

L'analyse syntaxique statistique du français : état des lieux

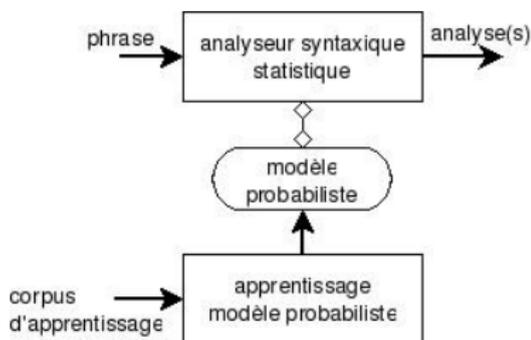
Anthony Sigogne

LIGM, Paris-Est Marne-la-Vallée

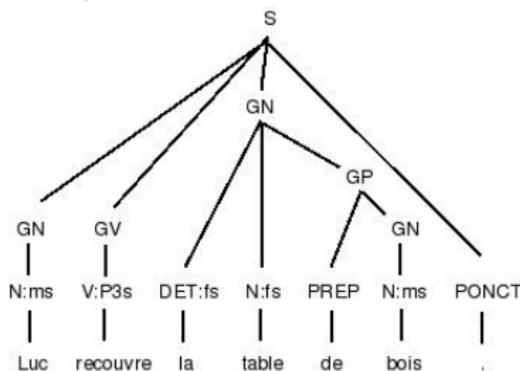
Séminaire de l'équipe Informatique Linguistique
5 juillet 2010

Contexte

Analyse syntaxique statistique



Analyse en constituants



Langue d'étude : le français

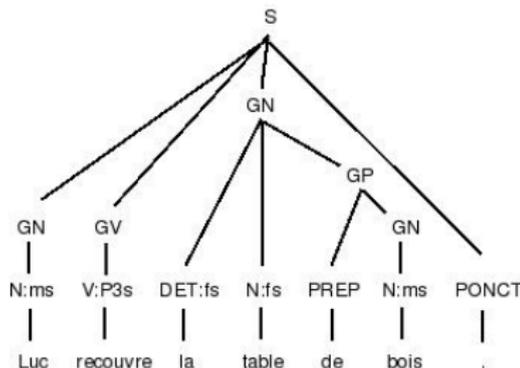
- corpus d'apprentissage en français
- phrases à analyser en français

Contexte

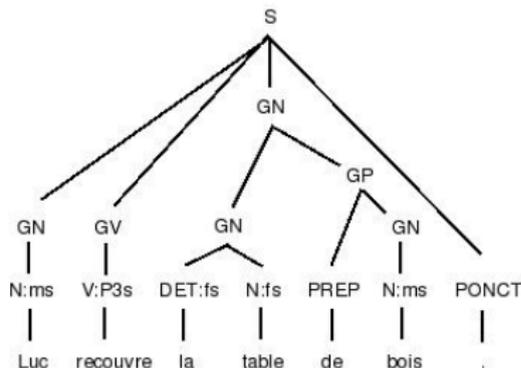
Mesures pour les évaluations : Parseval(Black, 1992)

- rappel : nombre de constituants corrects sur le nombre total de constituants de l'analyse correcte
- précision : nombre de constituants corrects sur le nombre total de constituants de l'analyse produite
- f-score : combinaison du rappel et de la précision, $2 \cdot \frac{\text{precision} \cdot \text{rappel}}{\text{precision} + \text{rappel}}$

analyse correcte



analyse produite



rappel : 6/6, précision : 6/7 et f-score : 0,92

Contexte

État de l'art

- anglais, f-score d'environ 93%, obtenu sur le Penn Treebank (Marcus *et al.* , 1994)
- français, f-score d'environ 86%, obtenu sur le French Treebank (Abeillé *et al.* , 2003)

Comment expliquer une telle différence ?

- reprises des analyseurs anglais appliqués directement sur le français
- optimisations des algorithmes pour l'anglais
- le schéma d'annotation du Penn TreeBank différent de celui du French TreeBank

Objectifs

Dans le cadre de la **langue française** :

- **créer** un analyseur syntaxique probabiliste :
 - déterminer la chaîne de traitements (algorithmes)
 - déterminer le ou les modèles probabilistes à utiliser
- **intégrer** les données du Lexique-Grammaire dans cette chaîne
 - transformation du lexique en un format TAL
 - intégration dans la chaîne de traitements
- **évaluer** l'analyseur obtenu (avec et sans le Lexique-Grammaire)

Plan de la présentation

- 1 Etat de l'art
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - Expériences sur le français

- 2 Analyseur syntaxique du français
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

Types de modèles probabilistes

Modèles génératifs :

- Grammaire hors-contexte probabiliste (PCFG)

Modèles discriminatifs :

- Machines à vecteurs de support (SVM)
- Entropie maximale (ME)
- Champs conditionnels aléatoires (CRF)

Plan de la présentation

- 1 **Etat de l'art**
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - Expériences sur le français

- 2 **Analyseur syntaxique du français**
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

Grammaire hors-contexte probabiliste : PCFG

Modèle statistique le plus basique étendu aux **CFG** (Booth, 1969)

- **extraction** de règles à partir du corpus $A \rightarrow B$
- deux types de règles :
 - **règles lexicales**, membre gauche = étiquette, dérivation = mot du lexique
 - **règles contextuelles**, membre gauche = constituant, dérivation = constituant(s)/étiquette(s)

- chaque règle est associée à une probabilité

$$P(A \rightarrow B) = \text{freq}(A \rightarrow B) / \text{freq}(A)$$

règles contextuelles	prob.	règles lexicales	prob.
S \rightarrow NP VP	0.80	DET \rightarrow le	0.10
S \rightarrow NP VP PP	0.10	DET \rightarrow la	0.20
S \rightarrow VP PP	0.10	N \rightarrow ballon	0.15
NP \rightarrow N	0.50	N \rightarrow chien	0.30

Comment utiliser les PCFG ?

Deux questions :

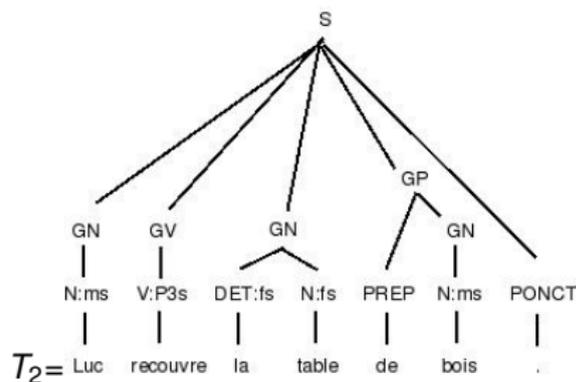
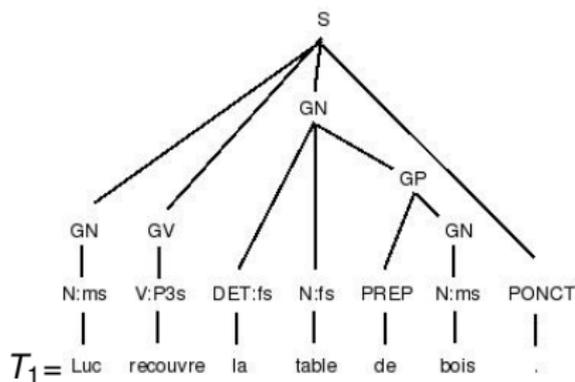
- comment produire les analyses possibles d'une phrase ?
- comment calculer la probabilité d'une analyse ?

Attribuer une probabilité à chaque arbre T possible d'une phrase S :

- probabilité de l'arbre T : produit des probabilités de chaque règle apparaissant dans l'arbre.
- meilleur arbre = le plus probable : $T_{\star}(S) = \operatorname{argmax}_T P(T, S)$

Exemple

S="Luc recouvre la table de bois."



$$P(T_1, S) = P(S \rightarrow \text{GN GV GN PONCT}) * P(\text{GN} \rightarrow \text{N:ms}) * \dots = 0.05$$

$$P(T_2, S) = P(S \rightarrow \text{GN GV GN GP PONCT}) * P(\text{GN} \rightarrow \text{N:ms}) * \dots = 0.01$$

$$T_*(S) = \text{argmax}_T P(T, S) = T_1$$

Comment produire les analyses d'une phrase avec PCFG ?

Algorithmes dérivés de ceux utilisés pour les CFG :

- CKY (Ney, 1991)
- Earley (Earley, 1970)

Spécificités des algorithmes :

- programmation dynamique, utilisation d'une matrice
- méthode "forward-backward"

Déroulement de ces algorithmes :

- phase "forward", calcule les arbres possibles et les probabilités partielles
- phase "backward", permet de retrouver l'analyse la plus probable

Problèmes des PCFG

Les PCFG héritent des CFG, d'où certains problèmes liés à l'estimation des probabilités

Hypothèses d'indépendance trop fortes

- CFG : la dérivation d'un non-terminal est indépendante du contexte de ce noeud dans l'arbre
- PCFG, la probabilité d'une règle est donc indépendante du contexte du noeud

- cas du **NP** (anglais)

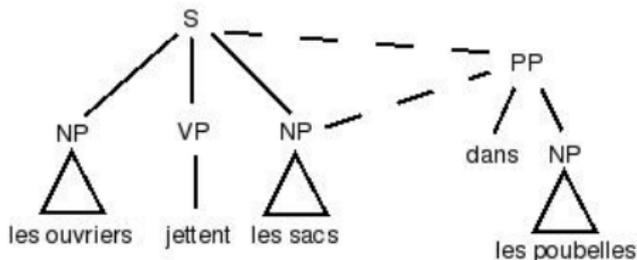
	Pronom	Autres
Sujet	91%	9%
Objet	34%	66%

Solution :

- **parent annotation** (Johnson, 1998; Petrov *et al.* , 2006) :
 - $NP_{sujet} \rightarrow PRO = NP^S \rightarrow PRO$
 - $NP_{objet} \rightarrow PRO = NP^{VP} \rightarrow PRO$

Manque de conditionnement lexical

- les mots ne sont utilisés qu'au niveau des règles lexicales
- pourtant important lors des attachements PP et COORD



Solution :

- **lexicalisation des règles contextuelles** (ajout des têtes)
 - S(jettent) → VP(jettent) NP(sacs) PP(poubelles)
 - NP(sacs) → NP(sacs) PP(poubelles)

Evaluation

analyseur	rappel	précision	f-score	solution
Charniak97	86,7	86,6	86,6	lexicalisation
Collins99	88,1	88,3	88,2	lexicalisation
Charniak00	89,6	89,5	89,5	lexicalisation
Berkeley	89,6	89,8	89,7	parent annotation

Les performances s'améliorent de plus en plus, cependant :

- limitation des performances dûe au modèle PCFG
- il est nécessaire de modifier soit le corpus soit la grammaire

Les résultats sont meilleurs pour Berkeley :

- débat sur l'utilité de lexicaliser les PCFG
- la question se posera également sur le français

Plan de la présentation

- 1 **Etat de l'art**
 - Grammaire hors-contexte probabiliste : PCFG
 - **Analyseurs probabilistes discriminatifs**
 - Apprentissage semi supervisé
 - Expériences sur le français
- 2 **Analyseur syntaxique du français**
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

Analyseurs syntaxiques discriminatifs

Les analyseurs discriminatifs sont basés sur un **modèle discriminatif** :

- Maximum-Entropy (ME)
- Conditional Random Fields (CRF)

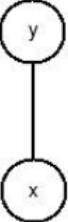
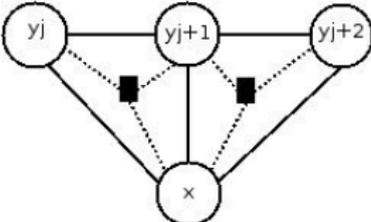
Modèle ME (Jaynes, 1957) et CRF (Lafferty *et al.* , 2001) :

- représenter un processus stochastique par une loi de probabilité

Principe :

- choisir un modèle qui respecte les observations du corpus (features)
- calculer les poids des features (apprentissage du modèle)

Modèles probabilistes discriminatifs

modèle	ME	CRF (linear chain)
prédiction	une classe	séquentiel
représentation		
probabilité	$p(y x)$	$p(\vec{y} \vec{x})$

Analyseurs syntaxiques discriminatifs

On peut classer les analyseurs discriminatifs en plusieurs catégories :

- **chunking-based** (Sang, 2001; Tsuruoka & Tsujii, 2005; Tsuruoka *et al.*, 2009)
 - transforme la tâche d'analyse syntaxique en une série d'étiquetages de séquences de chunks
- **reranking** (Collins, 2000; Charniak & Johnson, 2005)
 - reclasse les n plus probables analyses en sortie d'un analyseur
- **classifier-based** (Ratnaparkhi, 1997; Sagae & Lavie, 2005; Sagae & Lavie, 2006a)
 - utilisation d'une table LR ayant les opérations shift/reduce
- **reparsing** (Henderson & Brill, 1999; Sagae & Lavie, 2006b)
 - intersection des analyses effectuées par plusieurs analyseurs
- ...

Historiquement, les analyseurs chunking-based sont moins performants que les autres :

- série de tâches d'étiquetage indépendantes
- modèles statistiques non séquentiels (ME, memory-based) inadaptés

Pourquoi les utiliser ?

- avec un modèle stat. séquentiel (CRF), bonnes performances
- temps de calcul faible comparé aux analyseurs "whole-sentence"

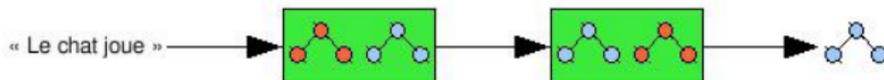
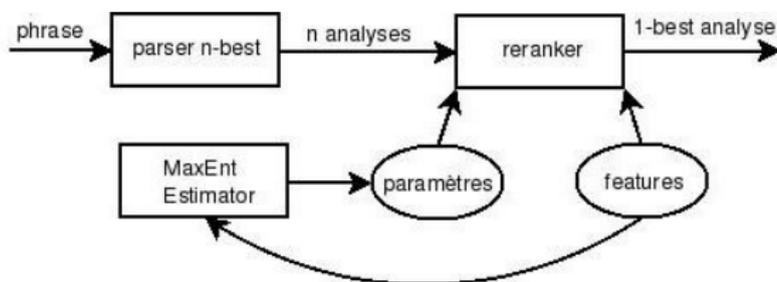
analyseur	f-score	modèle stat.	durée(min.)
Charniak00	89,5	PCFG	23
Sang01	80,49	memory-based	-
Tsuruoka05	85,9	ME	1,5
Tsuruoka09	88,92	CRF	1,7

Reranker (Charniak & Johnson, 2005)

La technique de reranking (Collins, 2000) part d'un simple constat :

- l'arbre syntaxique le plus probable en sortie d'un analyseur n'est **pas forcément le "meilleur"** (en terme de f-score)
 - l'analyse correcte est parfois dans les n plus probables analyses
 - ou bien, une meilleure analyse se trouve parmi les n
- expérience : f-score = 89,5%, oracle f-score = **96,8%**
- reclassement des n plus probables analyses avec un reranker
 - attribution d'un nouveau score à chaque arbre
 - ce score est calculé sur la globalité de l'arbre

Architecture du reranker



analyseur	algorithmme	f-score
Charniak00	-	89,5
Charniak05	Charniak00 + Reranker	91,0

Plan de la présentation

- 1 **Etat de l'art**
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - **Apprentissage semi supervisé**
 - Expériences sur le français
- 2 **Analyseur syntaxique du français**
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

Apprentissage semi supervisé

Techniques d'apprentissage d'un modèle statistique :

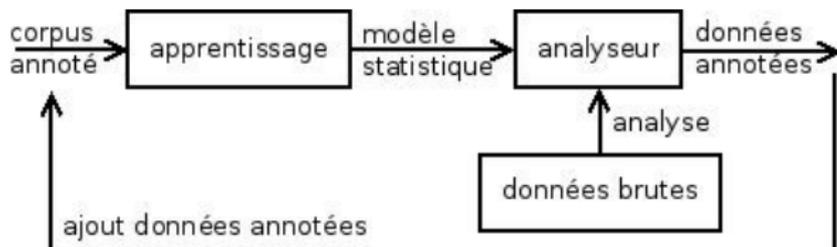
- **supervisé**, apprentissage uniquement sur un corpus annoté
- **semi supervisé**, apprentissage sur un corpus annoté et des données brutes
- **non supervisé**, apprentissage uniquement sur des données brutes

Pourquoi vouloir utiliser des techniques semi/non supervisées ?

- pas de corpus annoté existant pour la langue
- corpus d'apprentissage de petite taille
- analyse de textes ayant des domaines différents de celui du corpus
- des données textuelles de taille infinie

Techniques d'apprentissage semi supervisé

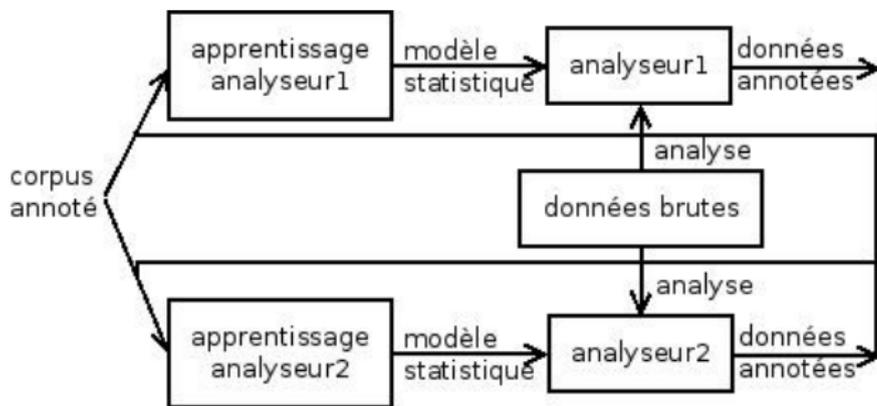
Self-training (Charniak, 1997)



Evaluations :

analyseur	f-score	+self-training
(Charniak, 1997)	86,6	86,6
(Steedman, 2003)	74,4	74,3
(Reichart & Rappoport, 2007)	75	80
(Huang & Harper, 2003)	90,63	91,46

Co-training (Blum & Mitchell, 1998)



Evaluations :

analyseur	f-score	+co-training
(Sarkar, 2001)	70,5	80,1
(Steedman, 2003)	78,6	79,0
(McClosky <i>et al.</i> , 2006)	90,3	92,1

Plan de la présentation

- 1 **Etat de l'art**
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - **Expériences sur le français**

- 2 **Analyseur syntaxique du français**
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

Différences entre le FTB et le PTB

Corpus annotés :

- French Treebank (FTB), corpus de référence du français (Abeillé *et al.* , 2003)
- Penn Treebank (PTB), corpus de référence de l'anglais (Marcus *et al.* , 1994)

Caractéristiques	FTB	PTB
date création	2003	1994
tokens	385000	1000000
tokens/phrase	31	24
formes/mot	16	12
# étiquettes morpho.	13(subcat 80)	44
constituants/arbre	19,6	24

Observations :

- FTB a été créé 10 ans après le PTB → moins d'articles
- flexion riche du français
- platitude du FTB en comparaison du PTB

Premières expériences sur le français

Les premières expériences sont faites avec des portages d'analyseurs basés sur des PCFG lexicalisées (Arun & Keller, 2005; Schlueter & Genabith, 2007)

analyseur	f-score	langue
baseline	65,83	français
Collins99	79,65	français
Collins99	88,20	anglais

Pourquoi des résultats aussi faibles ?

- platitude du FTB → modification du FTB
 - modifications structurelles (VP,COORD)

Nouvelles évaluations :

analyseur	f-score	analyseur
Collins99	80,45	Arun05
Collins99	82,44	Schlueter07

Expériences récentes sur le français

De nouvelles expériences (Crabbe & Candito, 2008; Seddah *et al.* , 2009) remettent en cause les critiques sur le corpus

analyseur	f-score	Arun05	Schluter07	anglais
Collins99	82,52	80,45	82,44	88,20
Charniak00	84,27	-	-	89,5
Berkeley	86,41	-	-	89,6

On peut faire plusieurs remarques :

- les résultats de Arun/Schluter sont moins bons sur le corpus modifié
- les analyseurs lexicalisés subissent une forte baisse
- l'analyseur Berkeley a les meilleurs résultats comme pour l'anglais

Il semblerait que les analyseurs lexicalisés fonctionnent moins bien sur le français (idem pour l'allemand (Dubey & Keller, 2003))

Discussion sur les expériences

- ⇒ peu d'expériences effectuées sur le français
- ⇒ analyseurs probabilistes basés uniquement sur les PCFG (portages)
- ⇒ résultats mitigés et faibles comparés à l'anglais
 - est-ce à cause du corpus ?
 - ⇒ la transformation du FTB s'est révélée inefficace
 - la lexicalisation est-elle possible pour d'autres langues que l'anglais ?
 - ⇒ échecs sur le français et l'allemand
 - les particularités du français bien capturées par les PCFG ?

Plan de la présentation

- 1 Etat de l'art
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - Expériences sur le français
- 2 **Analyseur syntaxique du français**
 - **Contexte de mon travail**
 - Architecture de l'analyseur
 - Perspectives

References43

Contexte de mon travail

D'après l'état de l'art et les remarques sur les précédentes expériences du français :

- analyseurs probabilistes basés uniquement sur les PCFG (**portages**)
 - ⇒ déterminer les performances des analyseurs discriminatifs sur le français
- pas d'expériences sur des techniques d'apprentissage semi supervisées
- dictionnaire lexical toujours extrait du corpus
 - ⇒ utiliser des ressources externes : DELA, Prolex,...

Plan de la présentation

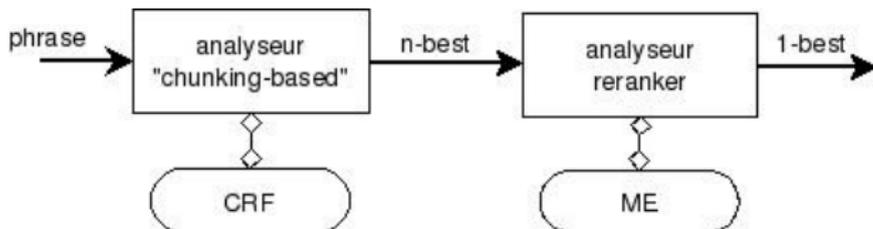
- 1 Etat de l'art
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - Expériences sur le français
- 2 **Analyseur syntaxique du français**
 - Contexte de mon travail
 - **Architecture de l'analyseur**
 - Perspectives

References43

Architecture de l'analyseur

Chaîne de traitements syntaxiques à expérimenter :

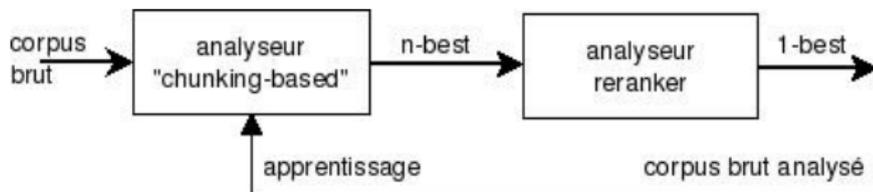
- un analyseur "chunking-based" (Tsuruoka *et al.* , 2009) calculant les n plus probables analyses
- un reranker (Charniak & Johnson, 2005) reclassant les n analyses



Apprentissage semi supervisé

Pour augmenter les performances de l'analyseur chunking-based, on peut utiliser une technique de co-training (McClosky *et al.* , 2006) :

- 1ère étape, apprentissage de l'analyseur chunking-based et le reranker sur le corpus d'apprentissage
- 2ème étape, apprentissage de l'analyseur chunking-based sur les analyses produites par la chaîne de traitements sur un corpus brut



Dictionnaires lexicaux

La phase d'étiquetage morpho-syntaxique de l'analyseur chunking-based peut assigner des étiquettes à partir de :

- corpus d'apprentissage
 - Avantage :
 - jeu d'étiquette identique au corpus
 - Désavantage :
 - beaucoup de mots inconnus (noms propres notamment)
- ressources externes : DELA, Prolex, ...
 - Avantage :
 - grande couverture des mots, moins de mots inconnus
 - Désavantage :
 - jeu d'étiquettes différent de celui du corpus
- combinaison des deux possibilités ?

Plan de la présentation

- 1 Etat de l'art
 - Grammaire hors-contexte probabiliste : PCFG
 - Analyseurs probabilistes discriminatifs
 - Apprentissage semi supervisé
 - Expériences sur le français
- 2 Analyseur syntaxique du français
 - Contexte de mon travail
 - Architecture de l'analyseur
 - Perspectives

References43

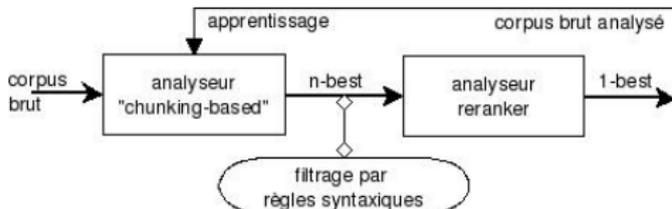
A court terme

Création de la chaîne de traitements syntaxiques :

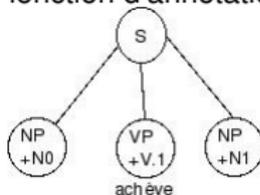
- implémentation de l'analyseur chunking-based (presque terminé)
- intégration du reranker
 - Disponible en ligne
- premières évaluations

A long terme

- intégration du Lexique-Grammaire (lexique syntaxique) dans la chaîne de traitements
 - fonction de filtrage des n-best



- fonction d'annotation des arbres



- évaluations avec ou sans technique d'apprentissage semi supervisé

FIN

Merci pour votre attention.

References I

Abeillé, A., Clément, L., & Toussanel, F. 2003.

Building a treebank for French.

In: Abeillé, Anne (ed), Treebanks.

Arun, A., & Keller, F. 2005.

Lexicalization in crosslinguistic probabilistic parsing: The case of french.

Pages 306–313 of: ACL2003.

Black, E. 1992.

Meeting of interest group on evaluation of broad-coverage grammars of English.

In: Linguist List 3587.

Blum, A., & Mitchell, T. 1998.

Combining labeled and unlabeled data with co-training.

Pages 92–100 of: Computational Learning Theory.

References II

Booth, T.L. 1969.

Probabilistic representation of formal languages.

Pages 74–81 of: Switching and Automata Theory.

Charniak, E. 1997.

Statistical parsing with a context-free grammar and word statistics.

Pages 598–603 of: AAAI.

Charniak, E., & Johnson, M. 2005.

Coarse-to-fine n-best parsing and MaxEnt discriminative reranking.

Pages 173–180 of: ACL2005.

Collins, M. 2000.

Discriminative reranking for natural language parsing.

Pages 175–182 of: Machine Learning: Proceedings of the Seventeenth International Conference ICML 2000.

References III

Crabbe, B., & Candito, M. 2008.

Expériences d'analyse syntaxique statistique du français.

In: TALN2008.

Dubey, A., & Keller, F. 2003.

Probabilistic parsing for german using sister-head dependencies.

Pages 96–103 of: ACL2003.

Earley, J. 1970.

An efficient context-free parsing algorithm.

Pages 94–102 of: Communications of the Association for Computing Machinery.

Henderson, J., & Brill, E. 1999.

Exploiting diversity in natural language processing: Combining parsers.

In: Empirical Methods in Natural Language Processing.

References IV

Huang, Z., & Harper, M. 2003.

Self-training PCFG grammars with latent annotations across languages.

In: EACL.

Jaynes, E. 1957.

Information theory and statistical mechanics.

Pages 620–630 of: Physical Review 106.

Johnson, M. 1998.

PCFG models of linguistic tree representations.

Pages 613–632 of: Computational Linguistics.

Lafferty, J.D., McCallum, A., & Pereira, F.C.N. 2001.

Conditional Random Fields : Probabilistic models for segmenting and labeling sequence data.

Pages 282–289 of: ICML 2001.

References V

- Marcus, M.P., Santorini, H., & Marcinkiewicz, M.A. 1994.
Building a large annotated corpus of English : The Penn Treebank.
Pages 313–330 of: Computational Linguistics.
- McClosky, D., Charniak, E., & Johnson, M. 2006.
Effective self-training for parsing.
Pages 152–159 of: ACL2006.
- Ney, H. 1991.
Dynamic programming parsing for context-free grammars in continuous
speech recognition.
Pages 336–340 of: IEEE Transactions on Signal Processing.
- Petrov, S., Barret, S., Thibaux, L., & Klein, D. 2006.
Learning accurate, compact, and interpretable tree annotation.
*Pages 433–440 of: International Conference on Computational
Linguistics.*

References VI

Ratnaparkhi, A. 1997.

A linear observed time statistical parser based on maximum entropy models.

In: EMNLP1997.

Reichart, R., & Rappoport, A. 2007.

Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets.

In: ACL.

Sagae, K., & Lavie, A. 2005.

A classifier-based parser with linear run-time complexity.

Pages 125–132 of: IWPT.

Sagae, K., & Lavie, A. 2006a.

A best-first probabilistic shift-reduce parser.

In: COLING-ACL.

References VII

Sagae, K., & Lavie, A. 2006b.

Parser Combination by reparsing.

Pages 129–132 of: ACL.

Sang, E. 2001.

Transforming a chunker to a parser.

Pages 177–188 of: Computational Linguistics in the Netherlands.

Sarkar, A. 2001.

Applying co-training methods to statistical parsing.

Pages 95–102 of: NAACL.

Schluter, N., & Genabith, J. Van. 2007.

Preparing, restructuring and augmenting a french treebank: Lexicalised parsers or coherent treebanks?

Pages 200–209 of: PACLIC2007.

References VIII

Seddah, D., Candito, M., & Crabbe, B. 2009.

Adaptation de parsers statistiques lexicalisés pour le français: une évaluation complète sur corpus arborés.

In: TALN2009.

Steedman, M. 2003.

Bootstrapping statistical parsers from small datasets.

In: EACL.

Tsuruoka, Y., & Tsujii, J. 2005.

Chunk parsing revisited.

Pages 133–140 of: IWPT2005.

Tsuruoka, Y., Tsujii, J., & Ananiadou, S. 2009.

Fast full parsing by linear-chain conditional random fields.

Pages 790–798 of: ACL 2009.