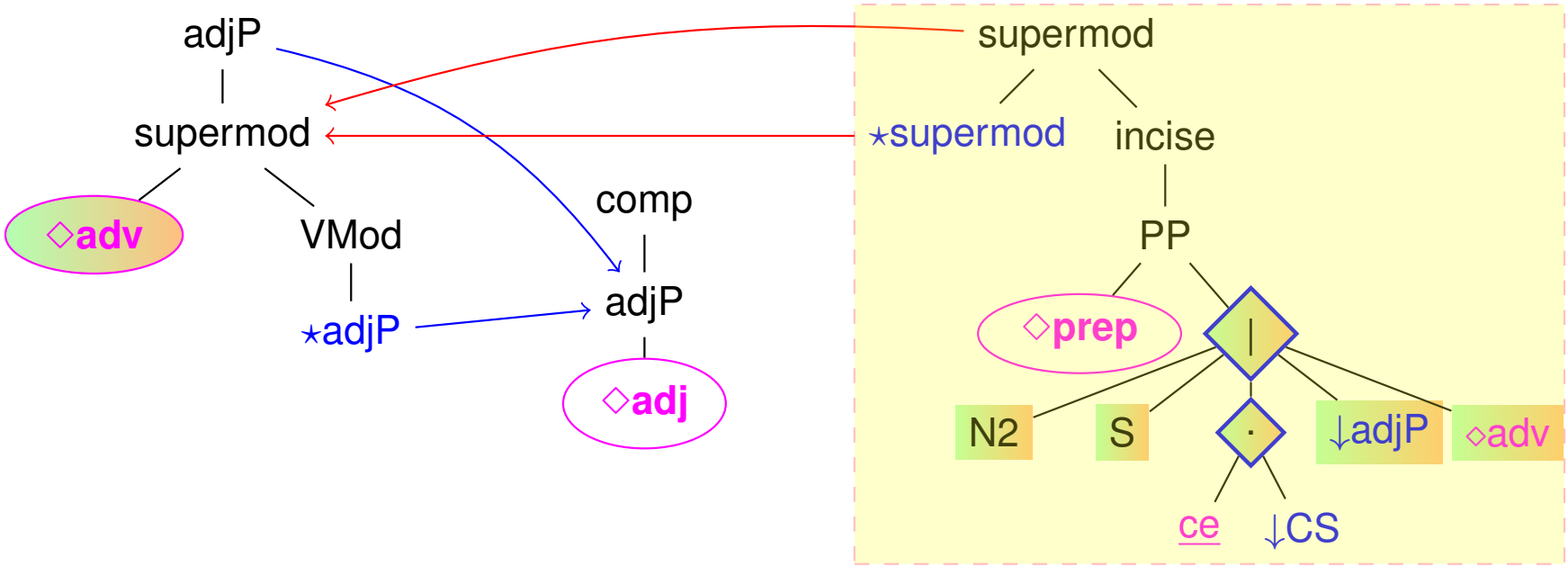




Exemple 2 autour des comparatives

- Des classes pour les comparatives et superlatives
- Phénomène similaire pour *(suffisamment|trop|assez) fort pour travailler*
  $\Longrightarrow$ légères modifs de quelques classes et une classe supplémentaire

```
class mod_on_quantity_modifier {
  <: mod_on_superlative; <: prep;
  node(Foot).top.supermod_kind = value(quantity);
  Incise >> PP; Foot < PP;
  node(prep).bot.pcas = value(pour); }
```



Compiler la méta-grammaire

Compilateur MGCOMP, développé avec DYALOG

Compiler la méta-grammaire

Compilateur MGCOMP, développé avec DYALOG

Étape 1: Classes terminales

Héritage des contraintes par les classes terminales (+ vérif contraintes)

Compiler la méta-grammaire

Compilateur MGCOMP, développé avec DYALOG

Étape 1: Classes terminales

Héritage des contraintes par les classes terminales (+ vérif contraintes)

Étape 2: Classes neutres

- Croisement des classes terminales pour neutraliser ressources & besoins
  - $C_1[-R \cup \mathcal{K}_1] x C_2[+R \cup \mathcal{K}_2] = (C_1 x C_2)[=R \cup \mathcal{K}_1 \cup \mathcal{K}_2]$
  - (Espace de nom) $\Longrightarrow$ import classe productrice avec renommage
    $C_1[-N::R \cup \mathcal{K}_1] x C_2[+R \cup \mathcal{K}_2] = (C_1 x N::C_2)[=N::R \cup \mathcal{K}_1 \cup N::\mathcal{K}_2]$
- Réduction des gardes (quand possible)
- Vérification des contraintes

Compiler la méta-grammaire

Compilateur MGCOMP, développé avec DYALOG

Étape 1: Classes terminales

Héritage des contraintes par les classes terminales (+ vérif contraintes)

Étape 2: Classes neutres

- Croisement des classes terminales pour neutraliser ressources & besoins
  - $C_1[-R \cup \mathcal{K}_1] x C_2[+R \cup \mathcal{K}_2] = (C_1 x C_2)[=R \cup \mathcal{K}_1 \cup \mathcal{K}_2]$
  - (Espace de nom) $\Longrightarrow$ import classe productrice avec renommage
    $C_1[-N::R \cup \mathcal{K}_1] x C_2[+R \cup \mathcal{K}_2] = (C_1 x N::C_2)[=N::R \cup \mathcal{K}_1 \cup N::\mathcal{K}_2]$
- Réduction des gardes (quand possible)
- Vérification des contraintes

Étape 3: Arbres TAG/TIG

Utilisation des contraintes des classes neutres pour construire les arbres

FRMG: Meta Grammaire du Français

- Sous-catégorisation des verbes : sujet `subj`, attribut `acomp`, `object`, `vcomp`, `scomp`, `wh-comp`, `prep-vcomp`, `prep-scomp` `prep-object`, `prep-acomp` au plus 3 arg. verbaux (sujet inclus)
- Constructions auxiliaires, verbes à contrôle
- Diverses réalisations (NP, clitique, infinitive, complétive, . . .) et position (pre, post, post-clitique) du sujet
- extraction des arguments et compléments (questions, relatives, clivées, et topic)
- constructions passives, actives, causatives (partiel)
- coordinations partielles (avec quelques ellipses), comparatives, superlatives
- modifieurs (incises) du verbe à diverses positions (participiales, PP, adv, . . .),
- verbes «supports» (prendre conscience de)
- ponctuation

FRMG en quelques tables

Classes	Arbres	Init.	Aux.	Aux. Env.	Aux. Gauches	Aux. Droits
279	207	44	163	36	46	81

Distribution par catégories d'arbres

ancrés	v	coo	adv	adj	csu	prep	aux	np	nc	det	pro	¬ancrés
142	21	26	40	20	6	5	2	3	1	1	5	65

Distribution par la catégorie de l'ancre

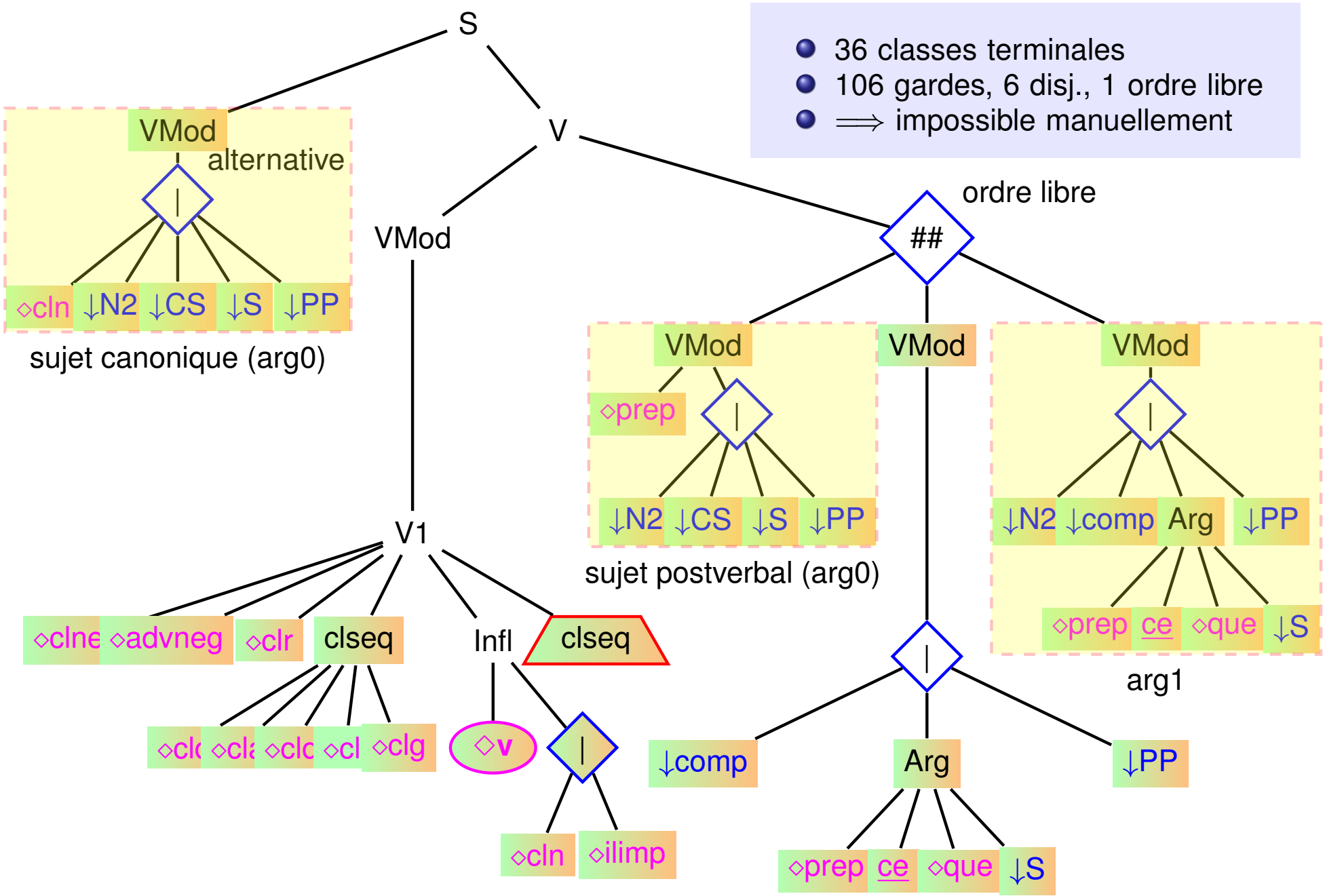
Canonique	Extr.	Actif	Passif	Quest.	Rel.	Clivées	Topic
7	23	13	7	4	4	10	5

Distribution par phénomènes syntaxiques

Gardes	Disjonctions	Entrelacement	Étoiles de Kleene
2609	152	22	27

Utilisation des opérateurs de factorisation

Arbre #198: verbe canonique (simplifié)



Lien lexique / Grammaire: ancrage par *hypertag*

Grammaire FRMG

hypertag #198

arg0	**arg0**	extracted	-
		kind	**subj**
		pcas	-
		real	**real0** - | **CS** | **N2** | **PP** | **S** | **cln** | **prel** | **pri**
arg1	**arg1**	extracted	-
		kind	**kind1** - | **acomp** | **obj** | **prepacomp** | **prepobj**
		pcas	**pcas1** + | - | **apres** | **à** | **avec** | **de** | **par** | ...
		real	**real1** - | **CS** | **N** | **N2** | **PP** | **S** | **V** | **adj** | **cla** | ...
arg2	**arg2**	extracted	-
		kind	**kind2** - | **prepacomp** | **prepobj** | **prepscomp** | **prepvcomp** | **scomp** | **vcomp** | **wh-comp**
		pcas	**pcas2** - | + | **apres** | **à** | ...
		real	**real2** - | **CS** | **N** | **N2** | **PP** | **S** | ...
cat	**v**		
diathesis	**active**		
refl	**refl**		

Lien lexique / Grammaire: ancrage par *hypertag*

Grammaire FRMG
hypertag #198

```
[
  arg0   arg0   [ extracted  -
                  kind       subj
                  pcas       -
                  real       real0  - | CS | N2 | PP | S | cln | prel | pri ]

  arg1   arg1   [ extracted  -
                  kind       kind1  - | acomp | obj | prepacomp | prepobj
                  pcas       pcas1  + | - | apres | à | avec | de | par | ...
                  real       real1  - | CS | N | N2 | PP | S | V | adj | cla | ... ]

  arg2   arg2   [ extracted  -
                  kind       kind2  - | prepacomp | prepobj | prepscomp
                                    | prepvcomp | scomp | vcomp | wh-comp
                  pcas       pcas2  - | + | apres | à | ...
                  real       real2  - | CS | N | N2 | PP | S | ... ]

  cat        v
  diathesis  active
  refl       refl
]
```

Lexique LEFFF
hypertag «promettre»

```
[
  arg0   [ kind  subj | -
           pcas  - ]

  arg1   [ kind  obj | scomp | -
           pcas  - ]

  arg2   [ kind  prepobj | -
           pcas  à | - ]

  refl   -
]
```

Lien lexique / Grammaire: ancrage par *hypertag*

Grammaire FRMG
hypertag #198

```
arg0  arg0 = [ extracted  -
               kind       subj
               pcas       -
               real       real0 - | CS | N2 | PP | S | cln | prel | pri ]

arg1  arg1 = [ extracted  -
               kind       kind1 - | acomp | obj | prepacomp | prepobj
               pcas       pcas1 + | - | apres | à | avec | de | par | ...
               real       real1 - | CS | N | N2 | PP | S | V | adj | cla | ... ]

arg2  arg2 = [ extracted  -
               kind       kind2 - | prepacomp | prepobj | prep-
                                  scomp | prepvcomp | scomp | vcomp
                                  | whcomp
               pcas       pcas2 - | + | apres | à | ...
               real       real2 - | CS | N | N2 | PP | S | ... ]

cat        v
diathesis  active
refl       refl -
```

Lexique LEFFF
hypertag «promettre»

```
arg0  [ kind  subj | -
        pcas  - ]

arg1  [ kind  obj | scomp | -
        pcas  - ]

arg2  [ kind  prepobj | -
        pcas  à | - ]

refl  -
```

Les variables X propagent les valeurs des hypertags aux noeuds et gardes
$\Longrightarrow$ bloque les chemins impossibles dans les arbres factorisés

Degré et impact de la factorisation

- expansion complète des arbres $\Longrightarrow \sim$ 2 millions d'arbres (FRMG 2007)
- expansion partielle sur #198 (en gardant les disjonctions)
  + intersection avec LEFFF 195 cadres de sous-cat $\Longrightarrow$ 5729 arbres (+ 206)
- trop large pour calculer la relation coin-gauche
- test des 2 versions sur les 4000 phrases de **EasyDev**
  $\Longrightarrow$ **factorisation**: pas de surcoût and plus d'optimisation

analyseur	**avg**	**median**	90%	99%
+fact. -lc (207 arbres)	**1.33**	0.46	2.63	12.24
-fact. -lc (5935 arbres)	1.43	0.44	2.89	14.94
+fact. +lc (207 arbres)	**0.64**	0.16	1.14	6.22

Outline

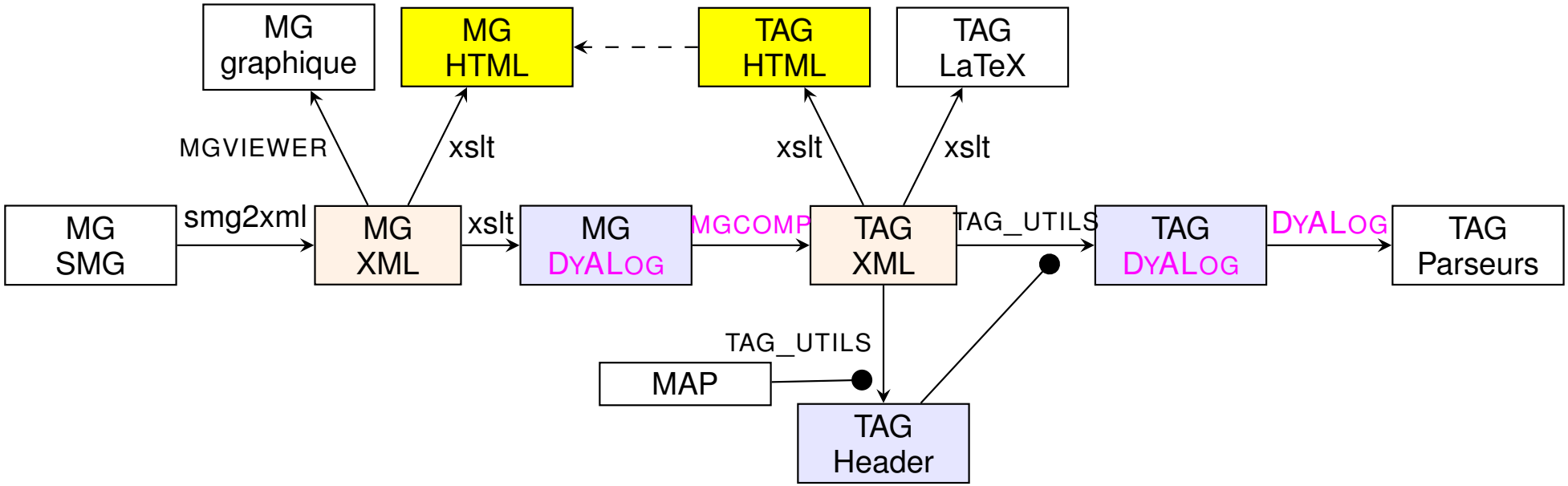
1 Concevoir

2 Utiliser

3 Améliorer

4 Exploiter

Mettre les briques ensemble !



Fonctionalités de l'analyseur

- identification des arbres auxiliaires TIG (gauches et droits) et TAG (enveloppants)
- compilation d'un analyseur tabulaire, décrivant l'ensemble des parcours d'un arbre sous forme d'une méta-transition 2SA
- modèle d'analyse par suspension-reprise au niveau des noeud
- analyse un DAG en entrée (SxPipe+ info Lefff)
- retourne l'ensemble des analyses complètes
  possibilité de retourner des analyses partielles couvrant la phrase
- mode 'just-in-time' en flux
- exploitation lexicalisation des arbres, quand disponible
  $\Longrightarrow$ filtrage d'une partie de la grammaire
- utilisation de la relation coin-gauche
- identification des traits non perturbés par adjonction
- ...

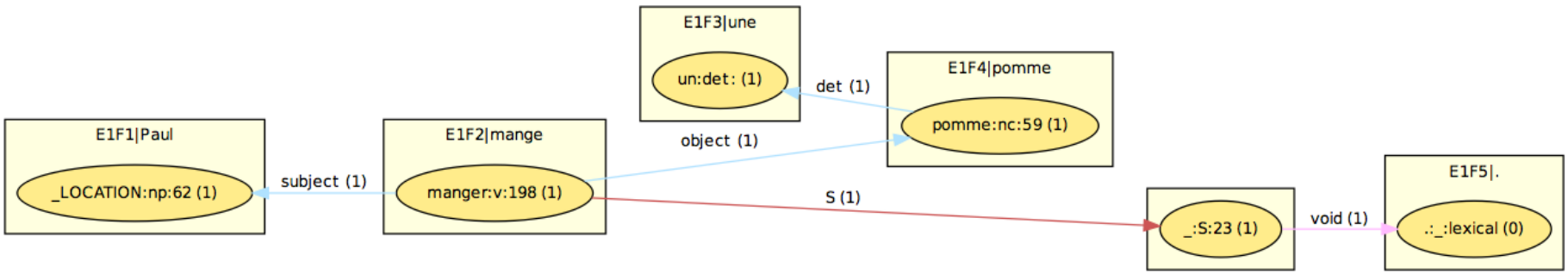
Modes d'utilisation

- en ligne de commande (longue séquence de pipe unix)
- en mode serveur PARSERD avec plusieurs clients
  - client WEB PARSER.PL (`http://alpage.inria.fr/parserdemo`)
  - client CALLPARSER pour des jeux de phrases
  - client DISPATCH.PL pour traitement distribué de gros corpus sur clusters ou grilles.
- avec un shell adapté avec FRMG_SHELL (démo)

Sorties syntaxique: Format DepXML

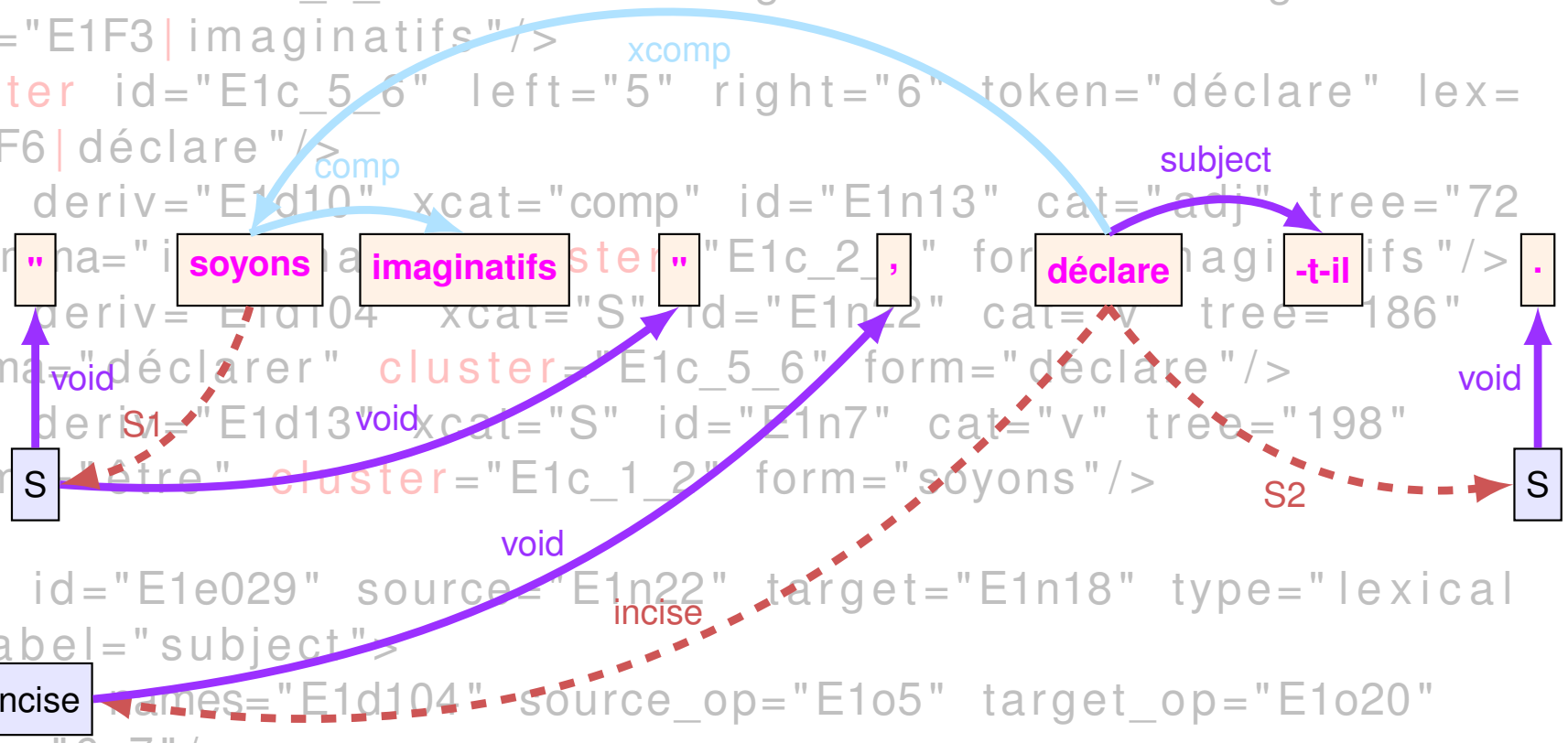
Transformation des forêts de dérivation TAG en

- forêt de dépendances gouverneur/gouverné
- format de réprésentation DepXML et vues graphiques
- désambiguisation (règles pondérées heuristiques)

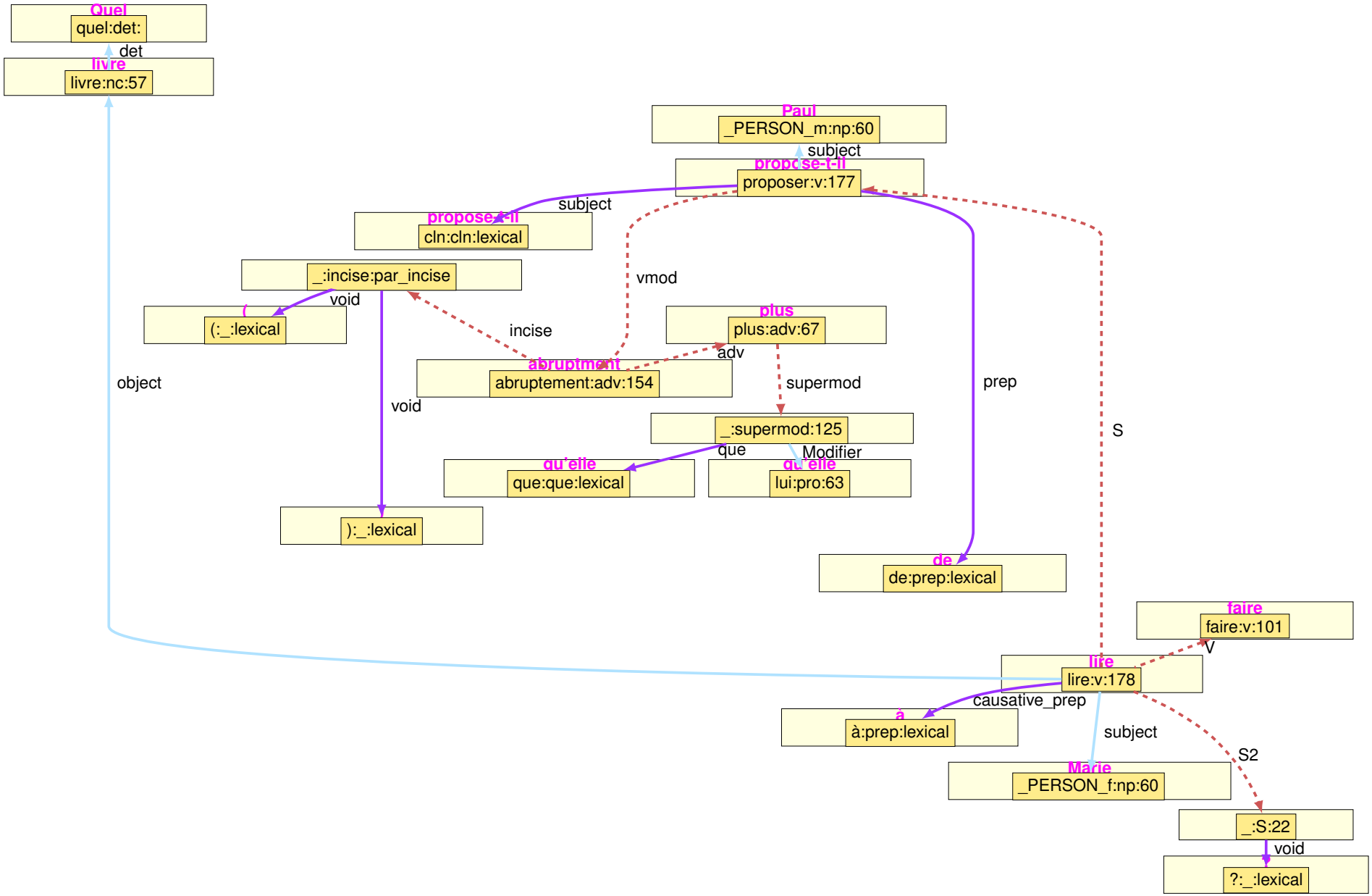


DepXML

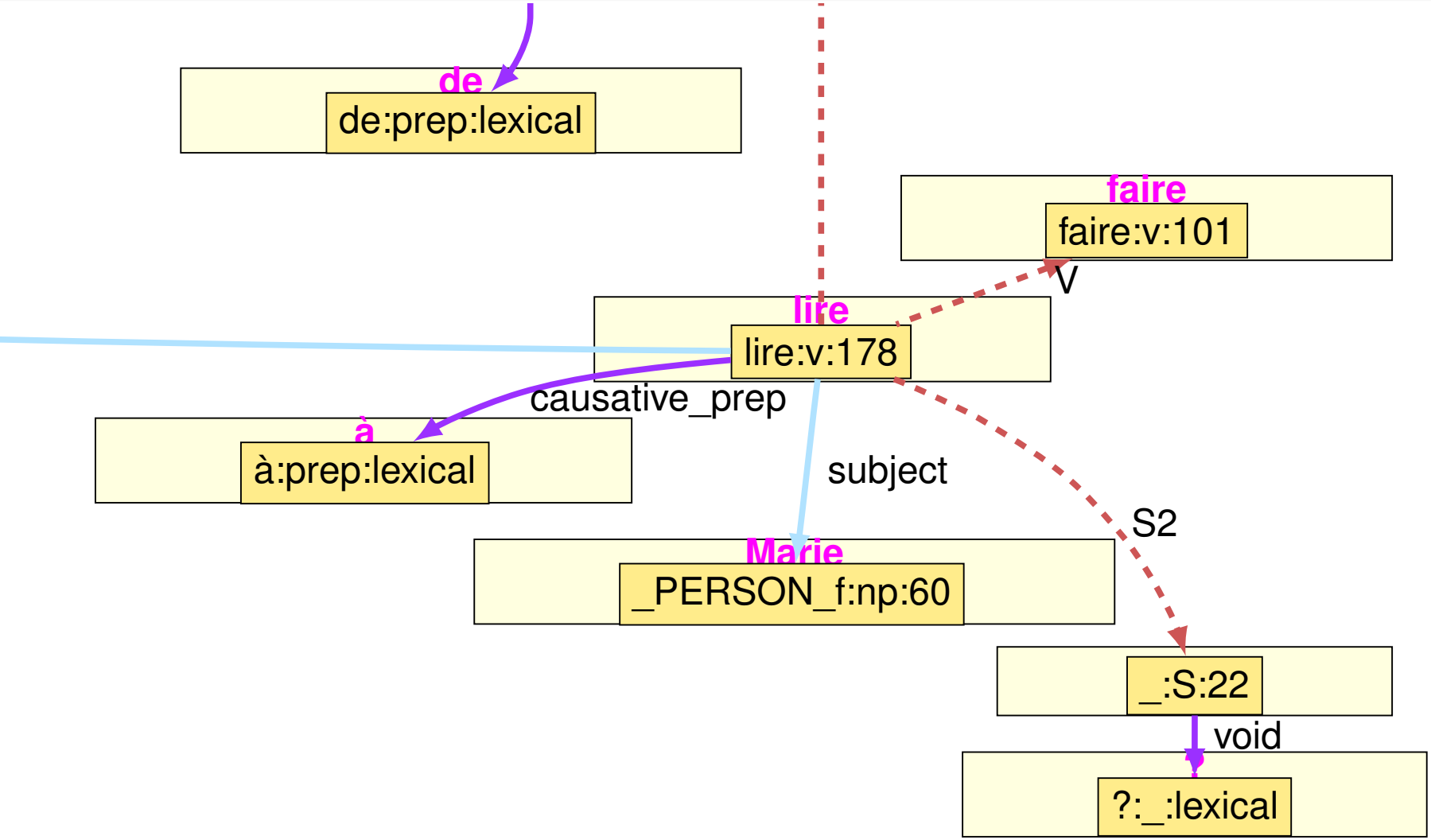
```
<?xml version="1.0"  encoding="ISO-8859-1"?>
<dependencies id="E1" mode="full">
  <cluster id="E1c_1_2" left="1" right="2" token="soyons" lex="
    E1F2|soyons"/>
  <cluster id="E1c_2_3" left="2" right="3" token="imaginatifs"
    lex="E1F3|imaginatifs"/>
  <cluster id="E1c_5_6" left="5" right="6" token="déclare" lex=
    "E1F6|déclare"/>
  <node deriv="E1d10" xcat="comp" id="E1n13" cat="adj" tree="72
    " lemma="i                    ster="E1c_2_3" form="imagi          ifs"/>
  <node deriv="E1d104" xcat="S" id="E1n22" cat="v" tree="186"
    lemma="déclarer" cluster="E1c_5_6" form="déclare"/>
  <node deriv="E1d13" xcat="S" id="E1n7" cat="v" tree="198"
    lemma="être" cluster="E1c_1_2" form="soyons"/>

  <edge id="E1e029" source="E1n22" target="E1n18" type="lexical
    " label="subject">
    <d                names="E1d104" source_op="E1o5" target_op="E1o20"
    span="6 7"/>
  </edge>
  <edge id="E1e011" source="E1n007" target="E1n013" type="subst
```



Quel livre Paul propose-t-il (plus abruptment qu'elle) de faire lire à Marie ?

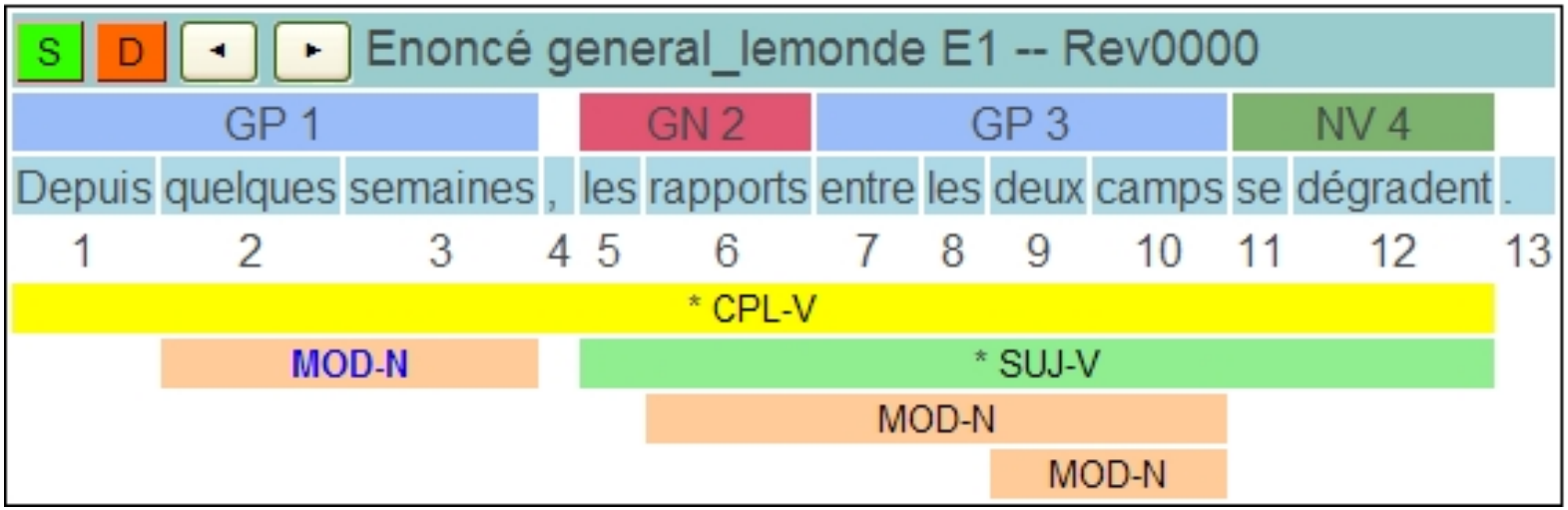


Quel livre Paul propose-t-il (plus abruptment qu'elle) de faire lire à Marie ?



Sorties EASy/Passage

- Possibilité de conversion vers les formats EASy et Passage
  (14 types de dépendances+ 6 types de chunks)
  format XML + vues graphiques



Désambiguisation

- algorithme de type 1-best en programmation dynamique, écrit en DyALog
- sommes de poids sur les dépendances
- poids fournis par des règles portant sur la dépendance et ses voisines
- poids manuellement définis
  quelques tentatives d'apprentissage
- temps de traitement du même ordre que pour l'analyse

```
%% Penalize inverted subjects
edge_cost_elem( '-INVERTED_SUBJ',
     edge{ label => subject,
           source => node{ cluster => cluster{ right => R } },
           target => node{ cluster => cluster{ left => L } }
         },
     -1000
  ) :- R =< L.
```

Outline

1 Concevoir

2 Utiliser

3 Améliorer

4 Exploiter

Améliorer FRMG

Les améliorations portent sur 4 axes

- augmenter la **couverture** en terme d'analyses complètes
  $\Longrightarrow$ méthodes non-supervisées sur large corpus: fouille d'erreurs
  $\Longrightarrow$ ouverture vers des techniques d'apprentissage

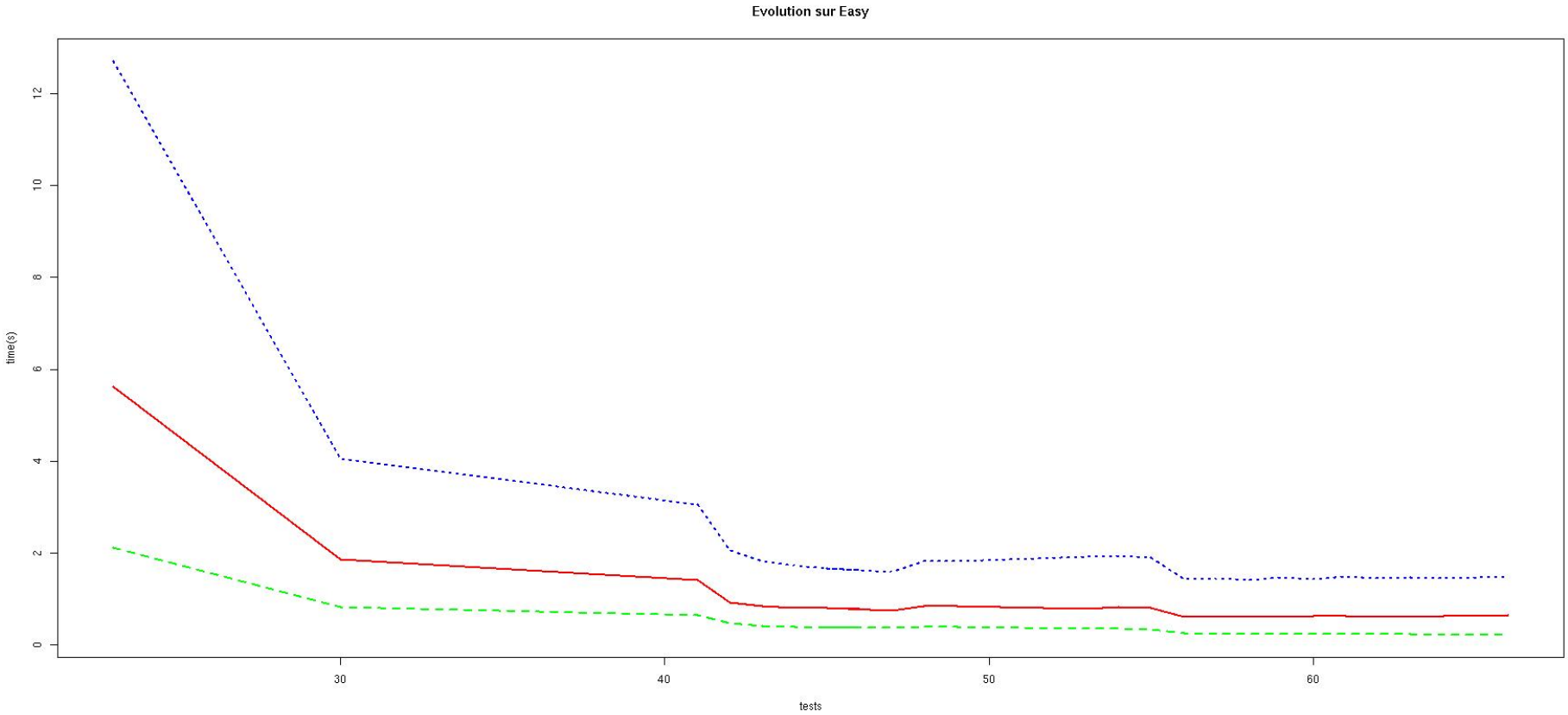
- améliorer la **qualité** des analyses
  $\Longrightarrow$ méthodes supervisées: utilisation d'une référence

- réduire le taux d'**ambiguïté** des analyses

- réduire les **temps d'analyse** et de désambiguisation.
  important mais pas essentiel: utilisation grille de machines (GRID5000)

Ces axes sont en partie contradictoires

Évolutions vitesse FRMG (2008)



Easydev: 3868 phrases, 58.20% couverture, timeout (20s): 0.4%

- Multiples runs sur jeux de tests et corpus, avec statistiques
- Test de nombreuses optimisations, la plupart inefficaces
- Mais gains instables: variations importantes en fonction de la grammaire

Améliorer la qualité

- Améliorer la qualité en exploitant des données de référence (treebank) données EASy
- résultats globaux, par type de corpus, groupes et relations
  log, évaluations, matrices de confusion
- visualisation des résultats
  Gestionnaire d'annotations avec service WEB EASYREF

Matrice de confusion (chunks et relations)

réf $\Longrightarrow$ hyp	B_GN	GN	E_GN	BE_GN	B_GP
B_GN	**5459 (91%)**	52 (0.88%)	14 (0.28%)	115 (1.94%)	118 (1.99%)
GN	50 (4.65%)	**725 (67%)**	149 (13.86%)	19 (1.77%)	12 (1.12%)
E_GN	4 (0.07%)	68 (1.15%)	**5345 (90.04%)**	140 (2.36%)	10 (0.17%)
BE_GN	106 (3.22%)	45 (1.37%)	78 (2.37%)	**2663 (80.97%)**	7 (0.21%)
B_GP	166 (2.02%)	30 (0.37%)	2 (0.02%)	30 (0.37%)	**7760 (94.61%)**

Nouveau: Matrices de différences entre runs.

Évolution des performances

	% analyse	Groupes			Relations		
	totales	prec.	rappel	f	prec.	rappel	f
R006 (05/07)	42.16%	78.12%	71.27%	74.54%	62.29%	46.63%	53.34%
R027 (12/07)	56.06%	83.66%	82.90%	83.28%	64.27%	55.66%	59.65%
R076 (02/09)	59.47%	84.23%	82.91%	85.56%	63.36%	55.62%	59.24%
R101 (06/09)	59.56%	83.24%	79.63%	81.40%	63.1%	53.40&	57.85%
R139 (09/09)	64.73%	87.41%	86.00%	86.70%	65.10%	59.03%	61.92%
R157 (10/09)	67.03%	87.71%	86.84%	87.28%	65.62%	60.26%	62.82%
R206 (01/10)	65.79%	87.92%	88.60%	88.26%	66.12%	62.13%	64.06%
R240 (11/10)	69.01%	88.24%	89.20%	88.72%	66.36%	63.32%	64.81%

Campagne	f-mesure groupes	f-mesure relations
2004	69%	41%
2007	89%	63%

Note: Nos outils d'évaluation donnent des valeurs plus faibles que les valeurs officielles.

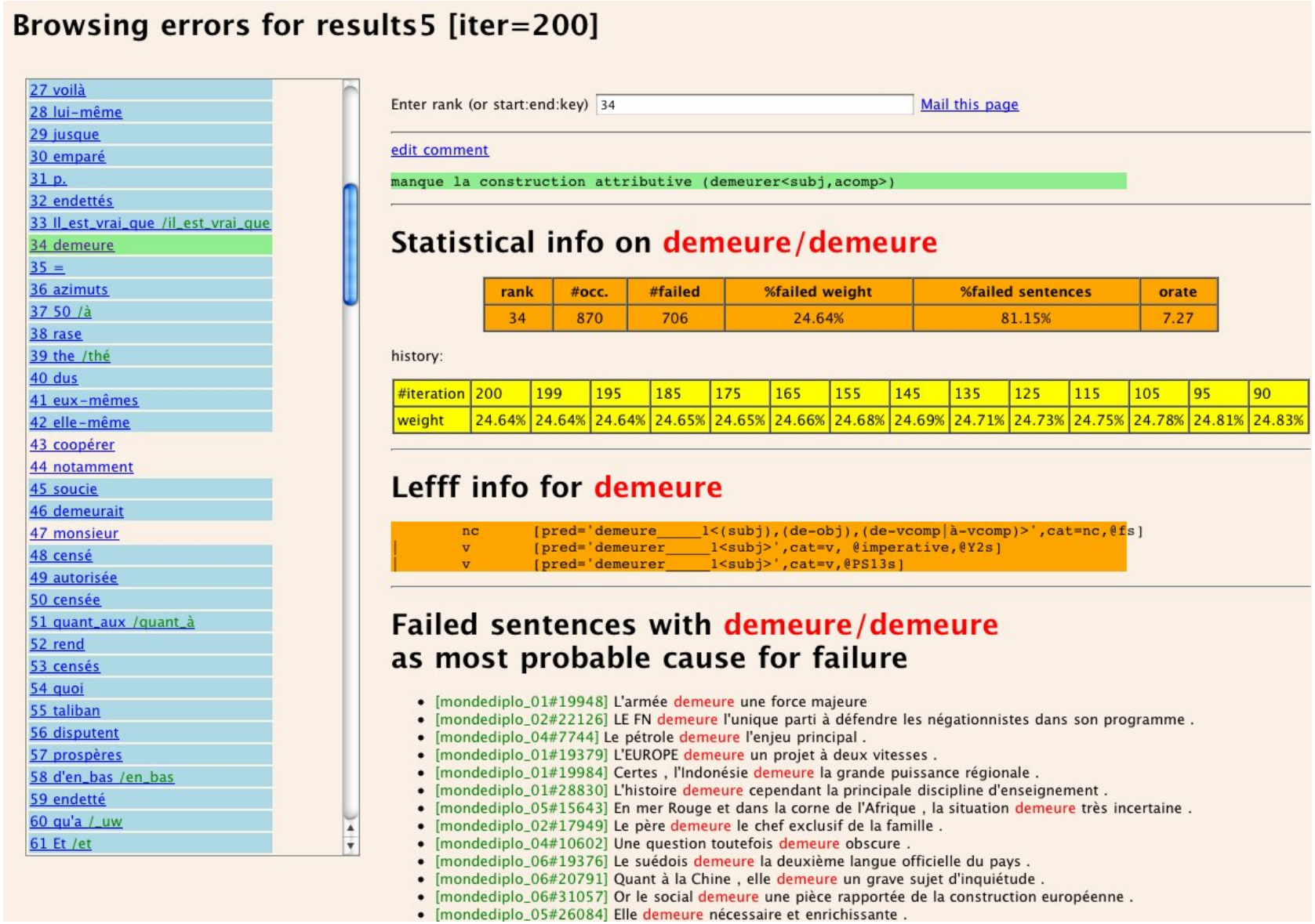
Couverture

Jeux	#phrases	Couv.	t. moy. (s)	amb.	couv. 09/09
EUROTRA	334	100%	0.09	0.81	10/10
TSNLP	1661	95.18%	0.04	0.48	11/10
EasyDev	3879	69.01%	0.46	1.10	11/10
JRCacquis	1.1M	51.26%	1.41	1.1	59.46%
Europarl	0.8M	70.19%	1.69	1.36	78.33%
EstRep	1.6M	67.05%	0.69	0.92	75.06%
Wikipedia	2.2M	69.11%	0.49	0.87	79.48%
Wikisource	1.5M	61.08%	0.71	0.89	66.79%
AFP	1.6M	52.15%	0.51	1.06	

Fouille et correction d'erreurs

- Pour améliorer la couverture: recherche des manques dans le lexique, la grammaire, ...
- traitement de gros corpus, suivi de fouille d'erreurs
  - identifier les mots trop souvent présents dans des phrases non analysables
  - surtout si en co-occurrence avec des mots sans problèmes
  - processus itératif de type EM
  - fournit des phrases où le mot est le principal suspect
- possibilité de suggérer des corrections:
  - ré-analyse les phrases en sous-spécifiant le suspect
  - fait ressortir les analyses les plus fréquentes devenues possibles au niveau du suspect

Interface WEB de fouille d'erreurs



Browsing errors for results5 [iter=200]

27 voilà
28 lui-même
29 jusque
30 emparé
31 p.
32 endettés
33 Il_est_vrai_que /il_est_vrai_que
34 demeure
35 =
36 azimuts
37 50 /à
38 rase
39 the /thé
40 dus
41 eux-mêmes
42 elle-même
43 coopérer
44 notamment
45 soucie
46 demeurait
47 monsieur
48 censé
49 autorisée
50 censée
51 quant_aux /quant_à
52 rend
53 censés
54 quoi
55 taliban
56 disputent
57 prospères
58 d'en_bas /en_bas
59 endetté
60 qu'a /_uw
61 Et /et

Enter rank (or start:end:key) 34 Mail this page

edit comment

manque la construction attributive (demeurer<subj,acomp>)

Statistical info on demeure/demeure

rank	#occ.	#failed	%failed weight	%failed sentences	orate
34	870	706	24.64%	81.15%	7.27

history:

#iteration	200	199	195	185	175	165	155	145	135	125	115	105	95	90
weight	24.64%	24.64%	24.64%	24.65%	24.65%	24.66%	24.68%	24.69%	24.71%	24.73%	24.75%	24.78%	24.81%	24.83%

Lefff info for demeure

```
        nc      [pred='demeure_____1<(subj),(de-obj),(de-vcomp|à-vcomp)>',cat=nc,@fs]
|       v       [pred='demeurer_____1<subj>',cat=v, @imperative,@Y2s]
|       v       [pred='demeurer_____1<subj>',cat=v,@PS13s]
```

Failed sentences with demeure/demeure as most probable cause for failure

- [mondediplo_01#19948] L'armée demeure une force majeure
- [mondediplo_02#22126] LE FN demeure l'unique parti à défendre les négationnistes dans son programme .
- [mondediplo_04#7744] Le pétrole demeure l'enjeu principal .
- [mondediplo_01#19379] L'EUROPE demeure un projet à deux vitesses .
- [mondediplo_01#19984] Certes , l'Indonésie demeure la grande puissance régionale .
- [mondediplo_01#28830] L'histoire demeure cependant la principale discipline d'enseignement .
- [mondediplo_05#15643] En mer Rouge et dans la corne de l'Afrique , la situation demeure très incertaine .
- [mondediplo_02#17949] Le père demeure le chef exclusif de la famille .
- [mondediplo_04#10602] Une question toutefois demeure obscure .
- [mondediplo_06#19376] Le suédois demeure la deuxième langue officielle du pays .
- [mondediplo_06#20791] Quant à la Chine , elle demeure un grave sujet d'inquiétude .
- [mondediplo_06#31057] Or le social demeure une pièce rapportée de la construction européenne .
- [mondediplo_05#26084] Elle demeure nécessaire et enrichissante .

Outline

1 Concevoir

2 Utiliser

3 Améliorer

4 Exploiter

Extraire des informations: SAPIENS

Qu'a déclaré Nicolas Sarkozy au sujet des femmes pendant la campagne présidentielle 2007 ? Et Ségolène Royal ? Et François Bayrou ?

La réponse avec SAPIENS, une production ALPAGE et SCRIBO pour AFP

Algorithmes et outils libres pour l'annotation semi-automatique et collaborative de documents numériques

`http://www.scribo.ws/`

Agence France-Presse — Caisse nationale des associations familiales
Dominique Voynet — François Bayrou — François Fondard
Françoise Laurent — Janine Mossuz-Lavau — Jean-Marie Le Pen — José Bové
Marie-George Buffet — Marine Le Pen — Michelle Perrot
Nicolas Sarkozy — Olivier Besancenot — PARIS
Ségolène Royal — Thérèse Clerc

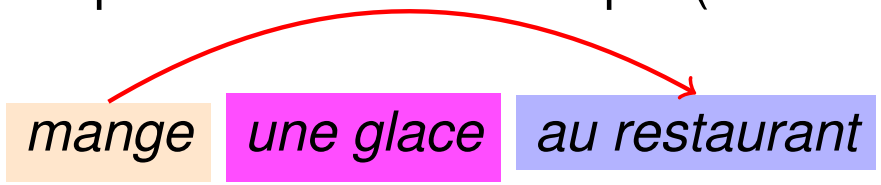
AFP, 2007-05-01
"Le temps des femmes est venu!" clame un millier de manifestant(e)s à Paris
PARIS, 1 mai 2007 (AFP) - Plusieurs centaines de femmes et d'hommes se sont rassemblés mardi Place de la Bastille à Paris à l'appel d'un collectif d'association pour les droits des femmes, pour s ...

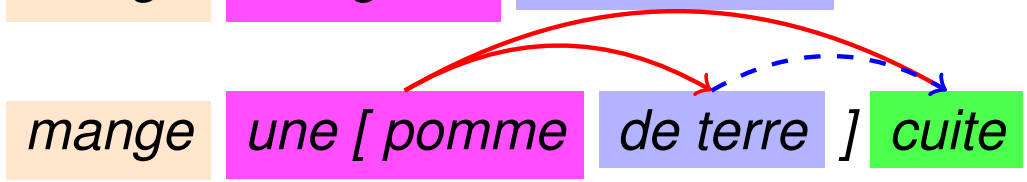
Nicolas Sarkozy a affirmé mardi que les lois sur la contraception ou l'interruption volontaire de grossesse n'avaient pas pour origine les événements de mai 1968. *"La loi sur la contraception a été votée en 1967 par Lucien Neuwirth"* et celle de *"l'interruption volontaire de grossesse, c'est 1974!"*, s'est-il exclamé.

Des dépendances: pour quoi faire ?

Les dépendances sont un bon de point de départ pour l'extraction d'information

- elles indiquent *qui* fait *quoi*, *quand*, *où*, plus d'autres rôles
- mais beaucoup de problèmes d'ambiguïtés,
  beaucoup au niveau sémantique (lexicale)





- *chemins de dépendances caractéristiques*

$\Longrightarrow$ besoin de connaissance sur le monde

- utilisation de resources structurées existantes (ontologies)
- par apprentissage de régularité d'usage à partir de larges corpus
  $\Longrightarrow$ hypothèse distributionnelle de Harris: *des mots sémantiquement proches apparaissent dans des contextes similaires.*

Les corpus

Les expériences en cours dans ALPAGE utilisent les corpus suivants:

corpus	occ.	dépendances	mots	contextes
AFP (30mois)	216M	92.7M	378K	2M
all (AFP+reste)	711.8M	220M	1.3M	3.7M

- AFP News (2007, 2009, 6 mois 2010)
- Wikipedia français
- Wikisource français
- Est Republicain (journalistique)
- Euro Parliament (transcription de discours)
- JRC Acquis Communautaires (directives européennes)

Analyse syntaxique effectuée sur quelques centaines de coeurs sur quelques jours (sur GRID5000).

Expériences

Des motifs sont collectés et comptés à partir des résultats d'analyse (format DepXML ou Passage) et utilisés dans 2 grandes directions:

Concepts

- Extraction de terminologie
- Construction de réseau de mots
- Regroupement de mots en cluster (*synset*), plus regroupement hiérarchique
- extraction de relations ontologiques (par ex. hypéronymie)

Évènements

- Regroupement de verbes, dénotant un type d'évenement
  - `/transfer/` *donner*, *offrir*, *céder*
  - `/communication act/` *annoncer*, *indiquer*, *affirmer*
- Identification de paires reliées verbe-nom
  - *déclarer*/*déclaration* ;
  - *identifier*/*identification* ;
  - *commencer*/*commencement*/*début*
- Découverte de chemins de dépendances caractéristiques entre des paires d'entités d'un certain type

Extraction de terminologie

Extraction de séquences de mots fréquentes (`N prep N`, `N Adj`, ...)
avec une forte information mutuelle + forte autonomie + variants
$\rightsquigarrow$ 110Kterms pour AFP, 120Kterm pour FrWiki

```
conférence de presse {:GN conférence/nc GN:}__{:GP de/prep presse/nc GP:}  37056
```

Regroupement de mots

Objectif: repérer la proximité sémantique des mots, sous forme de cluster ou de réseau de mots.

Le point de départ est l'hypothèse distributionnelle de Harris:

> *Les mots sémantiquement proches apparaissent dans des contextes similaires.*

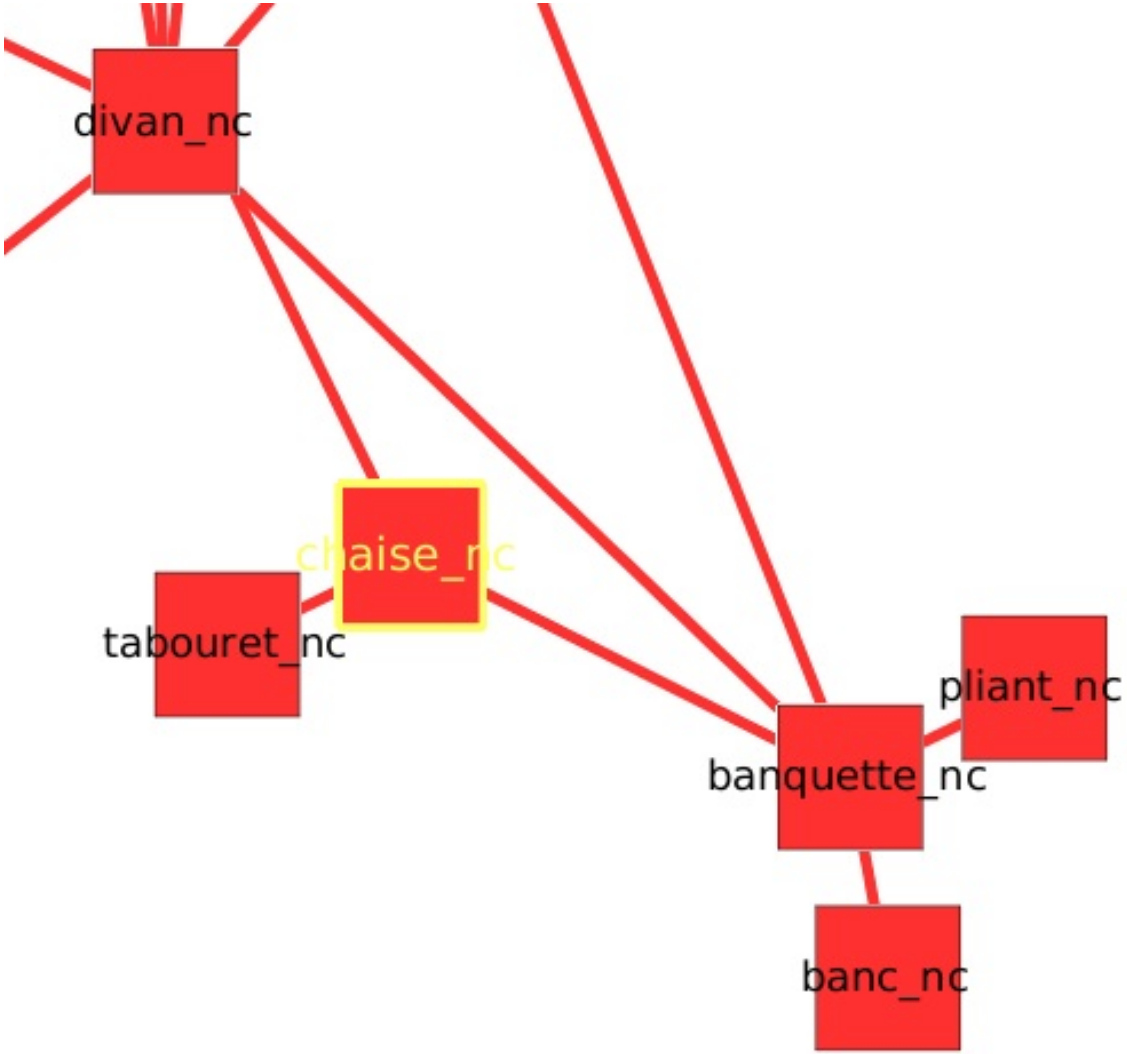
Différentes notions de contextes sont possibles:

- sac de mots (phrases, paragraphe)
- fenêtre de mots
- **contextes syntaxiques** sous forme de dépendances

Collecter et compter les dépendances (Passage)

<governor>	<rel>	<governee>	<freq>
chaise_nc	et	table_nc	235
asseoir_v	sur	chaise_nc	227
chaise_nc	modifieur	long_adj	168
chaise_nc	de=	poste_nc	115
tomber_v	sur	chaise_nc	103
chaise_nc	modifieur	musical_adj	102
se_asseoir_v	sur	chaise_nc	93
prendre_v	cod	chaise_nc	87
chaise_nc	modifieur	électrique_adj	82
chaise_nc	modifieur	vide_adj	80
chaise_nc	à=	porteur_nc	80
dossier_nc	de	chaise_nc	78
avoir_v	cod	chaise_nc	71
table_nc	et	chaise_nc	62
chaise_nc	de=	paille_nc	56

Autour d'une chaise



À quoi sert une chaise ?

Il est important de connaître tout ou partie des traits ayant motivé un regroupement.

	chaise divan	chaise tabouret	banquette divan	banquette canapé	banquette chaise	banquette banc
se asseoir sur [•]	●	●	●	●	●	●
asseoir sur [•]	●	●	●	●	●	●
allonger sur [•]	●		●	●		
dormir sur [•]	●		●	●	●	
tomber sur [•]	●		●	●	●	
monter sur [•]		●			●	
place sur [•]						●
grimper sur [•]		●			●	
installer sur [•]		●			●	
poser sur [•]		●			●	
coucher sur [•]	●		●	●		
siéger sur [•]						●
côté sur [•]			●	●		
se lever cpl de [•]	●					
se affaisser sur [•]	●					
jeter sur [•]			●	●		
être sur [•]						●
se installer sur [•]						●
retomber sur [•]	●					
endormir sur [•]				●		
se soulever sur [•]			●			

Désambiguisation “sémantique”

- Premières expériences d’utilisation de connaissances acquises pour la désambiguisation
- utilisation des contextes significatifs pour les attachements prépositionnels, sujets, objets, et autres catégories/fonctions.
- expansion des contextes via les proximités entre mots
- $\Longrightarrow$ attachements corrects sur des phrases comme
  - *il mange* *une tarte* *de sa mère* *aux fruits*
  - *il mange* *une tarte* *de la mère* *à son amie*
- Des gains, mais peu importants ! À suivre . . .

Conclusion

- possible d'exploiter à grande échelle une grammaire TAG non stochastique à large couverture
  développement facilité grâce aux méta-grammaires et à la factorisation
- une chaîne de traitement linguistique n'est jamais achevée
  $\Longrightarrow$ besoin d'outils de contrôle de plus en plus fin avec forte intégration
  $\Longrightarrow$? travail communautaire : initiative **mg2wiki**
- Outils et ressources librement disponibles
  installateur ALPI mais manque de documentation
- Existence d'une petite communauté d'utilisateurs autour de FRMG
  et développement de SPMG pour l'espagnol
- Quelques démos possibles:
  frmg_shell, Sapiens, Error Mining, Term View, EasyRef, réseau de mots