Le couteau suisse du TAL

Alexis Nasr
en collaboration avec : Frédéric Béchet, José Deulofeu, Benoit Favre, Carlos Ramisch, André Valli

Séminaire TALEP - 20/10/2015

Comment utiliser un analyseur syntaxique pour :

- détecter des disfluences
  - oui donc il y a il y a trois euh trois RER sur quatre
- reconnaître des mots composés
  - Il parle de la bière
  - Il boit de la bière
- segmenter des transcriptions de l'oral
  - je sais pas // j'ai vu une émission la dernière fois

Analyse par arbre couvrant

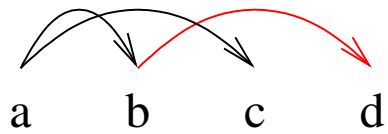
- Pas de grammaire de réecriture
- Etant donné une phrase $S = m_1 \dots m_n$ et un jeu d'étiquettes fonctionnelles $\mathcal{F}$ tout arbre couvrant T sur S constitue une analyse possible de S.
- Une contrainte syntaxique : la projectivité
- Calcul du score d'un arbre :

$$s(T) = \sum_{\psi \in \psi(T)} s(\psi)$$

- $\psi(T)$ est l'ensemble de toutes les parties pertinentes (facteurs) de T
- $s(\psi)$ est le score de ψ

Projectivité

- Propriété des arbres ordonnés
- Un dépendant ne peut être séparé de son gouverneur dans la chaîne que par des descendants de ce dernier



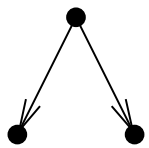
- Hypothèse linguistique raisonnable
- Réduction importante de l'espace de recherche

nb nœuds	3	4	5	6	7	8	9	10
arbres	9	64	625	7776	117649	2.10^6	43.10^6	1.10^9
proj	7	30	143	728	3876	21318	1200001	690690
rapport	1	2	4	11	30	98	358	1448

Décomposition

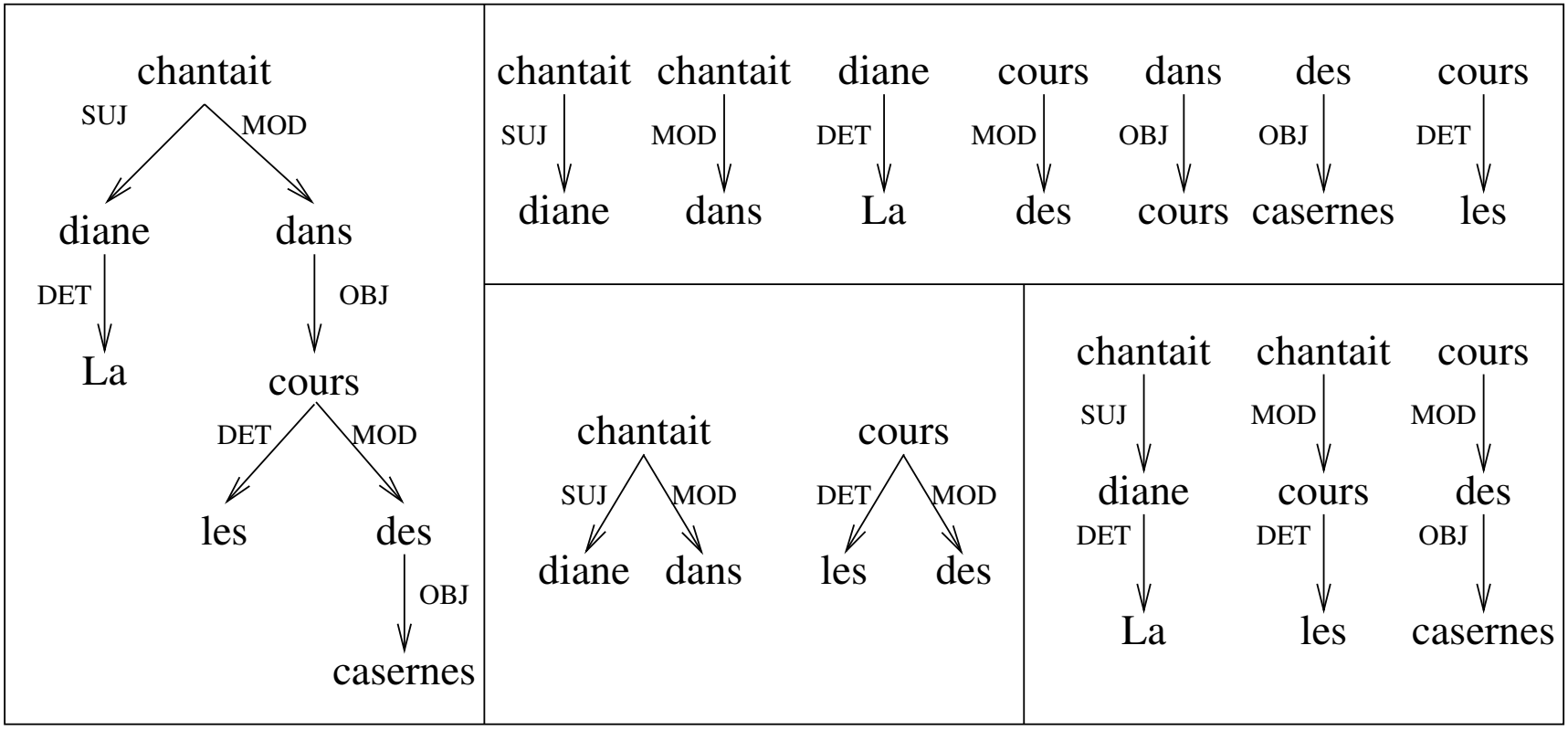
- modèles d'ordre 1 : un facteur est réduit à une dépendance :

- modèles d'ordre 2 : un facteur est de la forme :



or

Exemple


chantait
SUJ
MOD
diane
dans
DET
OBJ
La
cours
DET
MOD
les
des
OBJ
casernes
chantait
SUJ
diane
chantait
MOD
dans
diane
DET
La
cours
MOD
des
dans
OBJ
cours
des
OBJ
casernes
cours
DET
les
chantait
SUJ
MOD
diane
dans
cours
DET
MOD
les
des
chantait
SUJ
diane
DET
La
chantait
MOD
cours
DET
les
cours
MOD
des
OBJ
casernes

Score d'un facteur

- Un facteur est décomposé en un vecteur f de traits (features) élémentaires.

(diane, diane, N)	-DET->	(la, le, ART)
(X, X, N)	-DET->	(X ,X , ART)
(diane, X, N)	-DET->	(X ,X , ART)
(X, X, N)	-DET->	(la ,X , ART)
(X, diane, N)	-DET->	(X ,le , ART)
...		

- Chaque trait est associé à un poids w
- Le score du facteur est le produit scalaire $f \cdot w$
- Les poids sont calculés à l'aide d'un algorithme d'apprentissage en ligne, ici un perceptron
- Modèle souple, on peut facilement ajouter des traits

Recherche de l'arbre de meilleur score

$$\hat{T} = \arg\max_{T \in \mathcal{T}(S)} \sum_{X \in \psi(T)} s(X)$$

$\mathcal{T}(S)$ est l'ensemble des arbres projectifs possibles pour la phrase S

- Enumération de tous les arbres projectifs possibles pour une phrase
- Calcul du score de chaque arbre
- Sélection de l'arbre de meilleur score
- Programmation dynamique : $O(n^3)$

Détection de disfluences

- oui donc il y a il y a trois euh trois RER sur quatre
- deux approches :
  1. Détection et élimination de disfluences avant traitements linguistiques
  2. Intégration des disfluences dans la structure syntaxique et apprentissage d'un analyseur disfluent
     L'analyseur réalise en même temps l'analyse syntaxique et la détection de disfluences

Le corpus DECODA

- Collecté dans le cadre du projet ANR DECODA
- Conversation entre des usagers et des téléconseillés de la RATP
- 74 heures
- 2100 dialogues
- phénomènes spécifiques à l'oral (disfluences)
  - répétitions
  - mots tronqués
  - reformulations
  - pauses remplies

Le corpus DECODA

Annotations

- marquage des disfluences
- étiquetage en parties de discours
- analyse syntaxique

Approche incrémentale

- analyse automatique
- détection d'erreurs systématiques
- écriture de règles de correction
- réentrainement de outils ...

Le corpus DECODA

	turn nb.	token nb		
		TOTAL	REP	DM
TRAIN	93,561	521,377	15,484	35,183
TEST	3,639	25,231	882	1692

Le corpus de test a été entièrement vérifié à la main

Exemple

1	oui	ADV	DM	0	DISFLINK
2	donc	COO	NULL	0	ROOT
3	il	CLI	REP	2	DISFLINK
4	y	CLI	REP	3	DISFLINK
5	a	VRB	REP	4	DISFLINK
6	il	CLI	NULL	8	SUJ
7	y	CLI	NULL	8	AFF
8	a	VRB	NULL	2	DEP_COORD
9	trois	ADJ	REP	8	DISFLINK
10	euh	INT	DM	9	DISFLINK
11	trois	DET	NULL	12	DET
12	RER	NOM	NULL	8	OBJ
13	sur	PRE	NULL	12	MOD
14	quatre	ADJ	NULL	13	OBJ

Détecteur de disfluences

Modèle CRF
Différents ensembles de traits
Evaluation sur DECODA

Traits	PREC	REC	F-MEAS.
word n-gram	98.1	76.2	85.8
word + rep. feat	96.7	81.1	88.2
word + POS n-gram	97.5	83.5	89.9
word + POS + rep. feat.	96.0	85.1	90.2

Conclusions :

- Bonne précision
- Rappel plus faible (certaines répetitions sont difficiles à prédire sans la syntaxe)

Analyseurs

- Trois analyseurs
  - appris sur le FTB
  - appris sur DECODA sans disfluences
  - appris sur DECODA avec disfluences
- Quatre corpus
  - FTB
  - NODISF : DECODA sans disfluences
  - AUTO : DECODA avec élimination auto. des disfluences
  - DISF : DECODA avec disfluences

Analyseur FTB

		DECODA			FTB
		AUTO	NODISF	DISF	
1^{st} order	UAS	71.66	71.01	65.78	87.92
	LAS	65.19	64.28	58.28	85.54
2^{nd} order	UAS	72.29	71.87	66.09	89.71
	LAS	65.88	65.30	58.70	87.32

- Métriques
  - UAS Unlabeled Attachment Scores
  - LAS Labeled Attachment Scores
- Conclusions
  - très mauvaises performances sur DISF
  - meilleurs résultats sur AUTO mais ça reste médiocre (58.70 $\rightarrow$ 65.88)

DECODA sans disfluences

		DECODA			FTB
		AUTO	NODISF	DISF	
1^{st} order	UAS	85.52	86.50	80.08	77.71
	LAS	83.45	84.70	77.87	73.67
2^{nd} order	UAS	85.03	86.06	79.61	76.77
	LAS	82.96	84.26	77.40	72.76

Conclusions :

- Bien meilleur que FTB (65.88 $\rightarrow$ 83.45)
- Meilleurs résultats avec les modèles de premier ordre

DECODA avec disfluences

		DECODA			FTB
		AUTO	NODISF	DISF	
1^{st} order	UAS	85.90	86.47	85.83	77.93
	LAS	83.86	84.60	83.62	73.80
2^{nd} order	UAS	85.63	86.07	85.62	77.25
	LAS	83.61	84.19	83.56	73.20

Conclusions :

- Un peu mieux que l'approche en deux étapes (83.45 $\rightarrow$ 83.86)
- L'intégration des disfluences dans la syntaxe n'ajoute pas de bruit
- Architecture plus simple

Reconnaissance de mots composés

- Article partitif
  1. Il boit **de la** bière
  2. Il parle **de la** bière
- Conjonctions complexes
  3. Je mange **bien que** je n'aie pas faim
  4. Je pense **bien que** je n'ai pas faim

Statistiques - articles partitifs

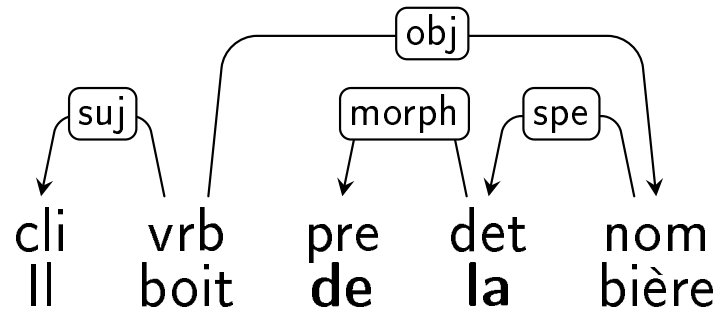
de+det	**#sent**	**det.**	**other**	**#occur**
le (du)	136	33.1	66.9	16,609,049
la	138	21.0	79.0	10,849,384
les (des)	129	77.5	22.5	23,395,857
l'	136	16.9	83.1	8,204,687
Total	539	36.5	63.5	59,058,977

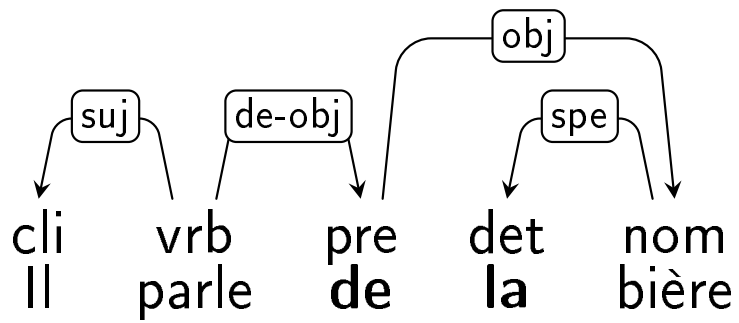
- données collectées sur web as a corpus
- constructions fréquentes
- constructions ambiguës

Statistiques - conjonctions complexes

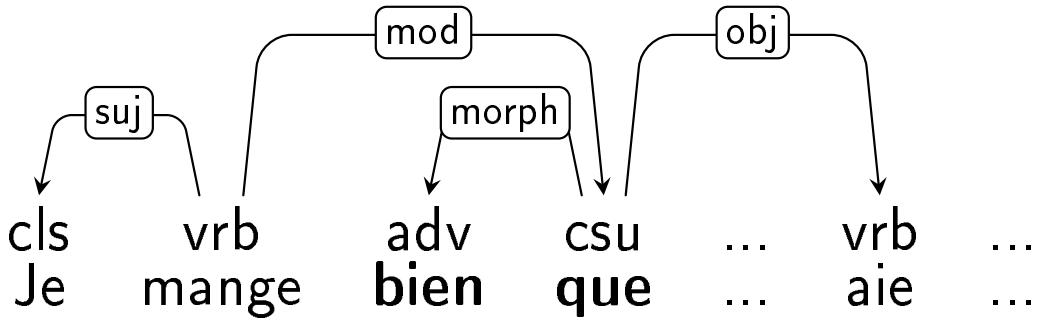
adv+*que*	#sent	conj.	other	#occur
ainsi	103	76.7	23.3	498,377
alors	110	88.2	11.8	291,235
autant	107	86.0	14.0	39,401
bien	99	37.4	62.6	156,798
encore	93	21.5	78.5	18,394
maintenant	120	55.8	44.2	16,567
tant	98	20.4	79.6	168,485
Total	730	56.4	43.6	1,189,257

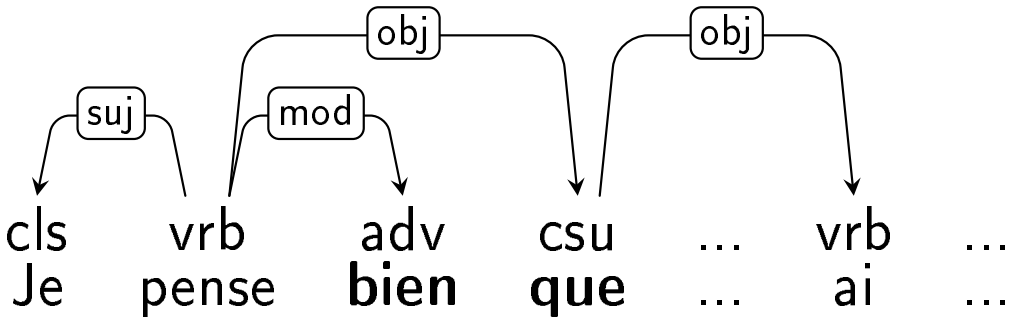
Représentation syntaxique - articles partitifs





Représentation syntaxique - conjonctions complexes





Prédiction du lien morph

- Tâche dépendante du lexique
  1. Il boit de_la bière
  2. Il parle de la bière
  3. Je pense bien que je n'ai pas faim
  4. Je mange bien_que je n'aie pas faim
- Pour généraliser, l'analyseur a besoin de connaître la valence des verbes
- *manger* n'admet pas un objet indirect introduit par *de*
- *parler* admet un objet indirect introduit par *de*
- *penser* admet une completive objet
- *parler* n'admet pas de completive objet

Dicovalence

- format

```
VAL$ penser: P0 P1
VTYPE$ predicator simple
EG$ je pense qu'il doit accepter ces offres ...
P0$ qui, je, nous, elle, il, ils, on, celui-ci, ceux-ci
P1$ que, ça, le(qpind), ça(qpind), le(inf), ça(inf)
```

- traits +/- que et +/- de

manger	-que	-de
penser	+que	-de
boire	-que	-de
parler	-que	+de

- Nombre d'entrées portant des traits de sous-cat

-que	+que	-de	+de
3,814	356	3,450	720

Traits de sous-cat

- Traits modélisant la co-occurrence des traits de sous-cat et de configurations syntaxiques

1	g.sf	g.pos	d.fct	d.pos	
2	g.sf	g.pos	d.fct	d.pos	gd.pos
3	g.sf	g.pos	d.fct	d.lem	gd.pos

- Deux exemples de traits

1	-que	vrb	obj	csu	
3	-que	vrb	mod	que	adv

Apprentissage

- Ajout des nouvelles features dans l'analyseur
- Ajout du lien morph dans le French Treebank
- Ajout des traits de sous-cat

```
46 on           on           CLS 48 SUJ    _
47 lui          lui          CLI 48 COMP   _
48 parle        parler       VRB 45 OBJ    +de_n
49 de           de           PRE 48 DEOBJ  _
50 l'           le           DET 51 SPEC   _
51 éventualité éventualité NOM 49 OBJ    _

68 qui          qui          PRQ 70 SUJ    _
69 a            avoir        VRB 70 AUX    -de_n
70 eu           avoir        VPP 63 COMP   -de_n
71 bien         bien         ADV 70 COMP   _
72 de           de           PRE 73 MORPH  _
73 le           le           DET 74 SPEC   _
74 mal          mal          NOM 70 OBJ    _
```

Prédiction du lien MORPH conjonctions complexes

	Baseline prec.		Stanford	Without SF			With SF		
adv+*que*	global	indiv.		Prec.	Recall	F1	Prec.	Recall	F1
ainsi que	76.7	76.7	81.44	96.00	91.14	93.50	95.94	89.87	92.81
alors que	88.2	88.2	95.10	92.78	92.78	92.78	93.81	93.81	93.81
autant que	86.0	86.0	92.00	86.95	65.21	74.53	86.66	70.65	77.84
bien que	37.4	62.6	55.22	86.84	89.18	88.00	91.66	89.18	90.41
encore que	21.5	78.5	64.52	72.72	80.00	76.19	92.85	65.00	76.47
maintenant que	55.8	55.8	87.01	85.24	77.61	81.25	90.91	74.62	81.96
tant que	20.4	79.6	90.91	78.94	75.00	76.92	82.35	70.00	75.67
Total	56.4	75.3	83.06	88.71	82.03	85.24	91.57	81.79	86.41

Prediction du lien MORPH déterminants partitifs

	Baseline prec.		Stanford	Without SF			With SF		
de+det	global	indiv.		Prec.	Recall	F1	Prec.	Recall	F1
de le	66.9	79.0	56.96	72.50	64.44	68.23	85.41	91.11	88.17
de la	79.0	77.5	22.83	58.13	86.20	69.44	81.25	89.65	85.24
de les	22.5	66.9	87.72	97.36	74.00	84.09	98.70	76.00	85.87
de l'	83.1	83.1	18.55	57.14	69.56	62.74	64.51	86.95	74.07
Total	63.5	76.6	44.37	77.00	73.09	75.00	**86.70**	**82.74**	**84.67**

Segmentation de transcriptions de l'oral

- Contrairement à l'écrit, pas de marques formelles de segmentation en "phrases"
- On peut utiliser des indices acoustiques, prosodiques, lexicaux ou syntaxiques
- segmentation manuelle d'un corpus de 233 945 mots
- apprentissage d'un modèle CRF fondé sur
  - les mots
  - les parties de discours
  - les pauses
- Performances : r=0.57 p=0.78 f=0.66
- on aimerait utiliser l'analyseur pour prédire la segmentation
- mais on ne peut faire tourner l'analyseur sur un flux de texte

Exemple

```
1  je          je          CLS  2   SUJ  260.43 260.46 L1
2  sais        savoir      VRB  0   ROOT 260.46 260.64 L1
3  pas         pas         ADN  2   COMP 260.64 260.76 L1
4  j'          je          CLS  6   SUJ  260.76 260.87 L1
5  ai          avoir       VRB  6   AUX  260.87 260.93 L1
6  vu          voir        VPP  0   ROOT 260.93 261.05 L1
7  une         un          DET  8   SPEC 261.05 261.14 L1
8  émission    émission    NOM  6   COMP 261.14 261.54 L1
9  la          la          DET  11  SPEC 261.54 261.6  L1
10 dernière    dernier     ADJ  11  COMP 261.6  261.86 L1
11 fois        foi         NOM  6   COMP 261.86 261.98 L1

1  il          il          CLS  2   SUJ  317.64 317.7  L1
2  est         être        VRB  0   ROOT 317.7  317.73 L1
3  un#peu      un peu      ADV  4   COMP 317.73 317.92 L1
4  fou         fou         ADJ  2   COMP 318.0  318.1  L1
5  lui         lui         PRO  0   ROOT 318.15 318.24 L1
6  hum         hum         INT  5   DISF 327.3  327.43 L1
```

Segmentation

- précision plus élevée que le rappel
- le système sous segmente :
  - il a tendance à rater des segmentations (43%)
  - quand il en prédit une, c'est souvent correct (78%)
- idée : finir la segmentation lors de l'analyse syntaxique
- plus ambitieux : un analyseur syntaxique pour flux textuel