Introduction aux étiqueteurs grammaticaux (taggeurs)

D'après le chapitre 10 de Manning and Schütze [1999].

- Qu'est-ce qu'un taggeur ?
- Comment faire un taggeur ?
  - approche markovienne
  - approche transformationnelle
- Cas particulier des segmenteurs (chunkers)

But d'un taggeur

But: associer chaque mot d'une phrase à une **étiquette grammaticale** (ou tag) comme: ADJ, NOMC, NOMP, DET, etc. On parle également d'étiquettes **Part Of Speech** (POS).

Exemple:

mot	tag	mot	tag
la	Dete-dart-ddef-femi-sing	à	Prep
séance	NomC-femi-sing	15	Quan-femi-plur-qdef
est	Verb-IndPré-sing-p3	h	NomC-femi-plur
ouverte	Verb-ParPas-femi-sing	43	Quan-masc-plur-qdef

Pourquoi c'est intéressant ?

- analyse partielle (shallow parsing)
- des taux de bon étiquetage raisonnables (supérieurs à 95%),
- utile dans certaines applications comme:
  - l'extraction d'information *(information extraction)*
  - la réponse automatique à des questions *(question answering)*

Idée: les POS suffisent souvent à identifier des groupes syntaxiques simples comme les groupes nominaux.

Exemple (fictif) d'extraction d'information

Tâche: remplir des formulaires découverte/auteurs à partir de textes.

> Kuhn Jeff , a physicist at the Institute for Astronomy at the University of Hawaii, and his colleagues may have found evidence of some kind of emission process in the plane of the planets.

champ	information
découvreur	Kuhn Jeff and his colleagues
status	physicist at the Institute for Astronomy at the University of Hawaii
découverte	some kind of emission process in the plane of the planets

Le jeu d'étiquettes (le *tag set*)

Dépend de l'application et de la précision requise. En général le jeu d'étiquettes est un ensemble de 50 à 400 étiquettes. Au RALI, un étiqueteur du français été entraîné sur un jeu de 330 étiquettes. En voici quelques unes:

tag	signification	exemple
NomC-masc-sing	Nom commun masculin singulier	haricot
NomC-femi-sing	Nom commum féminin singulier	poire
Verb-IndImp-sing-p3	verbe à l'indicatif du présent, 3ème personne du singulier	voulait
AdjQ-masc-plur	adjectif qualificatif masculin pluriel	nombreux
ConC	conjonction de coordination	et
ConS	conjonction de subordination	que

Exemple de tagset pris dans Charniak [1993] p.3

POS	signification	exemples
noun	nom commun	dog, equation, concerts
prop	nom propre	Alice, Romulus
pro	pronom	I,you,it,they,them
pos	possessif	my, your
verb	verbe	is, touch, went, remitted
adj	adjectif	red, large, remiss
det	article	the, a, some
prep	préposition	in, to, into
conj	conjonction	and, but, since
aux	auxiliaire	be, have
modal	vb. modaux	will, can, must, should
adv	adverbe	closely, quickly
wh	wh-mouvements	who, what, where
punc	ponctuation	. ? !

Est-ce difficile de tagger?

La belle ferme le voile

ART NOMC VERB ART NOMC	s'il s'agit d'une jolie femme qui ferme un voile.
ART ADJQ NOMC PRO VERB	si une ferme voile la vue de la chose dont on fait mention par *le*.

belle peut être à la fois un *adjectif féminin singulier* ou un *nom commun féminin singulier*.

ferme peut être à la fois un *adjectif singulier* (féminin ou masculin), un *nom commun féminin singulier*, ou encore un *verbe* (indicatif présent (1,3-ps), impératif présent (2ps), subjonctif présent (1,3-ps)).

voile peut-être à la fois un *nom commun singulier* (féminin ou masculin), un *verbe* (indicatif présent (1,3-ps), impératif présent (2ps), subjonctif présent (1,3-ps))

Est-ce difficile de tagger ?

Il existe cependant de nombreux mots qui ne sont étiquetables que par un seul tag:

mot	tag
âge	NomC-masc-sing
âne	NomC-masc-sing
ânerie	NomC-fem-sing
éducatif	AdjQ-masc-sing
électoraux	AdjQ-masc-plur
zyeutera	Verb-IndFutur-sing-p3

Cette proportion de mots peut dépasser 50% des types d'un grand corpus (dépend beaucoup du tagset et de la langue).

Quelle information utiliser pour tagger ?

Certaines séquences sont plus fréquentes que d'autres

- Il est par exemple plus fréquent d'avoir la séquence:

  ART ADJ NOMC (le blanc manteau de neige) que
  ART ADJ VERB

- Un taggeur qui se baserait sur cette information devrait normalement associer l'étiquette **NOMC** à ébauche plutôt que l'étiquette **VERB** (3ème personne du singulier de l'indicatif présent ou subjonctif) dans la phrase: la belle ébauche.

- **Pb:** l'information du contexte n'est pas forcément fiable

En pratique, cette information seule ne suffit pas (taux de 77%)

Quelle information utiliser pour tagger ?

La nature même du mot

belle est probablement plus fréquemment employé en français comme un adjectif que comme un nom commun.

En fait, Charniak [1993] reporte qu'un tagger simple qui étiquette un mot par son étiquette la plus fréquente permet d'obtenir des taux d'étiquetage de l'ordre de 90%.

$\hookrightarrow$ C'est en pratique souvent la performance de référence à laquelle on compare la performance d'un taggeur donné.

Note: pour obtenir l'étiquette la plus fréquente d'un mot, il faut compter les étiquettes associées à ce mot dans un corpus déjà étiqueté. Certains dictionnaires peuvent éventuellement fournir cette information.

Est-ce qu'un taux de 95% est un bon taux ?

5 erreurs tous les 100 mots. 1 phrase $\sim$ 20 mots $\Longrightarrow$ une erreur par phrase (en pratique, plusieurs erreurs peuvent intervenir dans la même phrase).

Bien sûr, tout dépend de l'application...

Note: Il est toujours difficile de comparer des taggeurs entraînés sur des corpus différents: quel est le pourcentage de mots qui possèdent plus d'une étiquette, quelle est la taille du vocabulaire, quel est le jeu d'étiquettes, le taux de mots inconnus à l'apprentissage et dans les tests, etc...

Voir l'action de recherche GRACE du réseau Francil de l'AUPELF-UREF Adda et al. [1999]

HMM et taggeurs

Soit w_1^n une séquence de n mots; alors on cherche la séquence de tags de plus forte probabilité:

$$\begin{aligned} \hat{t_1^n} &= \text{argmax}_{t_1^n}\, p(t_1^n|w_1^n) \\ &= \text{argmax}_{t_1^n}\, p(w_1^n|t_1^n) \times p(t_1^n) \end{aligned}$$

Avec hypothèse d'indépendance + hypothèse markovienne (ordre 1 ici):

$$p(w_1^n|t_1^n) \times p(t_1^n) = \prod_{i=1}^{n} p(w_i|t_i) \times p(t_i|t_{i-1})$$

D'où:

$$\hat{t_1^n} = \underset{t_1^n}{\text{argmax}} \prod_{i=1}^{n} \underbrace{p(w_i|t_i)}_{\text{émission}} \times \underbrace{p(t_i|t_{i-1})}_{\text{transition}}$$

HMM et taggeurs, çà n'est pas:

$$\hat{t_1^n} = \prod_{i=1}^{n} p(t_i|w_i) \times p(t_i|t_{i-1})$$

Rappelez-vous de l'idée du **canal bruité**: on présente des tags à un canal qui les transmet après les avoir modifiés en mots. Le but est de retrouver la séquence de tags. On cherche pour cela la séquence de tags dont la probabilité (selon un modèle du canal) est maximale sachant la séquence de mots observée en sortie du canal (décodage $= \mathrm{argmax}_t$).

Les résultats peuvent cependant ne pas être si mauvais que ca (Charniak [1993])

Entraînement d'un taggeur HMM: à partir d'un corpus d'entraînement étiqueté

Un état = une étiquette

On passe sur le corpus étiqueté et on applique les estimées MLE (fréquence relative):

$$p(w|t) = \frac{|(w,t)|}{\sum_w |(w,t)|} = \frac{|(w,t)|}{|t|}$$
$$p(t|t') = \frac{|t't|}{\sum_t |t't|} = \frac{|t't|}{|t'|}$$

où (w,t) désigne le fait que w est étiqueté par le tag t; et $t't$ représente la séquence de deux tags t' et t.

Note: $p(t|t')$ est simplement un modèle bigramme.

Dans Merialdo [1994], les auteurs utilisent un corpus annoté de 40 000 phrases, les taggeurs du RALI ont été entraînés à partir de corpus d'environ 100 000 mots ($\sim$ 5000 phrases par langue)

Entraînement d'un taggeur HMM

- Les probabilités d'émission $p(w|t)$ peuvent être rangées dans une matrice E de dimension $N \times M$ òu N est le nombre d'états et M le nombre de mots différents (types). $E[i,j]$ indique alors la probabilité que le $i$ème état génère le $j$ème type.

- Les probabilités de transition peuvent être rangées dans une matrice T de dimension $N \times N$. $T[i,j]$ indique alors la probabilité de transiter du $i$ème état vers le $j$ème.

En pratique, ces deux matrices sont creuses (contiennent de nombreux 0) et la dimension M est assez grande (50000 ou plus).

Problème avec l'estimateur MLE

- il se peut très bien qu'une transition ne soit pas observée dans $\mathcal{T}$, bien que légitime. Sa probabilité est cependant nulle.

- de manière encore plus probable, un mot n'a peut-être pas été étiqueté avec toute ses formes possibles. Par exemple, on a peut-être toujours rencontré garde comme un **NomC-masc-sing**, alors qu'il peut apparaître comme un **NomC-fem-sing** ou encore comme un **verbe** (à différents temps et personnes). Mais $p(garde|\text{NomC-fem-sing}) = 0$.

- pire encore, il existe des mots qui ne sont pas dans $\mathcal{T}$ mais qui apparaîtront lorsqu'on utilisera le modèle. Le décodeur ne marchera pas.

À propos du décodeur

On peut appliquer $viterbi$ pour obtenir la séquence de tags la plus probable selon notre HMM (cad on maximise $p(t_1^n|w_1^n)$), ou alors on peut utiliser un critère local (on maximise $p(t_i|w_1^n)$, pour tout i).

Merialdo [1994] montre que cela ne fait pas de grande différence.

Dans le décodage par viterbi, si une erreur se produit, alors elle a de fortes chances de se répercuter sur le ou les tag(s) suivant(s). En revanche dans l'approche locale, ceci ne se produit pas (mais il y a potentiellement plus de foyers d'erreurs).

Le plus courant est tout de même le décodage global (viterbi).

Gestion des mots inconnus

- **Une idée:** un mot inconnu peut potentiellement être associé à tous les tags $ouverts$. C'est-à-dire tous les tags sauf ceux tels que les prépositions ou les articles (dont on connaît tous les représentants).

  En pratique, cela implique d'avoir un jeu de paramètres $p(\text{UNK}|t)$ pour tous les tags autorisés $\Longrightarrow$ lissage

- **Une autre idée:** on peut s'aider des propriétés du mot à étiqueter pour lui attribuer son étiquette. Les terminaisons de mots comme iques, tions, ments peuvent fournir (en français) des pistes. Le fait qu'un mot soit en majuscule est également un indicateur (d'un nom propre par exemple).

  En pratique on cherche à modéliser des choses comme: $p(\text{UNK}, \text{end=iques}, \text{capital}|t)$ et on fait souvent l'hypothèse d'indépendance de tous ces traits.

Pourquoi s'arrêter à un taggeur bigramme ?

Si le corpus d'entraînement est assez grand, on peut calculer les paramètres d'un taggeur trigramme:

$$\hat{t_1^n} = \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} p(w_i|t_i) \times p(t_i|t_{i-2}t_{i-1})$$

En principe cela peut nous permettre de désambiguïser plus de choses.

Exemple: l'étiquette à associer à fatigue dans la fatigue dépend de ce qui précède la.

il	la fatigue	$\rightarrow$	Verb
de	la fatigue	$\rightarrow$	NomC

Pourquoi s'arrêter à un taggeur bigramme ?

Mais çà n'est pas nécessairement payant dans tous les cas. Notamment, il n'y a habituellement pas de dépendance forte entre deux tags séparés par une virgule: $p(t|NomC, \text{VIRGULE}) \approx p(t|\text{VIRGULE})$

On peut remédier à cela en combinant linéairement plusieurs types de modèles (bi- tri- grammes), ou encore en faisant des modèles à mémoire variable. Par exemple par analyse/correction: si on repère par exemple une erreur systématique de l'étiqueteur sur une séquence particulière alors on augmente la mémoire du modèle pour ce cas.

D'autres approches ont été proposées pour éviter de faire de l'analyse/correction manuelle Schütze and Singer [1994], Ristad and Thomas [1997a,b].

Pourquoi s'arrêter à un taggeur bigramme ?

Note: pour augmenter la mémoire d'un modèle, il suffit simplement d'ajouter des états:

mot	tag-1	tag-2	mix
BOS	BOS	BOS	BOS
il	PRON	BOS PRON	BOS PRON
a	AUX	PRON AUX	PRON AUX
dit	VB	AUX VB	AUX VB
,	VIRG	VB VIRG	VIRG
que	CONJ	VIRG CONJ	VIRG CONS
		...	

↪ On change seulement l'étiquetage du corpus, les techniques d'entraînement du modèle sont quant à elles inchangées.

Peut-on s'affranchir d'un corpus déjà étiqueté ?

$\hookrightarrow$ La réponse semble être plutôt négative Merialdo [1994], Elworthy [1994].

Ces travaux (et d'autres) semblent montrer qu'entraîner les paramètres du modèle markovien à l'aide de Baum-Welch (voir le jeu d'acétates `[hmm]`) n'est pas une très bonne solution (ou alors dans des contextes très particuliers).

Exemple de fait remarqué par Merialdo [1994]: si on possède un corpus d'entraînement (même de taille modeste) dont on se sert pour initialiser les paramètres, alors dès qu'on lance une itération de forward-backward, on dégrade les performances du modèle. Sauf si les corpus de test et d'entraînement sont très différents.

"Problème" avec les HMMS appliqués au tagging

- Les probabilités d'émission $(p(w|t))$ ont généralement une plage de valeurs beaucoup plus importante que celle des probabilités de transition $(p(t|t'))$. $\hookrightarrow$ Les probabilités d'émission sont plus faibles que les autres en général.

- Ceci est lié au fait que le nombre d'observations (mots) est très supérieur au nombre d'états (tags).

$\hookrightarrow$ Ceci a pour effet de donner plus de poids aux probabilités d'émission: ce sont elles qui font la différence pour une topologie donnée.

Cas particulier du tagging: le *Chunking*

Définition: Le chunking consiste à découper une phrase en groupes relevant d'une organisation syntaxique.

> [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September].

Père du chunking: Abney [1991] qui recherchait des corrélations entre les tags pour identifier des groupes.

L'histoire continue: avec une base (corpus d'entraînement et de test) mise à disposition pour la conférence **CONLL'2000** (COmputational Natural Language Learning).

http://cnts.uia.ac.be/conll2000/chunking/

Le corpus CONLL:le jeu de C-tags

des B-étiquettes marquant le début d'un groupe. Ex: $B\text{-}NP$ marque le premier mot d'un groupe nominal (noun phrase); $B\text{-}VP$ marque le début d'un groupe verbal, etc.

des I-étiquettes: marquant un mot dans un groupe qui n'est pas le premier mot du groupe. Ex: $I\text{-}NP$ indique qu'un mot est à l'intérieur d'un groupe nominal (d'au moins deux mots), mais n'en est pas le premier mot.

autres: étiquette O pour marquer des mots comme des parenthèses, ou autres signes de ponctuation qui n'appartiennent pas à un groupe.

Au total 22 étiquettes caractérisant les groupes adjectivaux, adverbiaux, verbaux, nominaux, etc. Les deux étiquettes les plus fréquentes sont $I\text{-}NP$ et $B\text{-}NP$, marquant respectivement le milieu d'un groupe nominal, et son début.

Le corpus CONLL: Exemple de phrase annotée

mot	tag	C-tag
He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP

mot	tag	C-tag
to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP
.	.	O

Le corpus CONLL

corpus	mots	types	happax
test	49389	8119	55%
train	220663	19123	49%

Environ 3000 mots du corpus de test n'ont pas été vus dans le corpus d'entraînement.

Les étiquettes ne sont pas représentées de manière égale:

I-NP	63307	B-ADVP	4227	I-PP	291	I-INTJ	9
B-NP	55081	B-SBAR	2207	I-CONJP	73	I-UCP	6
O	27902	B-ADJP	2060	I-SBAR	70	I-PRT	2
B-VP	21467	I-ADJP	643	B-CONJP	56	B-UCP	2
B-PP	21281	B-PRT	556	B-INTJ	31		
I-VP	12003	I-ADVP	443	B-LST	10		

Chunker = Tagger

- On prend la sortie d'un tagger "normal" qui constitue l'entrée d'un C-taggeur.

the deficit could narrow . . .
$\longrightarrow$ DT NN MD VB . . .
$\longrightarrow$ BOS-DT DT-NN NN-MD MD-VB . . .
$\longrightarrow$ B-NP I-NP B-VP I-VP . . .

- On peut également considérer d'autres combinaisons Osborne [2000]:

$$w_i \quad \overset{\Rightarrow}{HMM}_{-1} \quad POS_i$$

$$(w_i, POS_i) \quad \overset{\Rightarrow}{HMM}_{-2} \quad IOB_i$$

$$(POS_i, IOB_i, POS_{i+1}, IOB_{i+1}) \quad \overset{\Rightarrow}{HMM}_{-3} \quad \hat{IOB}_i$$

Chunker = Tagger

Si le corpus d'entraînement est suffisamment grand, on peut également entraîner directement un taggeur avec le jeu d'étiquettes des C-tags $\longrightarrow$ $p(w|\text{c-tag})$ et $p(\text{c-tag}|\text{c-tag'})$.

Exemple (viterbi sur un C-HMM d'ordre 1):

[mr. B-NP] [speaker I-NP] [, O] [our B-NP] [government I-NP] [has B-VP] [demonstrated I-VP] [its B-NP] [support I-NP] [for B-PP] [these B-NP] [important I-NP] [principles I-NP]

$\Longrightarrow$ [NP mr. speaker], [NP our government] [VP has demonstrated] [NP its support] [PP for] [these important principles]

Note: les HMMs ne donnent pas nécessairement les meilleurs résultats.

Chunker & HMM, quelques chiffres

ordre	Transition			Observation		
	A-matrix	nbp	%	B-matrix	nbp	%
HMM-1	$[24 \times 24]$	162	28%	$[24 \times 17259]$	24606	5.9%
HMM-2	$[157 \times 157]$	856	3.4%	$[157 \times 17259]$	38123	1.4%
HMM-3	$[829 \times 829]$	3213	0.4%	$[829 \times 17259]$	59804	0.4%

ordre est l'ordre du modèle HMM considéré (ordre 1 signifie: $p(t|t')$, ordre 2 signifie $p(t|t''t')$, etc.);

nbp est le nombre de paramètres stockés dans la matrice;

% indique le pourcentage "d'occupation" de la matrice (plus il est proche de 0, plus la matrice est creuse).

Performance/temps sans lissage des probabilités

Apprentissage par calcul des fréquences relatives (pas de lissage).

	training (8936 sent.)			test (2012 sent.)		
	-logp	time	err	-logp	err	nb
1	242.2	6.4u	6%	184.8	10.3%	573
2	227.3	70.2u	3.6%	154.3	8.9%	329
3	212.8	1287u	1.75%	117.6	9.6%	121

-logp $-log$ de la vraisemblance de l'observation (meilleur si faible)

time u-temps retourné par la commande unix $time$ (Pentium-III sous Linux, charge normale)

err pourcentage de mots mal étiquetés

nb nombre de décodages avec une réponse

Analyse des erreurs (HMM-4 sur training)

extrait d'une matrice de confusion

	B-ADJ	B-ADV	B-INT	B-NP	B-PRP	B-PRT	B-SBA	B-VP	I-ADJ	I-ADV	I-INT	I-NP	
B-ADJP	0	2	0	27	7	0	1	6	3	0	0	29	[121/2135 : 5.67]
B-ADVP	19	0	1	75	46	3	1	5	0	2	0	38	[212/4279 : 4.95]
B-INTJ	0	0	0	1	0	0	0	0	0	0	0	0	[1/31 : 3.23]
B-NP	4	14	0	0	8	0	6	15	0	8	0	867	[985/55451 : 1.78]
B-PP	1	17	0	69	0	6	42	14	0	1	0	154	[341/21260 : 1.60]
B-PRT	5	41	0	2	17	0	0	0	0	1	0	0	[67/613 : 10.93]
B-SBAR	0	4	0	36	26	0	0	0	0	1	0	4	[91/2247 : 4.05]
B-VP	0	7	0	18	85	0	0	0	0	0	0	141	[425/21696 : 1.96]
I-ADJP	0	1	0	2	2	0	0	0	0	2	0	2	[12/648 : 1.85]
I-ADVP	2	2	0	1	3	0	0	0	3	0	0	8	[21/447 : 4.70]
I-INTJ	0	0	0	0	0	0	0	1	0	0	0	0	[1/10 : 10.00]
I-NP	12	15	0	260	55	0	0	90	1	1	0	0	[657/62396 : 1.05]

erreurs: I-I (1.16%), B-I (57.61%), B-B (17.77%), O-I (16.96%), O-B (6.49%)

Taggeurs transformationnels (transformation-based taggers)[1]

Idée: transformer une séquence de tags (incorrecte) à l'aide d'une batterie ordonnée de règles transformationnelles qui permettent d'améliorer la séquence.

Deux composants:

patrons: liste des transformations admissibles

un algorithme d'apprentissage de l'ordonnancement de ces transformations

C'est l'idée sur laquelle est basé le plus cité des taggeurs: Brill [1992, 1995]

[1] D'après Manning and Schütze [1999], p. 363

Les patrons du taggeur de Brill

patron = contexte d'application + réécriture ($t_i \longrightarrow t'_i$)

schéma	t_{i-3}	t_{i-2}	t_{i-1}	t_i	t_{i+1}	t_{i+2}	t_{i+3}
1			—	⋆			
2				⋆	—		
3		—	—	⋆			
4				⋆	—	—	
5	—	—	—	⋆			
6				⋆	—	—	—
7			—	⋆	—		
8			—	⋆		—	
9		—		⋆	—		

⋆ est le site potentiel de réécriture, — indique où un trigger peut apparaître.

La ligne 7 se lit: si un *trigger* (à déterminer) apparaît juste avant t_i, et qu'un autre (à déterminer) apparaît juste après, alors une réécriture (à déterminer) de t_i peut avoir lieu.

Les patrons - Manning and Schütze [1999], p. 363

réécriture			contexte
NN	$\longrightarrow$	VB	le tag précédant est la prep. TO
VBP	$\longrightarrow$	VB	un modal (MD) est dans les 3 tags qui précèdent
JJR	$\longrightarrow$	RBR	le tag suivant est JJ
VBP	$\longrightarrow$	VB	un des deux mots précédants est *n't*

- la règle 1 dit: ré-étiquette un nom en verbe (à l'infinitif) s'il est précédé de la préposition **to** (contre-exemple: *go to school*).
- la règle 2 s'applique aux verbes ayant la même forme au passé et au présent (ex: *cut*, *put*) et dit qu'en présence d'un modal (max 3 mots avant), on devrait préférer la forme au présent (exemple: *you may cut*).
- la règle 3 transforme un adjectif comparatif (JJR) en un adverbe comparatif (RBR) s'il est suivi directement d'un adjectif (ex: *the more valuable*).
- la règle 4 est proche de la règle 2 pour le cas des négations (*shouldn't* est coupé en deux mots).

Les patrons du taggeur de Brill

Donc il y a des contextes (triggers) mettant en œuvre des étiquettes ou des mots.

Les word-triggers peuvent être conditionnés par la nature du mot considéré et d'autres contraintes sur les tags (ex: "le mot courant est w et le tag qui suit est t).

Il existe de plus des triggers morphologiques qui permettent de gérer par exemple les mots inconnus.

Exemple:

"remplace NN par NNS si le mot courant se termine par s"

$\hookrightarrow$ Bref, beaucoup de latitude dans les règles que l'on peut apprendre. C'est aussi un problème !

Apprentissage des taggeurs transformationnels

Input: Un corpus taggé (tag le plus fréquent pour un mot donné): C_0
Output: L'ensemble des meilleures règles de transformation ainsi que leur ordonnancement.

for k := 0 step 1 **do**
$\quad$ v := $\text{argmin}_{v_i} E(v_i(C_k))$
$\quad$ **si** $(E(C_k) - E(\text{v}(C_k)) < \epsilon)$ **alors** aller à *fin*
$\quad$ $C_{k+1} := \text{v}(C_k)$
$\quad$ $\tau_{k+1} := \text{v}$
end
fin: séquence ordonnée: $\tau_1, \ldots \tau_k$

$E(C_k)$ est le nombre de mots mal taggés dans le corpus C à l'itération k.
v(C) est le corpus obtenu en appliquant la règle v sur le corpus C; v_i une règle particulière.
ϵ spécifie notre tolérance à l'erreur.

Note: C'est un algorithme vorace (*greedy algorithm*).

Application des règles de ré-écriture

- application de la gauche vers la droite.
- choix de l'application immédiate ou retardée (Brill = retardée).
  Soit la règle "$A \longrightarrow B$ si A précède".
  Alors en mode retardé, la séquence $AAAA$ devient $ABBB$ (on marque les transformations à effectuer, puis on les fait), alors qu'en mode immédiat, elle devient $ABAB$ (on applique immédiatement la règle).

Note:

Brill reporte un taggeur appris de manière non supervisée (sans corpus taggé) avec un taux de 95.6%. Pour cela, il utilise l'information des mots non ambigus (qui possèdent un seul tag, selon le dictionnaire).

Exemple: can dans $The\ can\ is\ open$ sera taggé NN (et non MD), si dans le contexte "AT — BEZ", les mots non ambigus sont majoritairement étiquetés NN.

Discussion sur les taggeurs transformationnels

L'apprentissage par transformation ne semble pas sur-entraîner (comme c'est le cas par exemple pour les HMMs).

Le prix à payer est que si l'espace des transformations est grand, alors l'apprentissage ne peut se faire en temps raisonnable.

De nombreux contextes d'application des règles de réécriture sont potentiellement possibles. Ici l'expertise humaine à permi de réduire les patrons à un sous-ensemble intéressant et restreint.

Les temps de décodage avec les taggeurs transformationnels peuvent être beaucoup plus faibles qu'avec les modèles HMM, si correctement codés (voir Manning and Schütze [1999] page 368 pour plus d'information).

Quelques lectures sur le tagging[2]

- Brant [2000] décrit de manière particulièrement claire les "petits trucs" à appliquer (et habituellement non documentés) et qui expliquent les différences de performance que l'on impute habituellement aux méthodes d'apprentissage. C'est probablement le tagger avec le meilleur rapport performance/vitesse ($\sim$ 50000 tokens/sec.).
  TnT (le tagger décrit) est disponible gratuitement à:
  `http://www.coli.uni-sb.de/~thorsten/tnt`
- Lire Giménez and Màrquez [2003] pour une comparaison plus récente de TnT *vs* SVMs.
- Voir aussi l'approche maxent Ratnaparkhi [1996] et Toutanova and Manning [2000]
- Dans Toutanova et al. [2003] les auteurs montrent qu'un modèle unidirectionnel (gauche-droite ou droite-gauche) n'est pas nécessairement la meilleure solution et qu'un modèle bidirectionnel peut faire mieux en tagging.
- Une approche *memory-based* au tagging est également décrite dans Daelemans et al. [1996].

[2] Liste absolument non exhaustive.

Références

Steven Abney. Parsing by chunks. Robert Berwick and Steven Abney and Carol Tenny, "Principle-Based Parsing", Kluwer Academic, 1991.

Gilles Adda, Joseph Mariani, Patrick Paroubek, Martin Rajman, and Josette Lecomte. The grace evaluation for pos tagging for french language. In *Cahiers/Langues*, volume Vol. 2, Issue 2. http://www.john-libbey-eurotext.fr/en/revues/lan/index.htm, June 1999.

T. Brant. Tnt - a statistical part-of-speech tagger. In *ANLP,*, Seattle, WA, 2000.

Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

Eugene Charniak. *Statistical Language Learning*. MIT Press, 1993.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. Mbt: A memory-based part of speech

tagger generator. In *Proc. of Fourth Workshop on Very Large Corpora, ACL SIGDA*, pages 14–27, 1996.

David Elworthy. Does baum-welch reestimation help taggers? In *In Proceedings of the 4th ACL Conference on Applied Natural Language Processing (ANLP'94)*, Stuttgart, Germany, 1994.

J. Giménez and L. Màrquez. Fast and accurate part-of-speech tagging: The svm approach revisited. In *RANLP*, Borovets, Bulgaria, 2003.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

Bernard Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–172, 1994.

Miles Osborne. Shallow parsing as part-of-speech tagging. In *Proceedings of the 4th Computational Natural Language Learning Conference (CoNLL)*, Lisbon, Portugal, September 2000. ACL SigNLL.

A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *EMNLP*, Philadelphia, PA, 1996.

E. Ristad and R. Thomas. Hierarchical non-emitting markov models. Technical Report CS-TR-544-97, Department of Computer Science, Princeton University, 1997a.

E. S. Ristad and R. G. Thomas. Nonuniform markov models. In *Proc. ICASSP '97*, pages 791–794, Munich, Germany, 1997b.

Hinrich Schütze and Yoram Singer. Part-of-speech tagging using a variable memory markov model. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Las Cruces, New Mexico, June 1994.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*, 2003.

K. Toutanova and C. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference EMNLP/VLC*, pages 63–71, 2000.