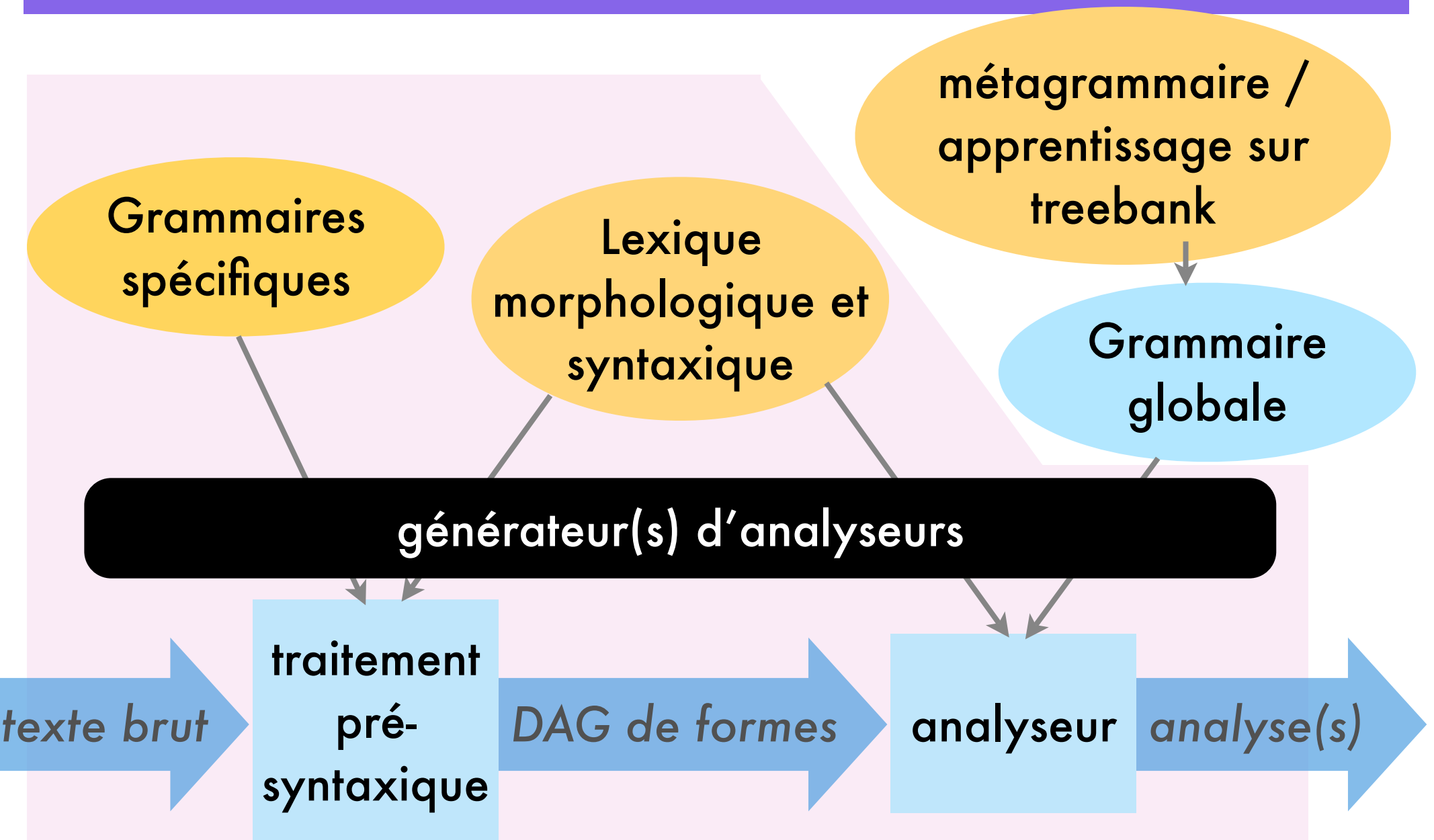
SxPipe, le Le*fff*, et tout ce dont on a besoin pour construire des systèmes complets d'analyse

Benoît Sagot

Séminaire de lancement de l'équipe Alpage
Jeudi 15 novembre 2007

Introduction



Plan

1. Traitement pré-syntaxique : SxPipe
2. Lexique morphologique et syntaxique : le L*efff* et ses petits frères
3. Vers la construction d'un système d'analyse complet
4. Conclusions

1. Traitement pré-syntaxique : SxPipe

Idée générale

- On cherche à analyser des **corpus bruts**
- Les analyseurs syntaxiques prennent en entrée des **suites (ou des DAG) de formes**
  - une forme est une entité syntaxiquement atomique
- Il faut donc savoir passer de l'un à l'autre

Formes, tokens, énoncés

- Un corpus brut est un flux de caractères
- Il faut donc découper ce flux
  – en unités typographiques raisonnables (**tokens**)
  – en éléments analysables individuellement (**énoncés**)
- puis identifier les **formes** correspondant aux tokens (forme = entrée du lexique utilisé)

tokens ↔ formes

- **Pas de correspondance biunivoque** entre formes et tokens
- **Ambiguïtés** de correspondance
- Formes spéciales pour les séquences de tokens formant des **entités nommées**
  - une entité nommée est une séquence de tokens respectant une certaine **grammaire locale** mais qui est **syntaxiquement atomique**

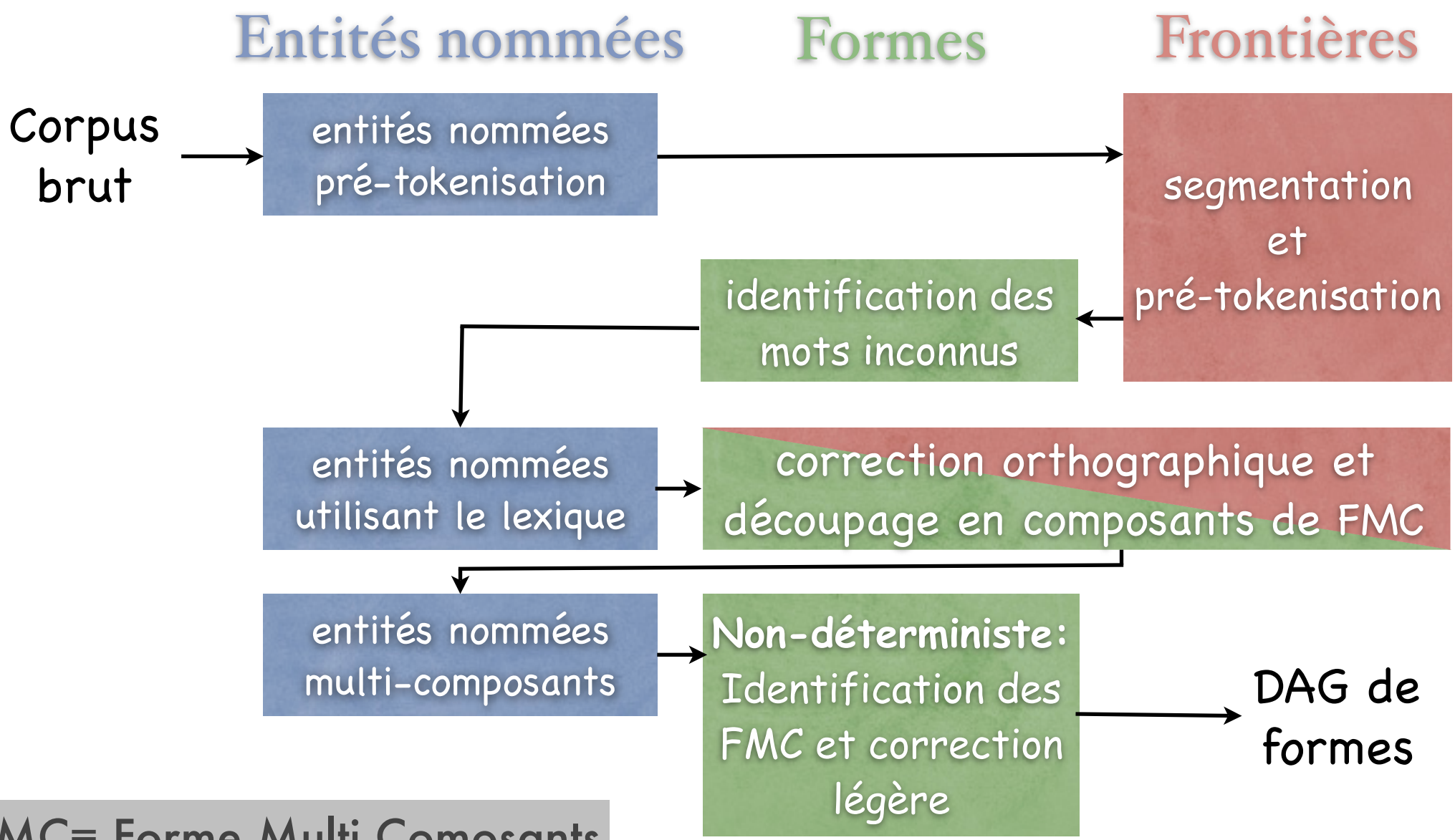
tokens ↔ formes

tokens	formes
au	à le
du	(du | de le)
séduc tion	séduction
joliefille	jolie fille
d’ ores et déjà	d’ores_et_déjà
à l’instar du	à_l’instar_de le
aujourd’ hui	aujourd’hui
habiter 2, rue la Paix - 75008	habiter _ADRESSE

SxPipe

- **SxPipe**, utilisé depuis 2 ans, est une chaîne de traitement pré-syntaxique complète
- Langues traitées
  - **français**
  - polonais
  - versions préliminaires pour l'anglais, le slovaque, le slovène

SxPipe 1



FMC= Forme Multi Comosants

SxPipe 2

- **Nouvelle version** en cours de développement
- 3 nouveautés
  1. **paramétrisation dynamique**, y compris détection automatique de la langue
  2. **correction orthographique** et découpage en formes (dont FMC) couplées et **ambiguës**
  3. reconnaissance ambiguë de **motifs CFG**

Paramétrisation dynamique

- Idée générale : le texte lui-même induit des **modifications des paramètres** de SxPipe
  - **modifications dynamiques** (phrase par phrase) : langue, présence ou non de diacritiques (phrases en majuscules...)
  - **modifications statiques** (passe préliminaire sur tout le document) : détection et adaptation au type de texte, noms propres et vocabulaire spécifiques…

Correction ambiguë

- Correction orthographique + découpage en formes sont désormais en un seul module **text2dag**, qui produit une sortie ambiguë

text (tokens)	dag (formes)
cest	c' est
apeller	appeler
segmen tation	segmentation
PELE	(pelé | pèle)
frere	(frère | ferre)
tere cuite	(terre_cuite | (ter | terre) cuite)

Motifs CFG

- Idée générale : on veut reconnaître de façon **ambiguë**, dans des **entrées ambiguës (DAG)**, des motifs
  - pour identifier des entitées nommées ou baliser des portions de textes
  - pour étiqueter certains mots dans le texte
- La puissance des CFG est parfois nécessaire
- Elle n'est pas vraiment un handicap dans le cas où elle est superflue

Motifs CFG

- Reconnaissance de motifs CFG:
  - Analyseur Earley du système **SYNTAX**
  - En-têtes de grammaires prédéfinis pour n'avoir à décrire que les motifs
  - Possibilité d'utiliser un **lexique**
  - Entrée : DAG
  - Sortie : DAG (frontière de la forêt partagée)
  - Paramétrages pour la désambiguïsation

Motifs CFG

- Déjà utilisé :
  - **Grammaire des nombres** écrits en toute lettres (par rapport à la même grammaire en perl, on gagne l'ambiguïté… et un facteur 10 en temps!)
  - Système **unitex2sxpipe** de conversion de graphes développés sous Unitex (Antoine Désir), avec une première application à **Ilimp** (Laurence Danlos)

SxPipe : bilan

- Le pré-traitement est crucial pour l'analyse syntaxique, symbolique ou probabiliste
  - **robustesse** : tout corpus doit être analysable, pas seulement la partie évaluation d'un treebank…
  - **guidage** : étiquetage ultra-lexicalisé des *il* impersonnels (bientôt *que*, *de*,… ?)
- Les entités nommées peuvent constituer une fin en soi (expressions temporelles…)

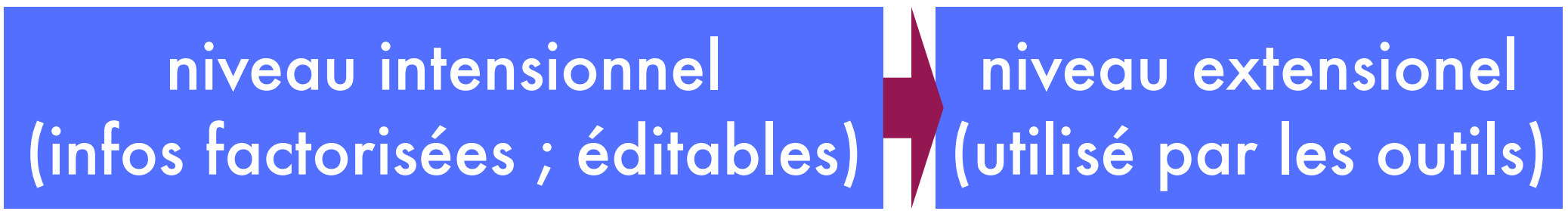
2. Lexique morphologique et syntaxique : le L*efff* et ses petits frères

Le Le*fff*

- Le **Le*fff*** est un **lexique électronique du français, à large couverture et librement disponible** (www.lefff.net)
  - > 110 000 lemmes, > 530 000 entrées
- Il donne des informations **morphologiques** (lemme, étiquette) et **syntaxiques** (cadre de sous-catégorisation)
- **Indépendant du formalisme** (utilisateurs TAG, LFG,…)

Alexina

- Le Le*fff* est développé dans le cadre de l'architecture Alexina
  - outils et technologies d'acquisition automatique d'informations lexicales (et bientôt interfaces d'édition de lexiques)
  - architecture à deux niveaux



Historique

- 2003 : pas de lexique TAL du français qui soit à large couverture et librement disponible → lancement du projet Le*fff*
- **Le*fff* 1** : lexique morphologique des verbes du français
  - acquis automatiquement puis validé
- **Le*fff* 2** : lexique morphologique et syntaxique, toutes catégories confondues

Actualités

- Depuis quelques temps, travaux (avec Laurence Danlos, Karën Fort…) de **comparaison/fusion** avec d'autres ressources (LG, Dicovalence), notamment :
  - constructions impersonnelles
  - adverbes en *-ment*
- Le*fff* 3 en cours de développement : architecture simplifiée, **interfaces** de développement, notion de **construction**

Le Lefff 2, sur le départ

advenir v18 @plaire

```
@plaire
< verbe
< sujet_standard
< sujet_verbal_possible
< verbe_transitif_ind_à
< objet_indirect_à_cld
< impersonnel_possible
;
```

Table de flexion v18

verbe_transitif_ind_à	Objà	à-sn
…		
impersonnel_possible	Imp	+=Default
impersonnel_possible	ImpInfinitive	+=Infinitive
impersonnel_possible	ImpPastParticiple	+=PastParticiple

adviendra v [pred='advenir$_1$<Suj:sn | cln | sinf | scompl,Objà:(à-sn | cld)>', cat=v, @pers, @F3s]
advenir$_1$ Default F3s

adviendra v [pred='advenir$_1$<Suj:sn | de-sinf | scompl,Objà:(à-sn | cld)>', cat=v, @impers,@F3s]
advenir$_1$ Imp F3s

Le Lefff 3 en avant-première

advenir v18 100;Lemma;v;<Suj:cln|scompl|sinf|sn,Objà:(cld|à-sn)>; cat=v; %actif,%impersonnel

Table de flexion v18

%actif = Infinitive:{Suj ()}
+ PastParticiple:{Macros @active,@avoir}
+ {Macros @pers}
+ Imperative:{Macros @imperative}

%impersonnel = %actif
+ {Skip Imperative} + {Skip Default}
+ {@pers @impers}
+ {Suj <Suj[cln>,sinf>de-sinf]}

adviendra v [pred='advenir$_1$<Suj:sn|cln|sinf|scompl,Objà:(à-sn|cld)>', cat=v, @pers, @F3s]
advenir$_1$ Default F3s

adviendra v [pred='advenir$_1$<Suj:sn|de-sinf|scompl,Objà:(à-sn|cld)>', cat=v, @impers,@F3s]
advenir$_1$ Imp F3s

Autres langues

- Autres lexiques Alexina
  - **polonais** (morpho, à partir de corpus)
  - **slovaque** (préliminaire, morpho)
  - **anglais** (morpho, syntaxe envisagée)
  - slovène (test d'un processus d'adaptation à Alexina d'un lexique **Multext**)
  - **espagnol** (le développement du lexique syntaxique débute en ce moment même)

Lefff : bilan

- Un lexique morphologique et syntaxique riche est indispensable
  - à l'analyse symbolique (FRMG, SxLFG,…)
  - à l'amélioration/au guidage d'analyseurs probabilistes

3. Vers la construction d'un système d'analyse complet

SxLFG ?

- Oui, mais…
- SxLFG est un générateur d'analyseurs LFG qui repose sur le système SYNTAX
- Une grammaire du français a été développée
- Mais l'avenir n'est peut-être plus dans le développement manuel

SYNTAX !

- SYNTAX permet la génération d'analyseurs CFG, y compris PCFG (CFG probabilisées)
- Des travaux sont en cours pour l'utilisation du (nouveau et secret) treebank d'Anne Abeillé pour l'apprentissage de grammaires

SxPipe et le Le*fff* : le retour

- Mais on ne veut pas savoir analyser qu'une section évaluation dudit treebank
- SxPipe est indispensable à la robustesse
- ⇒ Travail en cours
  - traitement par SxPipe du treebank d'Anne, pour apprendre des grammaires à partir de corpus traités par SxPipe
  - l'analyseur résultant traitera la sortie de SxPipe

SxPipe et le L*efff* : le retour

- Le L*efff* est une source d'informations importante, qu'il faudra apprendre à exploiter dans un analyseur syntaxique
- Peut-être sous forme de repli, en cas de non-couverture d'un événement (mot, cadre) par le treebank
  - choix argument/modifieur,
  - rattachements à longue distance
  - etc…

Conclusion

L'avenir du futur

- L'analyse syntaxique doit se diriger vers des techniques mixtes symboliques/probabilistes
- SxPipe est indispensable en amont
  - passage d'un flux de caractère à un DAG de formes (y compris entités nommées)
  - guidage de l'analyseur : étiquetage de certains mots et parenthésage (typé) par grammaires spécifiques ultra-lexicalisées

L'avenir du futur

- Le L*efff* est indispensable aux analyseurs syntaxiques symboliques,
- et doit le devenir pour les analyeurs probabilistes → analyseurs mixtes

L'avenir du futur

- Autres chantiers
  - multilingue (SxPipe multilingue, lexiques mutiples au sein d'Alexina)
  - analyse sémantique (ressources à la Wordnet, WSD)
  - applications réelles (collaborations industrielles)