Talismane : construction d'un analyseur syntaxique probabiliste

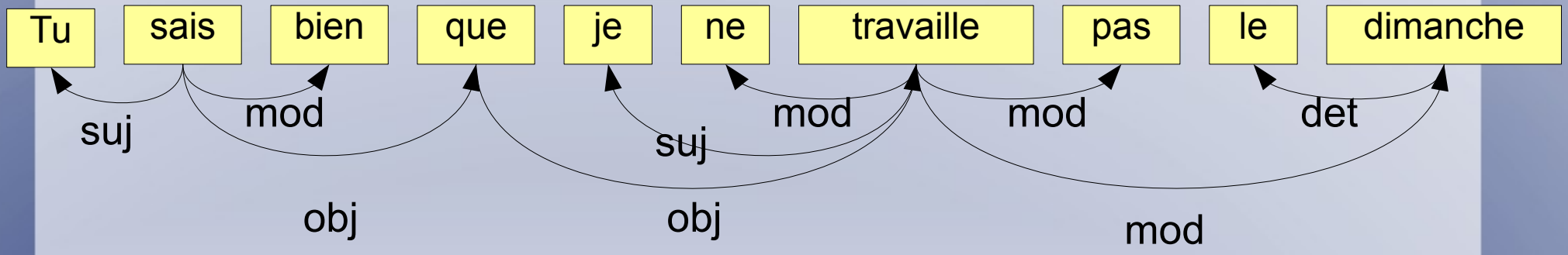
Assaf URIELI

Séminaire "Thématiques Actuelles du TAL"
Laboratoire CLLE-ERSS, Axe TAL
10 déc 2012

Analyse de syntaxe : problématique

- Analyse syntaxique automatique

  « Tu sais bien que je ne travaille pas le dimanche ! »



Analyse de syntaxe : problématique

- A partir du texte brute, niveaux croissants d'abstraction :
  - **Découpage en phrases**
  - **Tokenisation** : découpage en mots
  - **POS tagging** : attribution d'une catégorie grammaticale à chaque mot
  - **Parsing** : analyse des dépendances syntaxiques entre les mots

Apprentissage automatique

- **Corpus d'entraînement**
- **Corpus d'évaluation**
- **Traits**
  - Ressources externes
- **Classifieur probabiliste**

Apprentissage automatique : Talismane

- **Corpus d'entraînement :** French Treebank + FTBDep (80 %)
- **Corpus d'évaluation :** French Treebank + FTBDep (20 %)
- **Traits :** syntaxe de définition de traits
  - Ressources externes : LEFFF
- **Classifieur probabiliste :** OpenNLP MaxEnt
  (+SVM Linéaire, Perceptrons)

Corpus d'entraînement : French Treebank

<SENT nb="20">

<NP ***fct="SUJ"***><w cat="N" ee="N-C-ms" ei="NCms" lemma="m." mph="ms" subcat="C">**M.**</w> <w cat="N" ee="N-P-ms" ei="NPms" lemma="Teulade" mph="ms" subcat="P">**Teulade**</w> </NP>

<VN> <w cat="V" ee="V--P3s" ei="VP3s" lemma="pouvoir" mph="P3s" subcat="">**peut**</w> </VN><w cat="PONCT" ee="PONCT-W" ei="PONCTW" lemma="," subcat="W">**,**</w> <w compound="yes" cat="ADV" ee="ADV" ei="ADV" lemma="à juste titre"> <w catint="P">**à**</w> <w catint="A">**juste**</w> <w catint="N">**titre**</w> </w> <w cat="PONCT" ee="PONCT-W" ei="PONCTW" lemma="," subcat="W">**,**</w>

<VPinf ***fct="OBJ"***> <VN> <w cat="V" ee="V--W" ei="VW" lemma="considérer" mph="W" subcat="">**considérer**</w> </VN>

<Ssub ***fct="OBJ"***> <w cat="C" ee="C-S" ei="CS" lemma="que" subcat="S">**que**</w> <w cat="PONCT" ee="PONCT-W" ei="PONCTW" lemma=""" subcat="W">**"**</w>

<NP ***fct="SUJ"***> <w cat="D" ee="D-def-fs" ei="Dfs" lemma="le" mph="fs" subcat="def">**la**</w> <w cat="N" ee="N-C-fs" ei="NCfs" lemma="crédibilité" mph="fs" subcat="C">**crédibilité**</w> <PP> <w cat="P" ee="P" ei="P" lemma="de">**du**</w> <NP> <w cat="D" ee="D-def-ms" ei="Dms" lemma="le" mph="ms" subcat="def"></w> <w cat="N" ee="N-C-ms" ei="NCms" lemma="système" mph="ms" subcat="C">**système**</w> <AP> <w cat="A" ee="A-qual-ms" ei="Ams" lemma="conventionnel" mph="ms" subcat="qual">**conventionnel**</w> </AP> </NP> </PP></NP>

<VN> <w cat="V" ee="V--P3s" ei="VP3s" lemma="être" mph="P3s" subcat="">**est**</w> </VN><w compound="yes" cat="ADV" ee="ADV" ei="ADV" lemma="en jeu"> <w catint="P">**en**</w> <w catint="N">**jeu**</w> </w> <w cat="PONCT" ee="PONCT-W" ei="PONCTW" lemma=""" subcat="W">**"**</w>

</Ssub>

</VPinf>

<w cat="PONCT" ee="PONCT-S" ei="PONCTS" lemma="." subcat="S">**.**</w>

</SENT>

Corpus d'entraînement : FTBDep

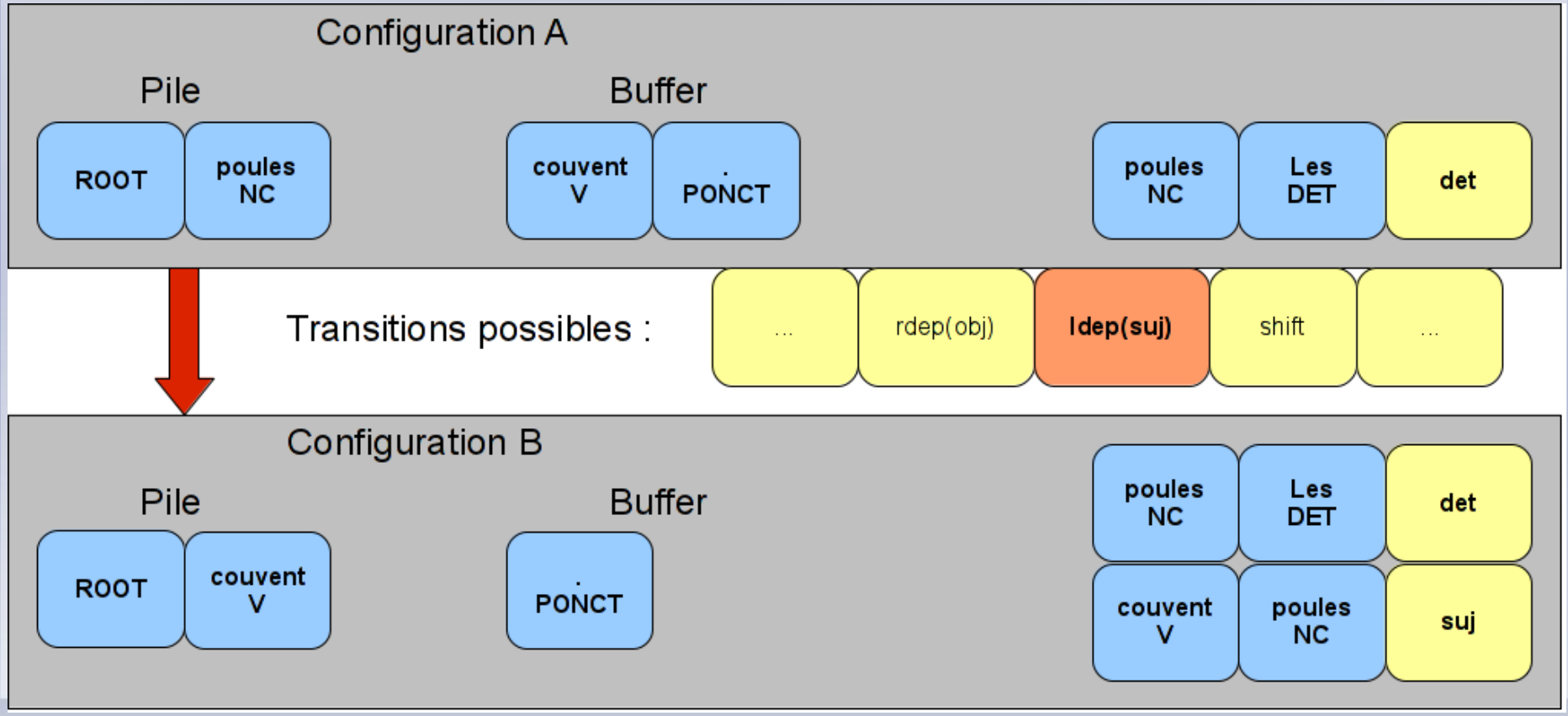
1	M.	m.	N	NC	g=m|n=s|s=c	3	suj
2	Teulade	Teulade	N	NPP	g=m|n=s|s=p	1	mod
3	peut	pouvoir	V	V	m=ind|n=s|p=3|t=pst	0	root
4	,	,	PONCT	PONCT	s=w	3	ponct
5	à_juste_titre	à_juste_titre	ADV	ADV	_	3	mod
6	,	,	PONCT	PONCT	s=w	3	ponct
7	considérer	considérer	V	VINF	m=inf	3	obj
8	que	que	C	CS	s=s	7	obj
9	"	"	PONCT	PONCT	s=w	15	ponct
10	la	le	D	DET	g=f|n=s|s=def	11	det
11	crédibilité	crédibilité	N	NC	g=f|n=s|s=c	15	suj
12	du	de	P+D	P+D	s=def	11	dep
13	système	système	N	NC	g=m|n=s|s=c	12	obj
14	conventionnel	conventionnel	A	ADJ	g=m|n=s|s=qual	13	mod
15	est	être	V	V	m=ind|n=s|p=3|t=pst	8	obj
16	en_jeu	en_jeu	ADV	ADV	_	15	mod
17	"	"	PONCT	PONCT	s=w	15	ponct
18	.	.	PONCT	PONCT	s=s	3	ponct

Transformation en classification

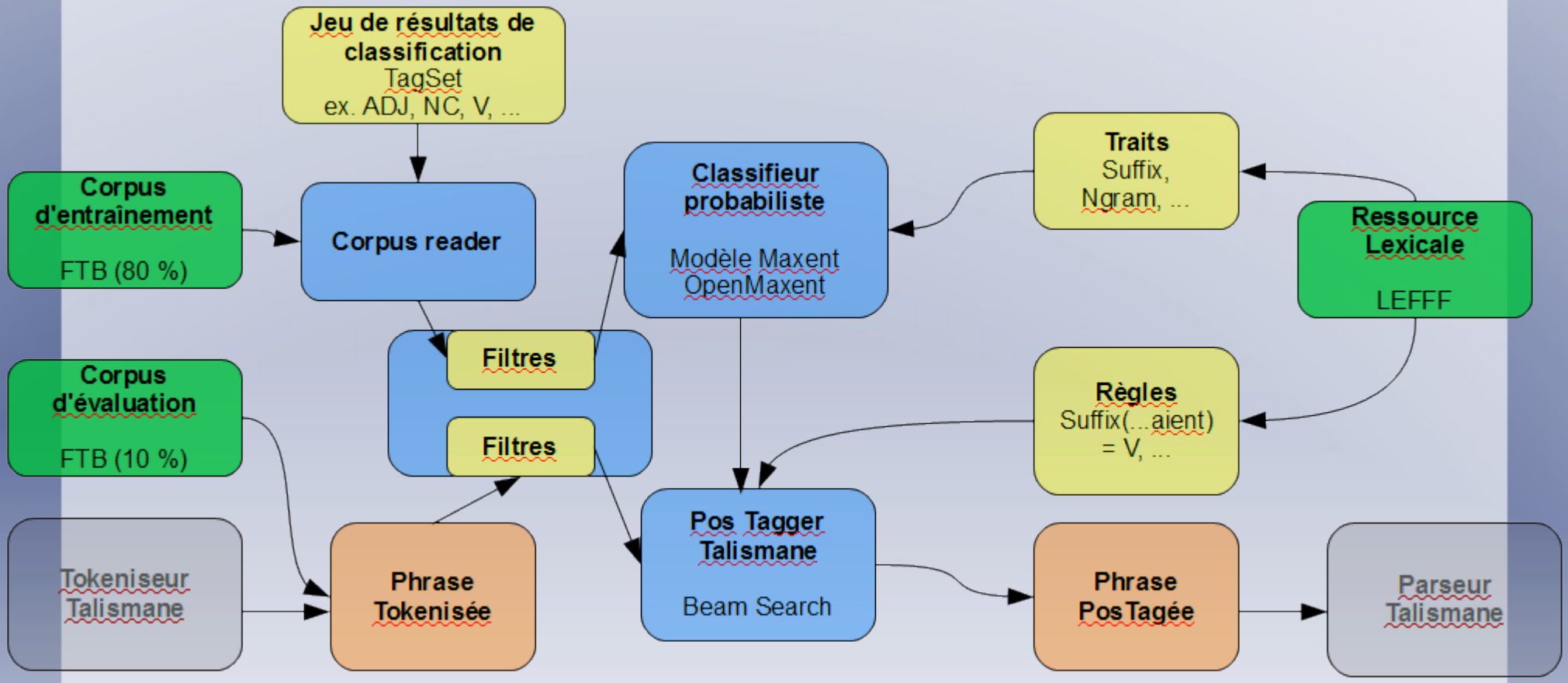
- Découpage en phrases :
  - Pour chaque symbole . ? ! » … Classification binaire : oui/non
- Tokenisation
  - Transformer une phrase en « tokens atomiques »
    - A-t-elle mangé de l'avoine ?
    - [A][-][t][-][elle][ ][mangé][ ][de][ ][l]['][avoine][ ][?]
    - [A][-t-elle][ ][mangé][ ][de l'][avoine][ ][?]
  - Pour l'intervalle entre chaque paire de tokens atomiques, classification binaire : séparation ou non ?
  - Liste de motifs (« patterns ») pour réduire l'espace de calcul

Transformation en classification

- Pos tagging :
  - Problème de classification « classique »
- Parsing : « Les poules couvent. »



Architecture Talismane (ex. Pos tagger)



Configuration : les Filtres

- Pour filtrer le flux d'entrée
- Filtres du texte brut :
  - Pour sauter un passage, remplacer un passage, insérer une coupure de phrase, ...
  - ex. « Ecris-moi à l'adresse assaf.urieli@univ-tlse2.fr. »
  - = « Ecris-moi à l'adresse AdresseMail. »
- Filtres du texte tokenisé :
  - Pour insérer des tokens vides (ex. après « du », avant « duquel »)

Configuration : les Traits

- Un « événement » dans le corpus d'entraînement existe pour Talismane comme une **classe** + un **vecteur de traits**.
- Exemple (POS tagger) : Il **chantait**.
  - Classe : Verbe
  - Trait 1 : suffixe(3) = « ait »
  - Trait 2 : préfixe(5) = « chant »
  - Trait 3 : le LEFFF le considère comme verbe imparfait 3 pers sing
  - Trait 4 : le pos-tag du token précedent = CLS (clitique sujet)
  - …
- Appliqués pendant l'entraînement **et** l'analyse

Configuration : les Traits

- Exemples pour le pos-tagger

Suffix	NLetterSuffix(IndexRange(2,5))
FirstLetterCaps	OnlyTrue(Regex("[A-Z][^A-Z].*"))

- Exemples pour le parseur

IsAConjugatedVerb	PosTag()=="V" | PosTag()=="VS" | PosTag()=="VIMP" | PosTag()=="VPP"
LemmaLastVerb	Lemma(BackwardSearch(Stack[0],IsAConjugatedVerb()))

Apprentissage automatique : résultat

- En entrée : corpus d'entraînement, traits, classifieur
- En sortie : le « modèle statistique » = une grande matrice de [traits x classes]

	ADJ	ADV	NC	...
suffix « ait »	0.02	0.01	0.12	...
ngram(2) DET	0.30	0.02	0.43	...
Lefff ADJ	0.64	0.12	0.21	...
...	...	...	...	...

Configuration : les Règles

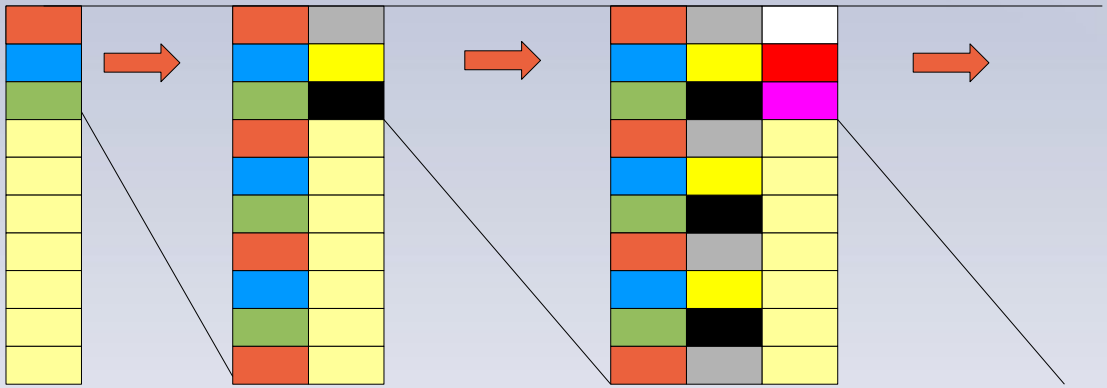
- Appliquées uniquement pendant l'analyse, pour restreindre ou remplacer le modèle probabiliste

Classes fermées	!CLS	Not(LexiconPosTag("CLS"))
Filtres	NPP	Word("AdresseMail")

Analyse : Recherche en faisceau

- La recherche en faisceau (beam search) permet de réduire l'espace de recherche.
- Après chaque étape, nous gardons uniquement les *n* analyses les plus probables comme base de l'étape suivante.

beam width =3



Recherce en faisceau : évaluation

- Jeu « dev » (+ exemples spécifiques)

	beam width 1			beam width 10		
type	**préc**	**rappel**	**f-score**	**préc**	**rappel**	**f-score**
ato	83.3	33.3	47.6	**87.5**	**46.7**	60.9
aux_pass	97.3	89.8	93.4	97.9	**94.3**	96.0
aux_tps	99.4	93.9	96.6	99.2	**96.5**	97.8
coord	100.0	52.6	69.0	100.0	**56.3**	72.0
dep_coord	99.3	80.3	88.9	99.4	**83.2**	90.6
mod_rel	98.7	67.7	80.3	99.2	**70.4**	82.3
root	100.0	83.9	91.2	100.0	**90.0**	94.8
suj	98.2	86.7	92.1	98.1	**90.3**	94.0
total			87.7			**89.1**
temps (ms/token)			**9.6**			90.3

Recherce en faisceau : évaluation

- Jeu « test »

	beam width 1			beam width 10		
type	**préc**	**rappel**	**f-score**	**préc**	**rappel**	**f-score**
ato	**85.7**	**50.0**	63.2	83.3	41.7	55.6
aux_pass	97.6	90.9	94.1	97.2	**94.6**	95.9
aux_tps	98.7	94.9	96.8	99.0	**97.3**	98.2
coord	97.4	50.9	66.9	97.5	**53.4**	69.0
dep_coord	99.9	81.0	89.5	99.8	**83.5**	90.9
mod_rel	99.6	65.2	78.8	99.2	**68.1**	80.8
root	100.0	85.4	92.1	100.0	**90.8**	95.2
suj	98.7	87.4	92.7	98.6	**90.2**	94.2
total			88.1			**89.5**
temps (ms/token)			**9.4**			89.6

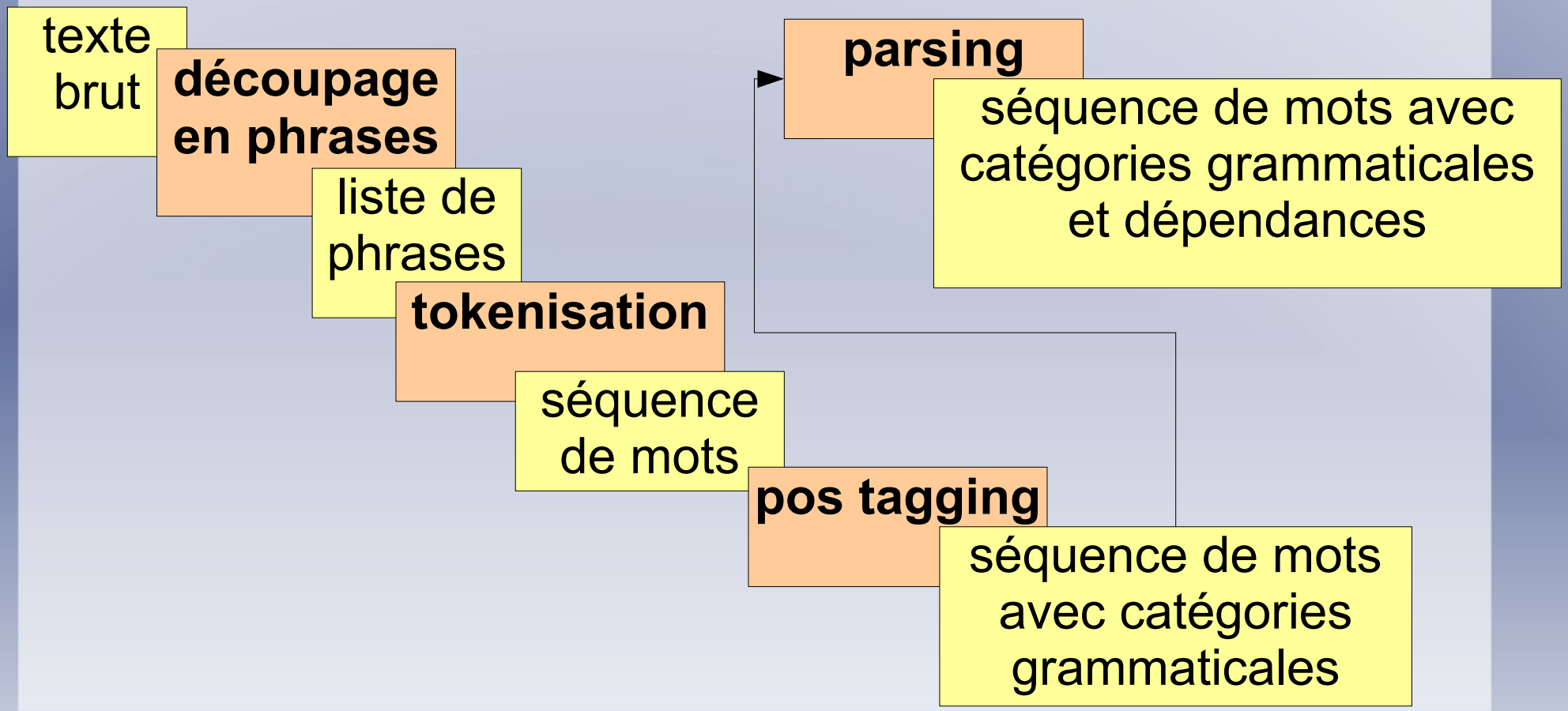
Evaluation : Comparaison

- Comparaison avec Candito, Nivre, Denis et Henestroza 2010

Parser	dev	test	time
Berkeley	86.5	86.8	12:46
MSTParser	87.5	**88.2**	14:39
MaltParser	86.9	87.3	**1:25**
Talismane beam 1	**87.7**	88.1	6:03
Talismane beam 10	**89.1**	**89.5**	56:50

Analyse en Cascade

- Chaque niveau d'analyse linguistique fournit la base au niveau prochain :



Cascade déterministe

- Chaque niveau fournit uniquement l'analyse la plus probable au niveau suivant (Nivre 2007)

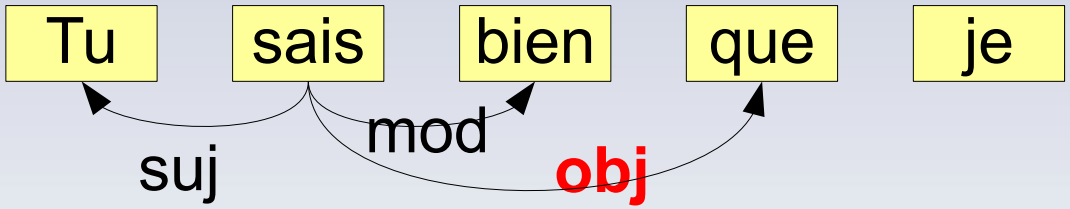
- Sagae et Lavie 2006 applique une recherche en faisceau à l'intérieur du parseur, tout en appliquant une méthode déterministe entre niveaux.

Cascade déterministe - suite

- Mais : souvent une analyse très probable à un niveau plus bas est moins probable à un niveau plus haut.
  - ex. « Tu sais **bien que** je ne travaille pas le dimanche ! »
- Hypothèse : propager les *n* analyses les plus probables = permettre un niveau plus haut de corriger les analyses d'un niveau plus bas.

Cascade avec propagation du faisceau

- Chaque niveau fournit les *n* analyses les plus probables au niveau suivant.
- Tokeniseur :
  - « Tu sais bien_que je ... »
  - « Tu sais bien que je ... »
- Pos Tagger
  - Tu/PRO sais/V bien_que/CS je/PRO …
  - Tu/PRO sais/V bien/ADV que/CS je/PRO …
- Parseur (« savoir » a besoin d'un objet)



Mise en oeuvre : les traits

- Pour assurer que « savoir » à un objet dans une phrase
  - Viser la transition « Reduce » : qui enlève le « haut de pile » (on suppose qu'il n'a plus de dépendances).
  - Donc, « Reduce » est plus probable si « savoir » a déjà un objet
- Traits (syntaxe Talismane) :

IsVerb(X)	PosTag(X)=="V" | PosTag(X)=="VS" | PosTag(X)=="VIMP" | PosTag(X)=="VPP" | PosTag(X)=="VINF" | PosTag(X)=="VPR"
HasObject(X)	DependentCountIf(X,DependencyLabel()=="obj")>0
ObjectArg	IfThenElseNull(IsVerb(Stack[0]), Concat(Lemma(Stack[0]), ToString(HasObject(Stack[0]))))
DirectObjectCompletion	IfThenElseNull(IsVerb(Stack[0]) & PredicateHasFunction(Stack[0],"obj"), HasObject(Stack[0]))
DirectObjectRequiredCompletion	IfThenElseNull(IsVerb(Stack[0]) & PredicateHasFunction(Stack[0],"obj"), Concat(ToString(PredicateFunctionIsOptional(Stack[0],"obj")), ToString(HasObject(Stack[0]))))

Mise en oeuvre : l'évaluation des traits

- Beam width = 1

	sans traits			avec traits		
type	**préc**	**rappel**	**f-score**	**préc**	**rappel**	**f-score**
a_obj	89.57	69.17	78.06	**89.82**	**71.11**	**79.38**
de_obj	87.50	**75.35**	**80.97**	**87.55**	73.26	79.77
obj	98.33	95.85	97.07	**98.37**	**95.93**	**97.14**
p_obj	82.81	60.00	69.58	**83.25**	60.00	**69.74**
suj	**98.36**	86.37	91.97	98.15	**86.61**	**92.02**
total			**87.74**			87.69

- Exemples spécifiques ?

Evaluation de la propagation

- En cours ! (article TALN)

La suite

- Construction des corpus d'évaluation divers et évaluation
- Optimisation, typologie et explication des améliorations
  - Largeur du faisceau
  - Avec ou sans propagation du faisceau
  - Avec différents types de traits
  - Différents valeurs de « cutoff » (seuil d'occurrence pour un trait)
- Nouveaux traits (sémantiques, ...)

Références

- Abeillé A. and Clément L., Building a Treebank for French, in TreeBanks, Springer, 2003.
- Candito M.-H., Crabbé B., and Denis P., Statistical French dependency parsing: treebank conversion and first results, Proceedings of LREC'2010, La Valletta, Malta, 2010.
- Candito M.-H., Nivre J., Denis P. and Henestroza Anguiano E., Benchmarking of Statistical Dependency Parsers for French, in Proceedings of COLING'2010, 2010, Beijing, China
- Crabbé B. and Candito M.-H., Expériences d'analyses syntaxique statistique du français, in Proceedings of TALN 2008, 2008, Avignon, France.
- Denis P. and Sagot B., Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort, in Proceedings of PACLIC, 2009.
- Ho C.-H. and Lin C.-J., Large-scale Linear Support Vector Regression, Technical report, 2012
- Nivre J., Algorithms for Deterministic Incremental Dependency Parsing, in Computational Linguistics, 2008, 34(4), 513-553.
- Ratnaparkhi, A, Maximum entropy models for natural language ambiguity resolution, University of Pennsylvania, 1998.
- Sagae K. and Lavie A., A best-first probabilistic shift-reduce parser, in Proceedings of the COLING/ACL on Main conference poster sessions, pages 691-698, 2006.
- Sagot B., Clément L., de La Clergerie E. and Boullier P., The Lefff 2 syntactic lexicon for French: architecture, acquisition, use, 2006
- Tangy L. Complexification des données et des techniques en linguistique : contributions du TAL aux solutions et aux problèmes. Mémoire d'Habilitation à Diriger des Recherches, Université de Toulouse, 2012

Questions ?

- Merci !