Introduction à l'étiquetage de séquences (tagging, chunking, ner)

felipe@iro.umontreal.ca

RALI
Dept. Informatique et Recherche Opérationnelle
Université de **Montréal**

V1.0

Last compiled: 9 octobre 2018

Plan

Plan

rali

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).

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 est-ce intéressant ?

- une problème canonique d'étiquetage de séquence
- des taux de bon étiquetage *raisonnables* (supérieurs à 95%),
- utile à d'autres tâches ou applications
  - input à l'analyse syntaxique
  - extraction d'information (information extraction)
  - réponse automatique à des questions (question answering)

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

rali

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 un ensemble de 40 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 imparfait, 3ème personne du singulier	voulait
AdjQ-masc-plur	adjectif qualificatif masculin pluriel	nombreux
ConC	conjonction de coordination	et
ConS	conjonction de subordination	que

Tagset populaire (PTB)

https://www.sketchengine.eu/penn-treebank-tagset/

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
FW	foreign word	les
IN	preposition, subordinating conjunction	in, of, like
IN/that	that as subordinator	that
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest

Tagset populaire (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	. ? !

rali

Est-ce difficile de tagger ?

La belle ferme le voile

- ART NOMC VERB ART NOMC ▷ une jolie femme qui ferme un voile.
- ART ADJQ NOMC PRO VERB ▷ une ferme voile la vue de la chose dont on fait mention par *le*.

belle ▷ *adjectif féminin singulier*
▷ *nom commun féminin singulier*.

ferme ▷ *adjectif singulier (féminin ou masculin)*
▷ *nom commun féminin singulier*
▷ *verbe (indicatif présent (1,3-ps), impératif présent (2ps), subjonctif présent (1,3-ps))*.

voile ▷ *nom commun singulier (féminin ou masculin)*
▷ *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 :

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

- Peut dépasser 50% des types d'un grand corpus

Quelle information utiliser pour tagger ?

- Il est plus fréquent en français d'avoir la séquence :

  ART ADJ NOMC (le blanc manteau de neige) que
  ART ADJ VERB *(est-ce même possible ?)*

- 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.

- L'information du contexte n'est pas forcément fiable

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

rali

Quelle information utiliser pour tagger ?

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

- (Charniak, 1993) mentionne qu'un tagger simple qui étiquette un mot par son étiquette la plus fréquente (et qui étiquette nom-propre un mot inconnu) permet d'obtenir des taux d'étiquetage de l'ordre de 90%

- requiert l'étiquette la plus fréquente d'un mot (listée dans certains dictionnaires)

rali

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
  plusieurs erreurs peuvent intervenir dans la même phrase
- bien sûr, tout dépend de l'application...
- Il est toujours difficile de comparer des taggeurs entraînés sur des corpus différents :
  - pourcentage de mots qui possèdent plus d'une étiquette dans le train ?
  - taille du vocabulaire ?
  - tagset ?
  - taux de mots inconnus ?

Voir l'action de recherche GRACE (Adda et al., 1999)

rali

À propos des taux d'erreurs

System name	Short description	Main publication	Software	Extra Data?***	All tokens	Unknown
TnT*	Hidden markov model	Brants (2000)	TnT	No	96.46%	85.86%
MElt	MEMM with external lexical information	Denis and Sagot (2009)	Alpage linguistic workbench	No	96.96%	91.29%
GENiA Tagger**	Maximum entropy cyclic dependency network	Tsuruoka, et al (2005)	GENiA	No	97.05%	Not availa
Averaged Perceptron	Averaged Perception discriminative sequence model	Collins (2002)	Not available	No	97.11%	Not availa
Maxent easiest-first	Maximum entropy bidirectional easiest-first inference	Tsuruoka and Tsujii (2005)	Easiest-first	No	97.15%	Not availa
SVMTool	SVM-based tagger and tagger generator	Giménez and Márquez (2004)	SVMTool	No	97.16%	89.01%
Morče/COMPOST	Averaged Perceptron	Spoustová et al. (2009)	[1]	No	97.23%	Not availa
Stanford Tagger 1.0	Maximum entropy cyclic dependency network	Toutanova et al. (2003)	Stanford Tagger	No	97.24%	89.04%
Stanford Tagger 2.0	Maximum entropy cyclic dependency network	Manning (2011)	Stanford Tagger	No	97.29%	89.70%
Stanford Tagger 2.0	Maximum entropy cyclic dependency network	Manning (2011)	Stanford Tagger	Yes	97.32%	90.79%
LTAG-spinal	Bidirectional perceptron learning	Shen et al. (2007)	LTAG-spinal	No	97.33%	Not availa
Morče/COMPOST	Averaged Perceptron	Spoustová et al. (2009)	[2]	Yes	97.44%	Not availa
SCCN	Semi-supervised condensed nearest neighbor	Søgaard (2011)	SCCN	Yes	97.50%	Not availa

mesuré sur le WSJ (PTB)

rali

À propos du tagging

- parfois difficile de donner un tag à un mot :
  - mot compressés : *cannot, gonna, wanna*, etc.
  - expressions multi-mots : *pomme de terre, vice et versa*, etc.
  - réelle ambiguité : *The Duchess was entertaining last night*

    adj ou vb ?

    (ex. pris de *Part-of-speech Tagging Guidelines for the Penn Treebank Project*)
  - utilisation / mention : *Le mot mot a 3 lettres*

rali

Un article intelligent sur le sujet

Part-of-Speech Tagging from 97% to 100%. Is it Time for Some Linguistics (**Manning:2011**).

- 97% de performance (sur le PTB) est supérieur à la performance humaine, mais :
  - des erreurs dans les annotations (15.5% des erreurs)
    - *Our market got/VBD hit/VB a/DT lot/NN harder/RBR on Monday than the listed market*

    il s'agit d'un verbe au participe passé
  - manque de constance dans les annotations (28% des erreurs)
    - *the 30's*

    30's étiqueté parfois comme **CD** (cardinal) ou **NNS** (noun, plural) ou encore **NN** (single noun) dans le PTB
  - plus de contexte (19.5% des erreurs)
    - *They/PRP set/VBP up/RP absurd/JJ situations/NNS, detached from reallity*

    verbe au présent ou au passé ?

Un article intelligent sur le sujet

Part-of-Speech Tagging from 97% to 100%. Is it Time for Some Linguistics Manning:2011.

- 200 règles pour corriger dans le PTB :
  - des erreurs VBD (passé) / VBD (verbe au participe passé)
  - des erreur sur l'étiquetage de *that* qui est un mot hautement ambigu
  - etc.

Table 6. Accuracy of taggers on the final test set *WSJ* 22–24.

Model	Corrected Data	Sentence Accuracy	Token Accuracy	Unknown Accuracy
NAACL 2003	no	55.75%	97.21%	88.50%
Replication	no	56.44%	97.26%	89.31%
5WSHAPES	no	56.65%	97.29%	89.70%
5WSHAPESDS	no	56.92%	97.32%	90.79%
5WSHAPESDS	yes	61.81%	97.67%	90.49%

Plan

HMM et taggeurs

- Soit w_1^n une séquence de n mots ; on cherche :

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

- 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 :

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

Entraînement d'un taggeur HMM

- Un état = une étiquette
- 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

- (Merialdo, 1994) utilise 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)

Problème avec l'estimateur MLE

- une transition légitime peut ne pas avoir été observée dans train. Sa probabilité devient cependant nulle.
- un mot n'a peut-être pas été étiqueté dans trainavec toutes ses formes possibles.
  - ex : on a peut-être toujours rencontré garde comme un **NomC-masc-sing** alors qu'il peut apparaître comme :
    - **NomC-fem-sing**
    - **verbe** (à différents temps et personnes).
  - Mais $p(garde|\text{NomC-fem-sing}) = 0$.
- mots inconnus (Viterbi ne marchera pas forcément)

À propos du décodeur

- Viterbi : $\hat{t}_1^n = \text{argmax}_{t_1^n}\, p(t_1^n|w_1^n)$
- Critère local : $\hat{t} = \text{argmax}_t\, p(t|w_1^n)$

- (Merialdo, 1994) montre que cela ne fait pas de grande différence :
  - avec viterbi, les erreurs arrivent (potentiellement) en grappes
  - avec l'approche locale, il y a (potentiellement) plus de foyers d'erreur

- Le plus courant est tout de même le décodage global (viterbi). Lire cependant (**Johnson:2007**).

rali

Gestion des mots inconnus

- **idée 1 :** un mot inconnu peut potentiellement être associé à tous les tags *ouverts* :
  - tag ouvert : tous les tags sauf ceux tels que les prépositions ou les articles (dont on connaît tous les représentants).
  - $p(\text{UNK}|t)$ pour tous les tags autorisés $\Longrightarrow$ lissage

- **idée 2 :** s'aider des propriétés formelles du mot à étiqueter :
  - Les suffixes comme iques, tions, ments peuvent fournir (en français) des indices
  - Le fait qu'un mot soit en majuscule est également un indicateur (nom propre, acronyme).
  - Par exemple en estimant : $p(\text{UNK}, \text{end=iques}, \text{capital}|t)$
    (on fait souvent l'hypothèse d'indépendance de ces traits)

rali

Pourquoi s'arrêter à un taggeur bigramme ?

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

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

- permet de désambiguïser plus de choses :

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

il la fatigue → Verb
de la fatigue → NomC

Augmenter l'ordre du modèle

- Pas toujours payant :
  - Ex : pas de dépendance forte entre deux tags séparés par une virgule :
    $p(t|NomC, \text{VIRGULE}) \approx p(t|\text{VIRGULE})$
- Combiner linéairement plusieurs types de modèles (bi- tri- grammes)
- Modèles à mémoire variable :
  - par analyse/correction manuelle : si on repère une erreur systématique de l'étiqueteur sur une séquence particulière alors on augmente la **mémoire** du modèle pour ce cas.
  - par analyse/correction automatique : (Ristad et Thomas, 1997a ; Ristad et Thomas, 1997b ; Schütze et Singer, 1994).

rali

Pourquoi s'arrêter à un taggeur bigramme ?

Note : pour augmenter la mémoire d'un modèle, il suffit 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.

rali

TNT (TNT)

- (**Zavrel:1999**) ont comparé 7 modèles et ont montré la supériorité de TNT (un HMM d'ordre 2, avec lissage, plus traitement des mots inconnus) tout en étant rapide (test et train)
- parmi les meilleurs systèmes selon (**Horsmann:2015**)
- quelques trucs :
  - ajouter un token de fin de phrase améliore les résultats : $p(EOS|t_n, t_{n-1})$ où n est le nombre de mots dans la phrase
  - lissage important $p(t_3|t_2, t_1) = \lambda_1 p_{ML}(t_3) + \lambda_2 p_{ML}(t_3|t_2) + \lambda_3 p_{ML}(t_3|t_2, t_1)$ où les λ sont non contextuels (mais appris)
  - gestion des mots inconnus à l'aide d'une table de suffixes (gestion assez "subtile")
  - paramètres doublés selon que le mot commence ou pas par une majuscule (c_i variable indicatrice) : $p(t_3, c_3|t_2, c_2, t_1, c_1)$
  - beam search plutôt que Viterbi pour aller plus vite (sans perte de perf.)

TreeTagger (**TreeTagger**)

- un modèle de Markov d'ordre k, sélection du contexte conditionnant par arbre de décision

TreeTagger et le contexte

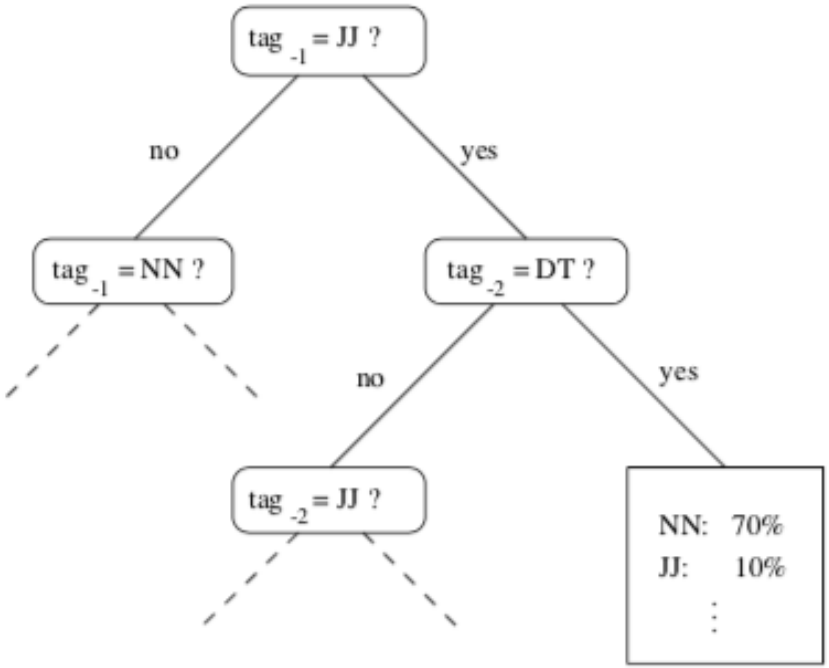

Figure 1: A sample decision tree (partially drawn).

TreeTagger (**TreeTagger**)

- un modèle de Markov d'ordre k, sélection du contexte conditionnant par arbre de décision
- gestion des mots inconnus avec un arbre de suffixes et de préfixes (façon de capturer la présence ou non d'une majuscule en début de mot)

TreeTagger et les mots inconnus

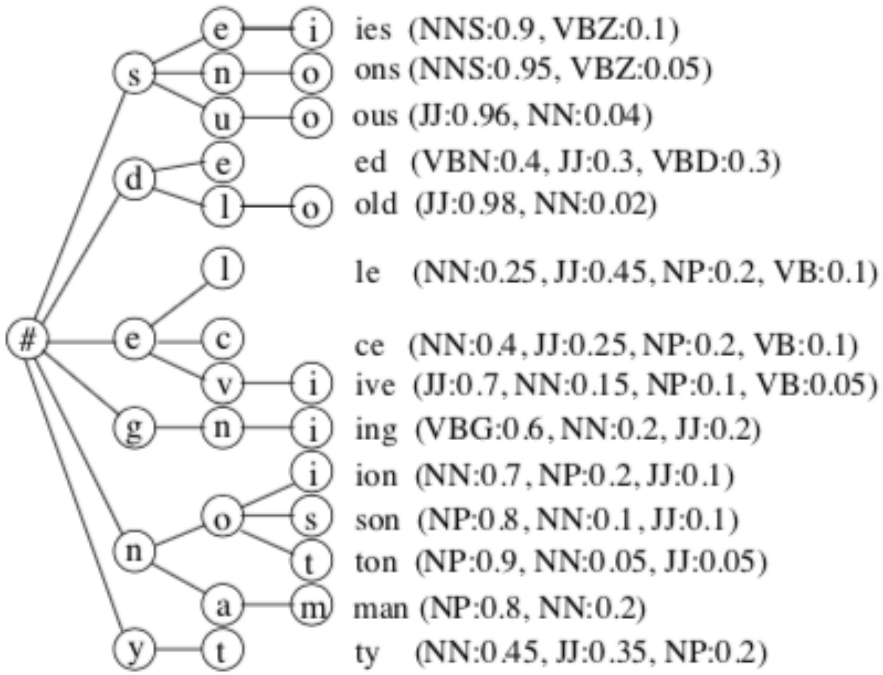

Figure 2: A sample suffix tree of maximal length 3.

TreeTagger (**TreeTagger**)

- un modèle de Markov d'ordre k, sélection du contexte conditionnant par arbre de décision
- gestion des mots inconnus avec un arbre de suffixes et de préfixes (façon de capturer la présence ou non d'une majuscule en début de mot)
- utilisation de la classe des mots dans le calcul des comptes $|t, w|$ en regroupant tous les mots ayant exactement les mêmes tags dans le train

Plan

Un article intéressant (Plank, Søgaard et Goldberg, 2016)

- modèle neuronal
  - plongement des mots et des caractères (pas les premiers à faire cela)
  - entraînement *multi-tâche* : prédire l'étiquette **et** la fréquence (*bin*)
    - **intuition :** donner au modèle la possibilité d'apprendre des distributions différentes selon la fréquence des mots

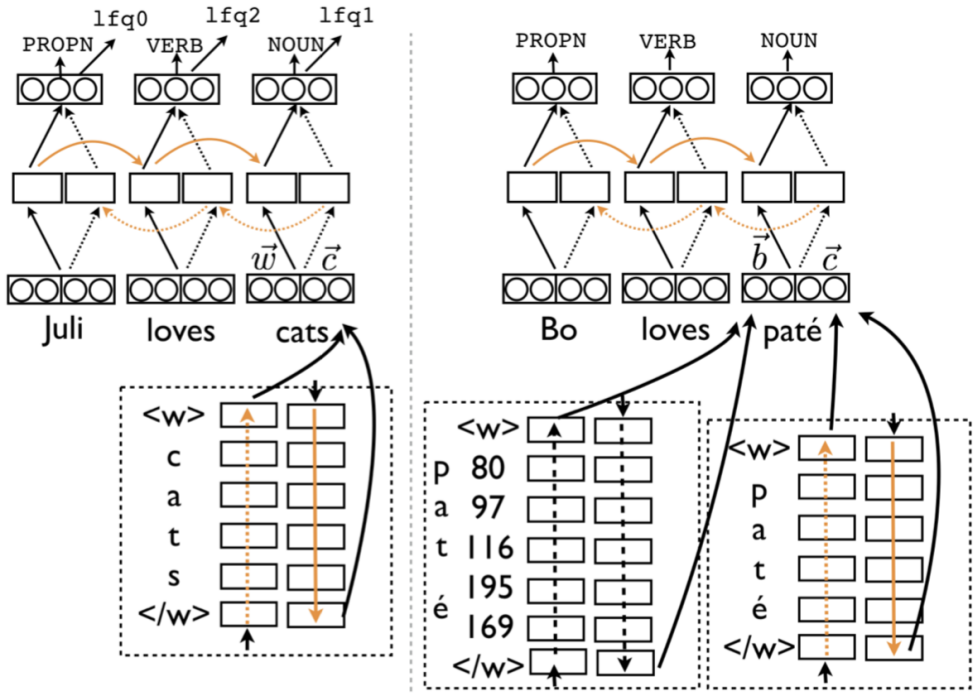

Figure 1: Right: bi-LSTM, illustrated with $\vec{b} + \vec{c}$ (bytes and characters), for $\vec{w} + \vec{c}$ replace $\vec{b}$ with words $\vec{w}$. Left: FREQBIN, our multi-task bi-LSTM that predicts at every time step the tag and the frequency class for the next token.

Un article intéressant (Plank, Søgaard et Goldberg, 2016)

- modèle neuronal
  - plongement des mots et des caractères (pas les premiers à faire cela)
  - entraînement *multi-tâche* : prédire l'étiquette **et** la fréquence (*bin*)
    - **intuition :** donner au modèle la possibilité d'apprendre des distributions différentes selon la fréquence des mots
- test sur :

  UD1.2 Universal Dependency project
  - $|tagset| = 17$
  - **22 langues** avec $|\text{train}| \geq 60k$ mots

  PTB Penn Tree Bank
  - $|tagset| = 45$

Un article intéressant (Plank, Søgaard et Goldberg, 2016)

- modèle neuronal
  - plongement des mots et des caractères (pas les premiers à faire cela)
  - entraînement *multi-tâche* : prédire l'étiquette **et** la fréquence (*bin*)
    - **intuition :** donner au modèle la possibilité d'apprendre des distributions différentes selon la fréquence des mots
- test sur :

  UD1.2 Universal Dependency project
  - $|tagset| = 17$
  - **22 langues** avec $|\text{train}| \geq 60k$ mots

  PTB Penn Tree Bank
  - $|tagset| = 45$
- comparaison à :

  TNT HMM + suffix tree pour les mots inconnus

  Plank:2014 CRF

  Bi-LSTM combinant représentation des mots et des caractères + quelques twists

Sur UD 1.2

	BASELINES		BI-LSTM using:				$\vec{w}+\vec{c}$ +POLYGLOT		OOV ACC		BTS
	TNT	CRF	$\vec{w}$	$\vec{c}$	$\vec{c}+\vec{b}$	$\vec{w}+\vec{c}$	bi-LSTM	FREQBIN	bi-LSTM	FREQBIN	
avg	94.61	94.27	92.37	94.29	94.01	96.08†	**96.50**	**96.50**	87.80	87.98	95.70
Indoeur.	94.70	94.58	92.72	94.58	94.28	96.24†	**96.63**	96.61	87.47	87.63	–
non-Indo.	94.57	93.62	91.97	93.51	93.16	95.70†	96.21	**96.28**	90.26	90.39	–
Germanic	93.27	93.21	91.18	92.89	92.59	94.97†	**95.55**	95.49	85.58	85.45	–
Romance	95.37	95.53	94.71	94.76	94.49	95.63†	**96.93**	**96.93**	85.84	86.07	–
Slavic	95.64	94.96	91.79	96.45	96.26	97.23†	97.42	**97.43**	91.48	91.69	–
ar	97.82	97.56	95.48	98.68	98.43	98.89	98.87	**98.91**	95.90	96.21	–
bg	96.84	96.36	95.12	97.89	97.78	**98.25**	98.23	90.06	90.06	90.56	97.84
cs	96.82	96.56	93.77	96.38	96.08	97.93	**98.02**	97.89	91.65	91.30	98.50
da	94.29	93.83	91.96	95.12	94.88	95.94	96.16	**96.35**	86.13	86.35	95.52
de	92.64	91.38	90.33	90.02	90.11	93.11	**93.51**	93.38	85.37	86.77	92.87
en	92.66	93.35	92.10	91.62	91.57	94.61	**95.17**	95.16	80.28	80.11	93.87
es	94.55	94.23	93.60	93.06	92.29	95.34	95.67	**95.74**	79.26	79.27	95.80
eu	93.35	91.63	88.00	92.48	92.72	94.91	95.38	**95.51**	83.55	84.30	–
fa	95.98	95.65	95.31	95.82	95.03	96.89	**97.60**	97.49	88.82	89.05	96.82
fi	93.59	90.32	87.95	90.25	89.15	95.18	95.74	**95.85**	88.35	88.85	95.48
fr	94.51	95.14	94.44	94.39	93.69	96.04	**96.20**	96.11	82.79	83.54	95.75
he	93.71	93.63	93.97	93.74	93.58	95.92	96.92	**96.96**	88.75	88.83	–
hi	94.53	96.00	95.99	93.40	92.99	96.64	96.97	**97.10**	83.98	85.27	–
hr	94.06	93.16	89.24	95.32	94.47	95.59	96.27	**96.82**	90.50	92.71	–
id	93.16	92.96	90.48	91.37	91.46	92.79	93.32	**93.41**	88.03	87.67	92.85
it	96.16	96.43	96.57	95.62	95.77	97.64	97.90	**97.95**	89.15	89.15	97.56
nl	88.54	90.03	84.96	89.11	87.74	92.07	92.82	**93.30**	78.61	75.95	–
no	96.31	96.21	94.39	95.87	95.75	97.77	**98.06**	98.03	93.56	93.75	–
pl	95.57	93.96	89.73	95.80	96.19	96.62	**97.63**	97.62	95.00	94.94	–
pt	96.27	96.32	94.24	95.96	96.2	97.48	**97.94**	97.90	92.16	92.33	–
sl	94.92	94.77	91.09	96.87	96.77	97.78	**96.97**	96.84	90.19	88.94	–
sv	95.19	94.45	93.32	95.57	95.5	96.30	96.60	**96.69**	89.53	89.80	95.57

Table 2: Tagging accuracies on UD 1.2 test sets. $\vec{w}$: words, $\vec{c}$: characters, $\vec{b}$: bytes. Bold/†: best accuracy/representation; +POLYGLOT: using pre-trained embeddings. FREQBIN: our multi-task model. OOV ACC: accuracies on OOVs. BTS: best results in Gillick et al. (2016) (not strictly comparable).

rali

Sur UD 1.2

	BASELINES		BI-LSTM using:				$\vec{w}+\vec{c}$ +POLYGLOT		OOV ACC		BTS
	TNT	CRF	$\vec{w}$	$\vec{c}$	$\vec{c}+\vec{b}$	$\vec{w}+\vec{c}$	bi-LSTM	FREQBIN	bi-LSTM	FREQBIN	
avg	94.61	94.27	92.37	94.29	94.01	96.08†	**96.50**	**96.50**	87.80	87.98	95.70
Indoeur.	94.70	94.58	92.72	94.58	94.28	96.24†	**96.63**	96.61	87.47	87.63	–
non-Indo.	94.57	93.62	91.97	93.51	93.16	95.70†	96.21	**96.28**	90.26	90.39	–
Germanic	93.27	93.21	91.18	92.89	92.59	94.97†	**95.55**	95.49	85.58	85.45	–
Romance	95.37	95.53	94.71	94.76	94.49	95.63†	**96.93**	**96.93**	85.84	86.07	–
Slavic	95.64	94.96	91.79	96.45	96.26	97.23†	97.42	**97.43**	91.48	91.69	–
ar	97.82	97.56	95.48	98.68	98.43	98.89	98.87	**98.91**	95.90	96.21	–
bg	96.84	96.36	95.12	97.89	97.78	**98.25**	98.23	90.06	90.06	90.56	97.84
cs	96.82	96.56	93.77	96.38	96.08	97.93	**98.02**	97.89	91.65	91.30	98.50
da	94.29	93.83	91.96	95.12	94.88	95.94	96.16	**96.35**	86.13	86.35	95.52
de	92.64	91.38	90.33	90.02	90.11	93.11	**93.51**	93.38	85.37	86.77	92.87
en	92.66	93.35	92.10	91.62	91.57	94.61	**95.17**	95.16	80.28	80.11	93.87
es	94.55	94.23	93.60	93.06	92.29	95.34	95.67	**95.74**	79.26	79.27	95.80
eu	93.35	91.63	88.00	92.48	92.72	94.91	95.38	**95.51**	83.55	84.30	–
fa	95.98	95.65	95.31	95.82	95.03	96.89	**97.60**	97.49	88.82	89.05	96.82
fi	93.59	90.32	87.95	90.25	89.15	95.18	95.74	**95.85**	88.35	88.85	95.48
fr	94.51	95.14	94.44	94.39	93.69	96.04	**96.20**	96.11	82.79	83.54	95.75
he	93.71	93.63	93.97	93.74	93.58	95.92	96.92	**96.96**	88.75	88.83	–
hi	94.53	96.00	95.99	93.40	92.99	96.64	96.97	**97.10**	83.98	85.27	–
hr	94.06	93.16	89.24	95.32	94.47	95.59	96.27	**96.82**	90.50	92.71	–
id	93.16	92.96	90.48	91.37	91.46	92.79	93.32	**93.41**	88.03	87.67	92.85
it	96.16	96.43	96.57	95.62	95.77	97.64	97.90	**97.95**	89.15	89.15	97.56
nl	88.54	90.03	84.96	89.11	87.74	92.07	92.82	**93.30**	78.61	75.95	–
no	96.31	96.21	94.39	95.87	95.75	97.77	**98.06**	98.03	93.56	93.75	–
pl	95.57	93.96	89.73	95.80	96.19	96.62	**97.63**	97.62	95.00	94.94	–
pt	96.27	96.32	94.24	95.96	96.2	97.48	**97.94**	97.90	92.16	92.33	–
sl	94.92	94.77	91.09	96.87	96.77	97.78	**96.97**	96.84	90.19	88.94	–
sv	95.19	94.45	93.32	95.57	95.5	96.30	96.60	**96.69**	89.53	89.80	95.57

Table 2: Tagging accuracies on UD 1.2 test sets. $\vec{w}$: words, $\vec{c}$: characters, $\vec{b}$: bytes. Bold/†: best accuracy/representation; +POLYGLOT: using pre-trained embeddings. FREQBIN: our multi-task model. OOV ACC: accuracies on OOVs. BTS: best results in Gillick et al. (2016) (not strictly comparable).

rali

Sur UD 1.2

	BASELINES		BI-LSTM using:				$\vec{w}+\vec{c}$ +POLYGLOT		OOV ACC		BTS
	TNT	CRF	$\vec{w}$	$\vec{c}$	$\vec{c}+\vec{b}$	$\vec{w}+\vec{c}$	bi-LSTM	FREQBIN	bi-LSTM	FREQBIN	
avg	94.61	94.27	92.37	94.29	94.01	96.08†	**96.5**0	**96.50**	87.80	87.98	95.70
Indoeur.	94.70	94.58	92.72	94.58	94.28	96.24†	**96.6**3	96.61	87.47	87.63	–
non-Indo.	94.57	93.62	91.97	93.51	93.16	95.70†	96.2	**96.28**	90.26	90.39	–
Germanic	93.27	93.21	91.18	92.89	92.59	94.97†	**95.5**5	95.49	85.58	85.45	–
Romance	95.37	95.53	94.71	94.76	94.49	95.63†	**96.9**3	**96.93**	85.84	86.07	–
Slavic	95.64	94.96	91.79	96.45	96.26	97.23†	97.4**3**	**97.43**	91.48	91.69	–
ar	97.82	97.56	95.48	98.68	98.43	98.89	98.8**1**	**98.91**	95.90	96.21	–
bg	96.84	96.36	95.12	97.89	97.78	**98.25**	98.2**5**	90.06	90.06	90.56	97.84
cs	96.82	96.56	93.77	96.38	96.08	97.93	**98.0**2	97.89	91.65	91.30	98.50
da	94.29	93.83	91.96	95.12	94.88	95.94	96.1**6**	**96.35**	86.13	86.35	95.52
de	92.64	91.38	90.33	90.02	90.11	93.11	**93.5**1	93.38	85.37	86.77	92.87
en	92.66	93.35	92.10	91.62	91.57	94.61	**95.1**7	95.16	80.28	80.11	93.87
es	94.55	94.23	93.60	93.06	92.29	95.34	95.6**7**	**95.74**	79.26	79.27	95.80
eu	93.35	91.63	88.00	92.48	92.72	94.91	95.3**8**	**95.51**	83.55	84.30	–
fa	95.98	95.65	95.31	95.82	95.03	96.89	**97.6**0	97.49	88.82	89.05	96.82
fi	93.59	90.32	87.95	90.25	89.15	95.18	95.7**4**	**95.85**	88.35	88.85	95.48
fr	94.51	95.14	94.44	94.39	93.69	96.04	**96.2**0	96.11	82.79	83.54	95.75
he	93.71	93.63	93.97	93.74	93.58	95.92	96.9**2**	**96.96**	88.75	88.83	–
hi	94.53	96.00	95.99	93.40	92.99	96.64	96.9**7**	**97.10**	83.98	85.27	–
hr	94.06	93.16	89.24	95.32	94.47	95.59	96.2**7**	**96.82**	90.50	92.71	–
id	93.16	92.96	90.48	91.37	91.46	92.79	93.3**3**	**93.41**	88.03	87.67	92.85
it	96.16	96.43	96.57	95.62	95.77	97.64	97.9**0**	**97.95**	89.15	89.15	97.56
nl	88.54	90.03	84.96	89.11	87.74	92.07	92.8**2**	**93.30**	78.61	75.95	–
no	96.31	96.21	94.39	95.87	95.75	97.77	**98.0**0	98.03	93.56	93.75	–
pl	95.57	93.96	89.73	95.80	96.19	96.62	**97.6**3	97.62	95.00	94.94	–
pt	96.27	96.32	94.24	95.96	96.2	97.48	**97.9**4	97.90	92.16	92.33	–
sl	94.92	94.77	91.09	96.87	96.77	97.78	**96.9**7	96.84	90.19	88.94	–
sv	95.19	94.45	93.32	95.57	95.5	96.30	96.6**0**	**96.69**	89.53	89.80	95.57

Table 2: Tagging accuracies on UD 1.2 test sets. $\vec{w}$: words, $\vec{c}$: characters, $\vec{b}$: bytes. Bold/†: best accuracy/representation; +POLYGLOT: using pre-trained embeddings. FREQBIN: our multi-task model. OOV ACC: accuracies on OOVs. BTS: best results in Gillick et al. (2016) (not strictly comparable).

rali

Sur le PTB

WSJ	Accuracy
Convnet (Santos and Zadrozny, 2014)	97.32
Convnet reimplementation (Ling et al., 2015)	96.80
Bi-RNN (Ling et al., 2015)	95.93
Bi-LSTM (Ling et al., 2015)	97.36
Our bi-LSTM $_{\vec{w}+\vec{c}}$	97.22

Table 3: Comparison POS accuracy on WSJ; bi-LSTM: 30 epochs, σ=0.3, no POLYGLOT.

- **note :** sur le PTB, le tagger de **Toutanova:2003** obtient 97.27%

Même veine (Yasunaga, Kasai et Radev, 2017)

- Étudient l'impact de l'**adversarial training Goodfellow:2015**

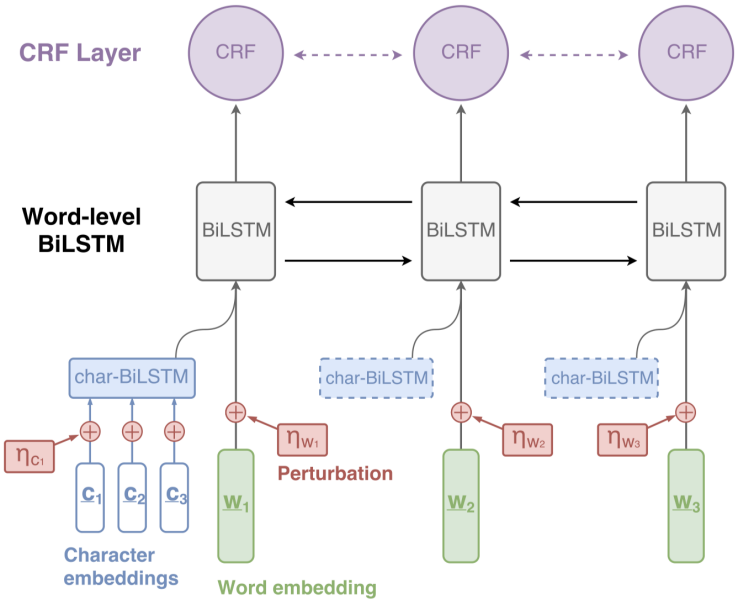

Figure 1: Illustration of our architecture for adversarial POS tagging. Given a sentence, we input the normalized word embeddings ($\underline{\boldsymbol{w}}_1, \underline{\boldsymbol{w}}_2, \underline{\boldsymbol{w}}_3$) and character embeddings (showing $\underline{\boldsymbol{c}}_1, \underline{\boldsymbol{c}}_2, \underline{\boldsymbol{c}}_3$ for w_1). Each word is represented by concatenating its word embedding and its character-level BiLSTM output. They are fed into the main BiLSTM-CRF network for POS tagging. In adversarial training, we compute and add the worst-case perturbation η to all the input embeddings for regularization.

- **idée :** générer des exemples trompeurs en perturbant légèrement des exemples du train de façon à dégrader au plus la fonction de perte du classificateur
- la fonction de perte inclut un terme pour ces éléments générés au fur et à mesure

Sur le PTB

Model	Accuracy
Toutanova et al. (2003)	97.27
Manning (2011)	97.28
Collobert et al. (2011)	97.29
Søgaard (2011)	97.50
Ling et al. (2015)	**97.78**
Ma and Hovy (2016)	97.55
Yang et al. (2017)	97.55
Hashimoto et al. (2017)	97.55
Ours – Baseline (BiLSTM-CRF)	97.54
Ours – Adversarial	97.58

Table 1: POS tagging accuracy on the PTB-WSJ test set, with other top-performing systems.

rali

Sur UD1.2

	leurs modèles		Plank, Søgaard *et* Goldberg, *2*		
	BiLSTM	+advers.	BiLSTM	TNT	CRF
$> 60k$ tokens	96.45	**96.65**	96.40	94.55	94.11
$< 60k$ tokens	91.20	**91.55**			

- Voir l'article pour une évaluation détaillée
  - les mots peu fréquents sont mieux modélisés
  - % de phrases correctement étiquetées :

    PTB 59.08% → 59.61%

    UD1.2 52.35% → 53.36%
  - améliore un analyseur syntaxique

Outch ! (**Liu:2018**)

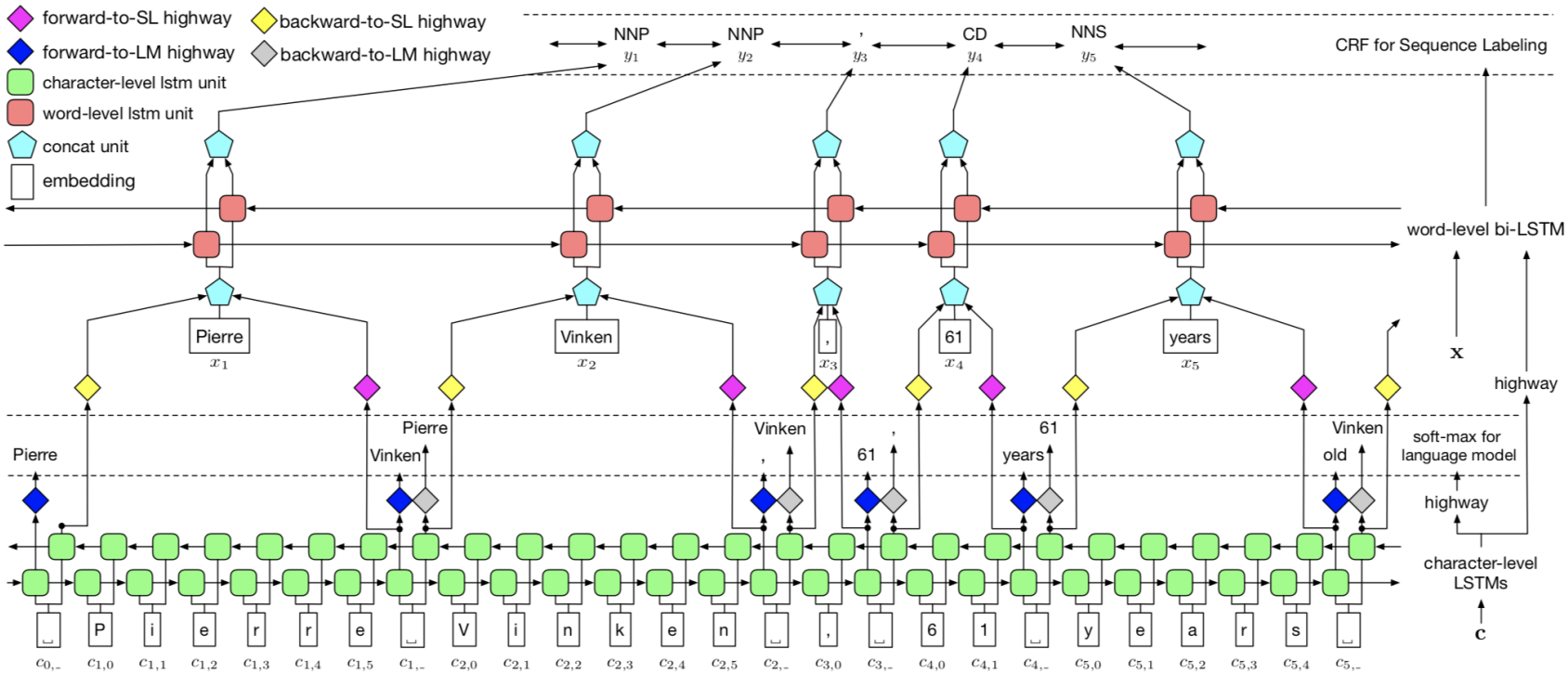

Figure 1: LM-LSTM-CRF Neural Architecture

Étiquetage sur le PTB

Ind & Model	Accuracy	
	Type	Value (±std)
0) Collobert et al. 2011†	reported	97.29
16) Manning 2011	reported	97.28
17) Søgaard 2011	reported	97.50
18) Sun 2014	reported	97.36
12) Rei 2017†‡	mean	96.97±0.22
	max	97.14
	reported	97.43
13) Lample et al. 2016†	mean±std	97.35±0.09
	maximum	97.51
14) Ma et al. 2016†	mean±std	97.42±0.04
	maximum	97.46
	reported	97.55
15) LM-LSTM-CRF †‡	mean±std	97.53±0.03
	maximum	97.59

Table 5: Accuracy on the WSJ dataset. We mark models adopting pre-trained word embedding as †, and record models which leverage language models as ‡.

rali

Plan

Taggeurs transformationnels (transformation-based taggers) [1]

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

- Deux composants :
  - patrons des transformations admissibles
  - apprentissage de l'ordonnancement des transformations

- taggeur populaire (open source) (Brill, 1992, 1995)

1. D'après (Manning et Schütze, 1999), p. 363

Les patrons du taggeur de Brill

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
- ligne 7 : 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 et 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

- Les triggers mettent en œuvre des étiquettes, des mots ou des traits sur les mots :
  - le mot courant est w et le tag qui suit est t
  - remplace NN par NNS si le mot courant se termine par s
- Beaucoup de latitude dans les règles que l'on peut apprendre
- c'est aussi un problème . . .

rali

Apprentissage des taggeurs transformationnels

In Un corpus taggé C_0
(ex : tag le plus fréquent pour chaque mot)

Out ordonnancement d'un sous-ensemble de règles :

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

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

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

rali

Application des règles de ré-écriture

- de la gauche vers la droite.
- immédiate ou retardée (Brill = retardée).
  Soit la règle $A \longrightarrow B$ *si* A *précède*

retardé $AAAA \rightarrow ABBB$
(on marque les transformations à effectuer, puis on les fait)
immédiat $AAAA \rightarrow ABAB$

- Brill reporte un taggeur appris de manière non supervisée (sans corpus taggé) avec un taux de 95.6%.
  - utilise l'information des mots non ambigus (qui possèdent un seul tag, selon le dictionnaire)
  - intuition : can dans The can is open sera taggé NN *(et non MD)*, si dans le contexte "ART — VB", les mots non ambigus sont majoritairement étiquetés NN.

Improving Part-of-Speech Tagging for NLP Pipelines

Jatav:2017

- comparent le parseur PCFG de Stanford **Klein:2003** à un tagger "maison" (pre + post processing)

	pcfg	rage
PTB-3 (4.5M de mots, journalistique)		
token-level	95.67	96.86
sentence-level	31.61	57.91
Reuters (21k mots, 110 articles de nouvelles)		
token-level	95.12	97.53
sentence-level	37.71	63.74
PubMed (12.5k mots de PubMed)		
token-level	91.91	96.36
sentence-level	25.54	56.62

Plan

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.
- Campagne d'évaluation : **CONLL'2000** (COmputational Natural Language Learning) :
  http://cnts.uia.ac.be/conll2000/chunking/

rali

Le corpus CONLL

B-tags marquent le début d'un groupe :

- B-NP marque le premier mot d'un groupe nominal (noun phrase) ;
- B-VP marque le début d'un groupe verbal, etc.

I-tags marquent un mot dans un groupe qui n'est pas le premier mot du groupe.

- I-NP indique qu'un mot est à l'intérieur d'un groupe nominal (d'au moins deux mots)

autres : O marque 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-NP* et *B-NP*, marquant respectivement le milieu d'un groupe nominal, et son début.

rali

Le corpus CONLL

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

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

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

- 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		

rali

Chunker = Tagger

- ex : la sortie d'un tagger "normal" constitue l'entrée d'un IOB-taggeur.

$$
\begin{aligned}
& \text{the deficit could narrow} \ldots \\
\longrightarrow \quad & \text{DT NN MD VB} \ldots \\
\longrightarrow \quad & \text{BOS-DT DT-NN NN-MD MD-VB} \ldots \\
\longrightarrow \quad & \text{B-NP I-NP B-VP I-VP} \ldots
\end{aligned}
$$

- variante de cette idée ("Shallow Parsing as Part-of-Speech Tagging") :

$$
\begin{aligned}
w_i &\underset{HMM-1}{\Rightarrow} POS_i \\
(w_i, POS_i) &\underset{HMM-2}{\Rightarrow} IOB_i \\
(POS_i, IOB_i, POS_{i+1}, IOB_{i+1}) &\underset{HMM-3}{\Rightarrow} I\hat{O}B_i
\end{aligned}
$$

Chunker = Tagger

- Si le corpus d'entraînement est suffisamment grand, on peut entraîner directement un taggeur avec le jeu d'étiquettes des IOB-tags

$$p(w|\text{iob-tag}) \text{ et } p(\text{iob-tag}|\text{iob-tag'})$$

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

CONLL 2000 : résultats

test data	precision	recall	$F_{\beta=1}$
Kudoh and Matsumoto	93.45%	93.51%	93.48
Van Halteren	93.13%	93.51%	93.32
Tjong Kim Sang	94.04%	91.00%	92.50
Zhou, Tey and Su	91.99%	92.25%	92.12
Déjean	91.87%	91.31%	92.09
Koeling	92.08%	91.86%	91.97
Osborne	91.65%	92.23%	91.94
Veenstra and Van den Bosch	91.05%	92.03%	91.54
Pla, Molina and Prieto	90.63%	89.65%	90.14
Johansson	86.24%	88.25%	87.23
Vilain and Day	88.82%	82.91%	85.76
baseline	72.58%	82.14%	77.07

Table 2: Performance of the eleven systems on the test data. The baseline results have been obtained by selecting the most frequent chunk tag for each part-of-speech tag.

rali

LM-BiLSTM-CRF (**Liu:2018**)

Extra Resource	Ind & Model	F_1 score	
		Type	Value (±std)
PTB-POS	19) Hashimoto et al. 2016†	reported	95.77
	20) Søgaard et al. 2016†	reported	95.56
CoNLL 2000 / PTB-POS dataset	3)Yang et al. 2017†	reported	95.41
1B Word dataset	4) Peters et al. 2017†‡	reported	96.37±0.05
None	21) Hashimoto et al. 2016†	reported	95.02
	22) Søgaard et al. 2016†	reported	95.28
	9) Yang et al. 2017†	reported	94.66
	12) Rei 2017†‡	mean	94.24±0.11
		max	94.33
		reported	93.88
	13) Lample et al. 2016†	mean	94.37±0.07
		maximum	94.49
	14) Ma et al. 2016†	mean	95.80±0.13
		maximum	95.93
	15) LM-LSTM-CRF †‡	mean	95.96±0.08
		maximum	96.13

Table 7: F_1 score on the CoNLL00 chunking dataset. We mark models adopting pre-trained word embedding as †, and record models which leverage language models as ‡.

D'autres schémas pour encoder les chunks

Tokens	IO	BIO	BMEWO	BMEWO+
Yesterday	O	O	O	BOS_O
afternoon	0	O	O	O
,	0	O	O	O_PER
John	I_PER	B_PER	B_PER	B_PER
J	I_PER	I_PER	M_PER	M_PER
.	I_PER	I_PER	M_PER	M_PER
Smith	I_PER	I_PER	E_PER	E_PER
traveled	O	0	O	PER_O
to	O	O	O	O_LOC
Washington	I_LOC	B_LOC	W_LOC	W_LOC
.	O	O	O	O_EOS

LingPipe

Plan

(Tjong Kim Sang et De Meulder, 2003)

- Reconnaître les entités nommées (*named-entities*) dans un texte consiste à identifier des mentions et à les classer à l'aide d'un jeu prédéfini d'étiquettes.
  1. **Rangers** yank **Price** team off **Stanley Cup** in **New York**
  2. [org Rangers] yank [per Price] team off [misc Stanley Cup] in [loc New York]
- Préliminaire à l'extraction d'information.
- Très étudié (benchmarks disponibles) :
  - règles
  - CRF (Finkel, Grenager et Manning, 2005)
  - perceptrons (Ratinov et Roth, 2009)
  - réseaux de neurones (Collobert et al., 2011) ; (Lample et al., 2016) ; (Chiu et Nichols, 2016)
- Mais :
  - corpus annotés rares
  - adaptation à un autre domaine déficiente (Augenstein, Derczynski et Bontcheva, 2017 ; Onal et Karagoz, 2015)

CONLL 2003 (Tjong Kim Sang et De Meulder, 2003)

- la tâche partagée de CONLL en 2002 était la reconnaissance d'entités nommées pour l'espagnol et le hollandais (Sang, 2002)
- celle de 2003 était dédiée à l'anglais et l'allemand. Le corpus anglais est encore utilisé pour *benchmarker* des approches

per	personnes
org	organisations
loc	lieu
misc	toute autre entité nommée

CONLL 2003 (Tjong Kim Sang et De Meulder, 2003)

- En anglais, les données proviennent du corpus Reuters : des nouvelles publiées entre 1996 et 1997
  - bcp de nouvelles sportives qui sont un peu particulières

English data	Articles	Sentences	Tokens
Training set	946	14,987	203,621
Development set	216	3,466	51,362
Test set	231	3,684	46,435

German data	Articles	Sentences	Tokens
Training set	553	12,705	206,931
Development set	201	3,068	51,444
Test set	155	3,160	51,943

Table 1: Number of articles, sentences and tokens in each data file.

English data	LOC	MISC	ORG	PER
Training set	7140	3438	6321	6600
Development set	1837	922	1341	1842
Test set	1668	702	1661	1617

German data	LOC	MISC	ORG	PER
Training set	4363	2288	2427	2773
Development set	1181	1010	1241	1401
Test set	1035	670	773	1195

Table 2: Number of named entities per data file

- + données non annotées du même domaine (17M de mots en anglais, 14M en allemand)

CONLL2003

```
Takuya NNP B-NP B-PER
Takagi NNP I-NP I-PER
headed VBD B-VP O
the DT B-NP O
winner NN I-NP O
in IN B-PP O
the DT B-NP O
88th JJ I-NP O
minute NN I-NP O
of IN B-PP O
the DT B-NP O
group NN I-NP O
C NNP I-NP O
game NN I-NP O
after IN B-PP O
goalkeeper NN B-NP O
Salem NNP I-NP B-PER
Bitar NNP I-NP I-PER
spoiled JJ I-NP O
a DT I-NP O
mistake-free NN I-NP O
display NN I-NP O
by IN B-PP O
allowing VBG B-VP O
the DT B-NP O
ball NN I-NP O
to TO B-VP O
slip VB I-VP O
under IN B-PP O
his PRP$ B-NP O
body NN I-NP O
. . O O
```

CONLL 2003 : anglais

- tâche : identification exacte des entités du test

English test	Precision	Recall	$F_{\beta=1}$
⋆ Florian	88.99%	88.54%	88.76±0.7
⋆ Chieu	88.12%	88.51%	88.31±0.7
⋆ Klein	85.93%	86.21%	86.07±0.8
Zhang	86.13%	84.88%	85.50±0.9
Carreras (b)	84.05%	85.96%	85.00±0.8
Curran	84.29%	85.50%	84.89±0.9
Mayfield	84.45%	84.90%	84.67±1.0
Carreras (a)	85.81%	82.84%	84.30±0.9
⋆ McCallum	84.52%	83.55%	84.04±0.9
Bender	84.68%	83.18%	83.92±1.0
Munro	80.87%	84.21%	82.50±1.0
Wu	82.02%	81.39%	81.70±0.9
⋆ Whitelaw	81.60%	78.05%	79.78±1.0
Hendrickx	76.33%	80.17%	78.20±1.0
De Meulder	75.84%	78.13%	76.97±1.2
Hammerton	69.09%	53.26%	60.15±1.3
Baseline	71.91%	50.90%	59.61±1.2

HMM + TBL + MaxEnt + RRM
MaxEnt seul
MaxEnt + HMM + CNN
RRM
SVM + HMM
AdaBoost.MH
CRF
HMM sur les caractères ++
Memory-based Learning
LSTM
grep des entités du train (avec une seule classe)

- un vote des 5 systèmes (⋆ déterminés sur dev) donne en anglais une f-mesure (F_1) de 90.3

CONLL 2003 : traits employés

	G	U	E	English	German
Zhang	+	-	-	19%	15%
Florian	+	-	+	27%	5%
Hammerton	+	-	-	22%	-
Carreras (a)	+	-	-	12%	8%
Chieu	+	-	-	17%	-
Hendrickx	+	+	-	7%	5%
De Meulder	+	+	-	8%	3%
Bender	+	+	-	3%	6%
Curran	+	-	-	1%	-
McCallum	+	+	-	?	?
Wu	+	-	-	?	?

Table 4: Error reduction for the two development data sets when using extra information like gazetteers (G), unannotated data (U) or externally developed named entity recognizers (E). The lines have been sorted by the sum of the reduction percentages for the two languages.

rali

Ontonotes 5.0

- multilingue (Chinois, Arabe, Anglais)
- annotations multiples (syntaxe, sens des mots, coréférence, NER, etc.)
- multiples genres

  625k newswrire
  300k webdata
  200k broadcast conversation
  200k braodcast news
  120k magazine

- 18 étiquettes : DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL, LANGUAGE, LAW, EVENT, PRODUCT, WORK_OF_ART, LOC, GPE, ORG, FAC, NORP, PERSON

Moins populaires

MUC-6 *newswire* du journal de Wall Street : per, loc, org + entités temporelles

WikiGold 40k mots d'articles Wikipedia tirés aléatoirement de certains domaines : per, loc, org et misc (Balasuriya et al., 2009)

Web 8k mots de 20 pages Web sur différents sujets : per, loc, org et misc (Ratinov et Roth, 2009)

Tweet 34k mots de 2400 tweets, 10 étiquettes (Ritter et al., 2011)

i2b2 29k mentions, dossiers de patients : 30 étiquettes patient, doctor, username, profession, room, hospital, etc.

rali

Extraits (automatiquement) de Wikipedia

WINER 3.2M articles de Wikipedia (2013), 1.3G mots, 54M phrases (41M avec au moins une annotation) : per, loc, org et misc (Ghaddar et Langlais, 2017)

WiFine les mêmes textes, mais 157.4M de mentions avec des annotations plus fines (112 ou 89 étiquettes) (Ghaddar et Langlais, 2018b)

Quelques systèmes populaires

- NeuroNER (python) facile d'emploi, très bonne documentation
- Spacy (python) neuronal depuis la version 2.
- CogComp (Illinois) (java) perceptron, gazetteers, traits spécifiques
- Stanford NER (java) CRF

(Lample et al., 2016)

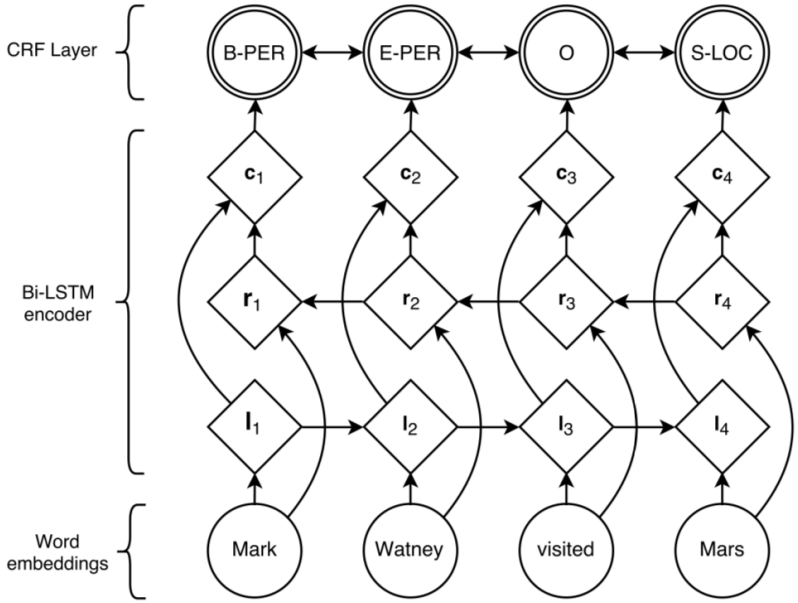

Figure 1: Main architecture of the network. Word embeddings are given to a bidirectional LSTM. $\mathbf{l}_i$ represents the word i and its left context, $\mathbf{r}_i$ represents the word i and its right context. Concatenating these two vectors yields a representation of the word i in its context, $\mathbf{c}_i$.

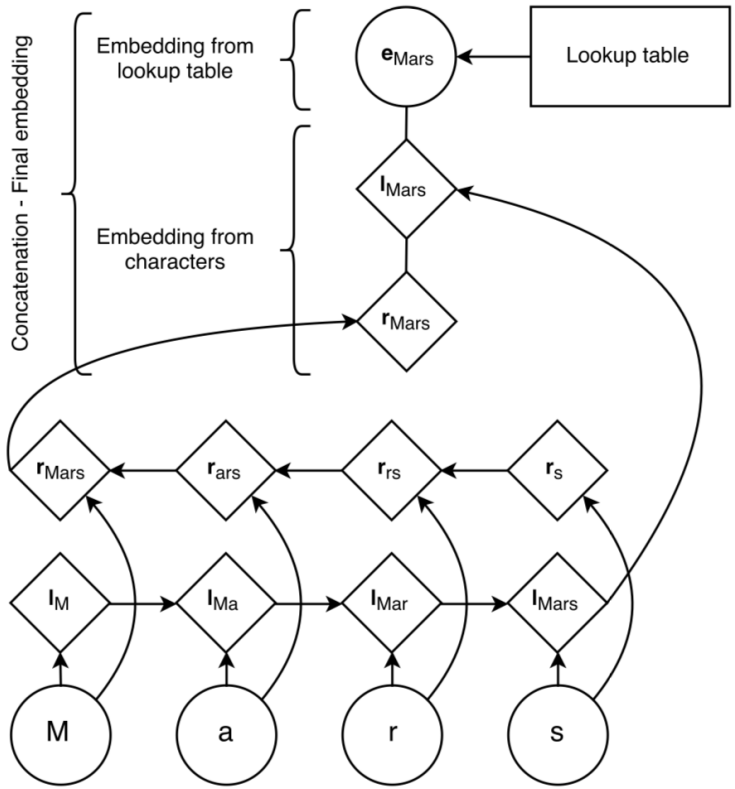

Figure 4: The character embeddings of the word "Mars" are given to a bidirectional LSTMs. We concatenate their last outputs to an embedding from a lookup table to obtain a representation for this word.

(Lample et al., 2016) sur CONLL 2003

Model	F_1
Collobert et al. (2011)*	89.59
Lin and Wu (2009)	83.78
Lin and Wu (2009)*	90.90
Huang et al. (2015)*	90.10
Passos et al. (2014)	90.05
Passos et al. (2014)*	90.90
Luo et al. (2015)* + gaz	89.9
Luo et al. (2015)* + gaz + linking	**91.2**
Chiu and Nichols (2015)	90.69
Chiu and Nichols (2015)*	90.77
LSTM-CRF (no char)	90.20
LSTM-CRF	**90.94**
S-LSTM (no char)	87.96
S-LSTM	90.33

Table 1: English NER results (CoNLL-2003 test set). * indicates models trained with the use of external labeled data

Model	Variant	F_1
LSTM	char + dropout + pretrain	89.15
LSTM-CRF	char + dropout	83.63
LSTM-CRF	pretrain	88.39
LSTM-CRF	pretrain + char	89.77
LSTM-CRF	pretrain + dropout	90.20
LSTM-CRF	pretrain + dropout + char	**90.94**
S-LSTM	char + dropout	80.88
S-LSTM	pretrain	86.67
S-LSTM	pretrain + char	89.32
S-LSTM	pretrain + dropout	87.96
S-LSTM	pretrain + dropout + char	90.33

Table 5: English NER results with our models, using different configurations. "pretrain" refers to models that include pre-trained word embeddings, "char" refers to models that include character-based modeling of words, "dropout" refers to models that include dropout rate.

rali

En pratique...

- Les approches (neuronales) état de l'art utilisent **aussi** des traits :

capitalisation	`allCaps`, `upperInitial`, `lowercase`, `mixedCaps`, `noinfo` (Collobert et al., 2011)
char	`upper case`, `lower case`, `punctuation`, `other`
gazeteers	listes d'ENs connues de chaque type ($>$ 2.3M entrées)

Text	Hayao	Tada	,	commander	of	the	Japanese	North	China	Area	Army
LOC	–	–	–	–	–	B	I	–	S	–	–
MISC	–	–	–	S	B	B	I	S	S	S	S
ORG	–	–	–	–	–	B	I	B	I	I	E
PERS	B	E	–	–	–	–	–	–	S	–	–

Figure 4: Example of how lexicon features are applied. The B, I, E, markings indicate that the token matches the Begin, Inside, and End token of an entry in the lexicon. S indicates that the token matches a single-token entry.

(Chiu et Nichols, 2016) Fig. 4

(Peters et al., 2017)

- Utilise un modèle de langue pour **encoder** le contexte de prédiction.

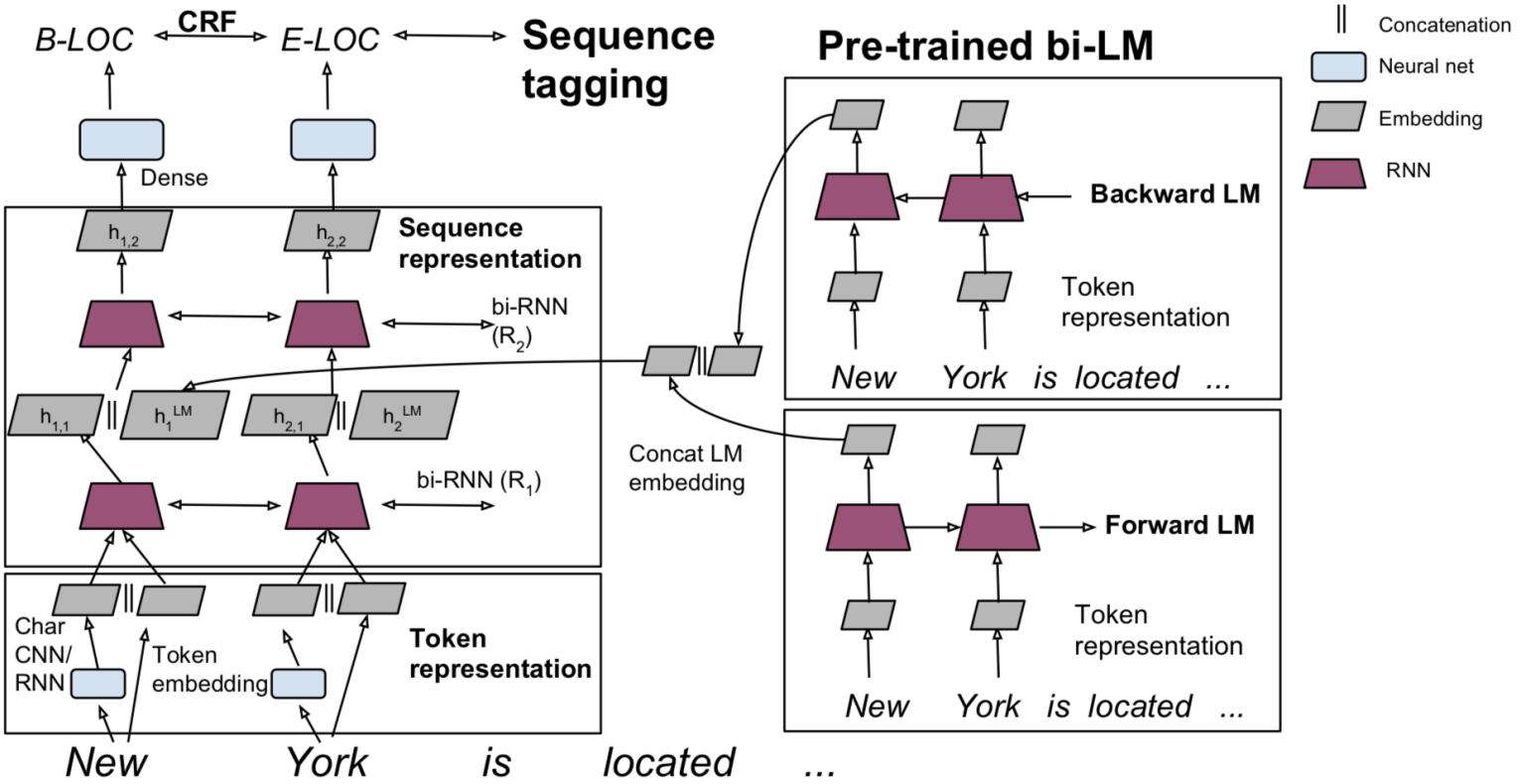

Figure 2: Overview of TagLM, our language model augmented sequence tagging architecture. The top level embeddings from a pre-trained bidirectional LM are inserted in a stacked bidirectional RNN sequence tagging model. See text for details.

(Peters et al., 2017)

- modèle de langue entraîné sur le 1B word corpus (3 semaines sur 32 GPUs)
- testé sur CONLL 2003 (NER) et CONLL 2000 (chunking)

Model	$F_1 \pm$ **std**
Chiu and Nichols (2016)	90.91 ± 0.20
Lample et al. (2016)	90.94
Ma and Hovy (2016)	91.37
Our baseline without LM	90.87 ± 0.13
TagLM	$\mathbf{91.93 \pm 0.19}$

Table 1: Test set F_1 comparison on CoNLL 2003 NER task, using only CoNLL 2003 data and unlabeled text.

Model	$F_1 \pm$ **std**
Yang et al. (2017)	94.66
Hashimoto et al. (2016)	95.02
Søgaard and Goldberg (2016)	95.28
Our baseline without LM	95.00 ± 0.08
TagLM	$\mathbf{96.37 \pm 0.05}$

Table 2: Test set F_1 comparison on CoNLL 2000 Chunking task using only CoNLL 2000 data and unlabeled text.

- ! ajoutent le dev au train

So what ? (Ghaddar et Langlais, 2018a)

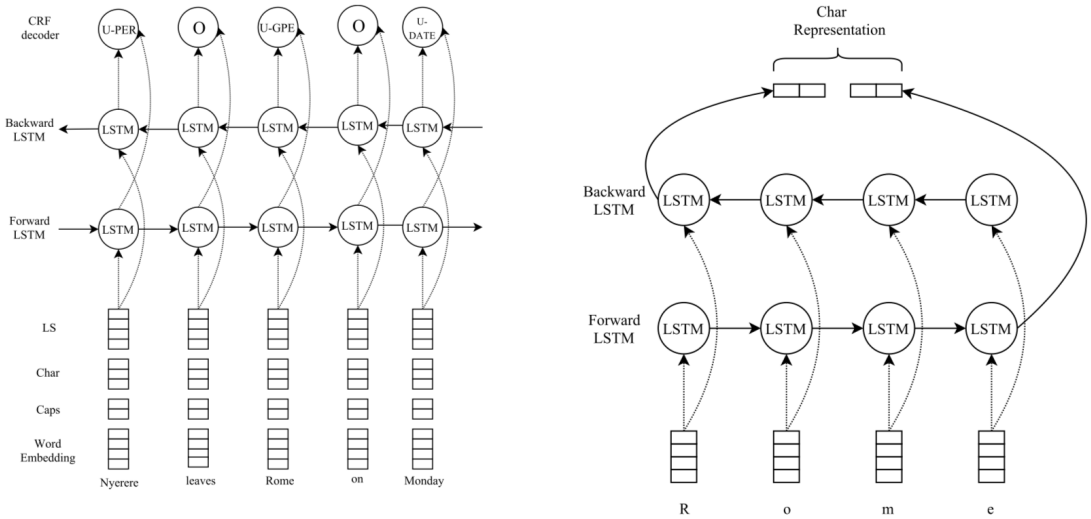

Figure 3: **Left Figure:** Main architecture of our NER system. **Right Figure:** Character representation of the word "Roma" given to the word-level bi-LSTM.

So what ? (Ghaddar et Langlais, 2018a)

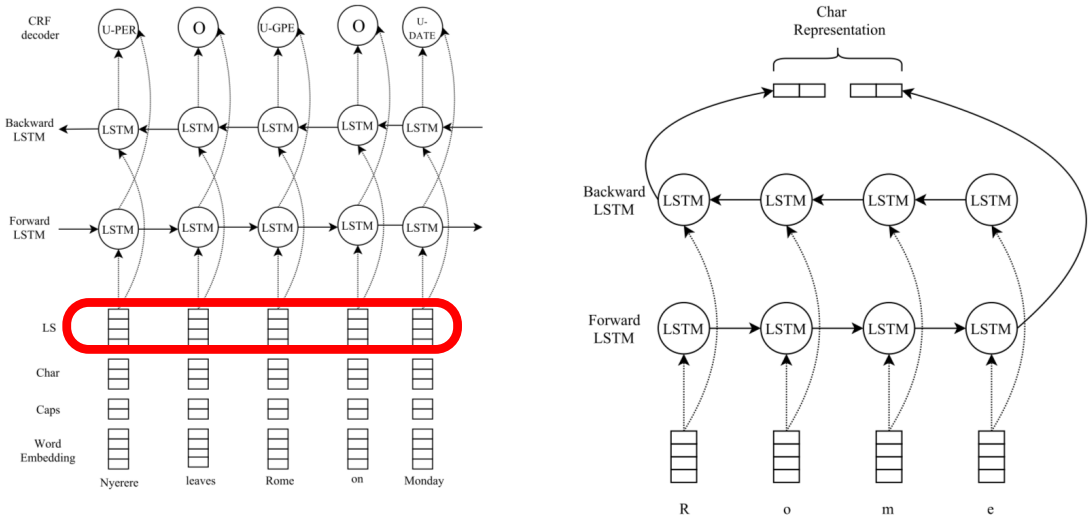

Figure 3: **Left Figure:** Main architecture of our NER system. **Right Figure:** Character representation of the word "Roma" given to the word-level bi-LSTM.

- une représentation fixe (dim. 120) qui capture le contexte,
- apprise sur WiFiNE à l'aide de `FastText`.

So what ? (Ghaddar et Langlais, 2018a)

(v1) On **October 9, 2009**, the **Norwegian Nobel Committee** announced that **Obama** had won the **2009 Nobel Peace Prize**.
(v2) On /date, the /organization/government_agency announced that /person/politician had won the /award.

- **idée :** plonger mots et étiquettes dans le même espace
- **note :**les étiquettes dans WiFiNE ont été obtenues automatiquement de Wikipedia (*distant supervision*)

(Ghaddar et Langlais, 2018a) : représentation LS

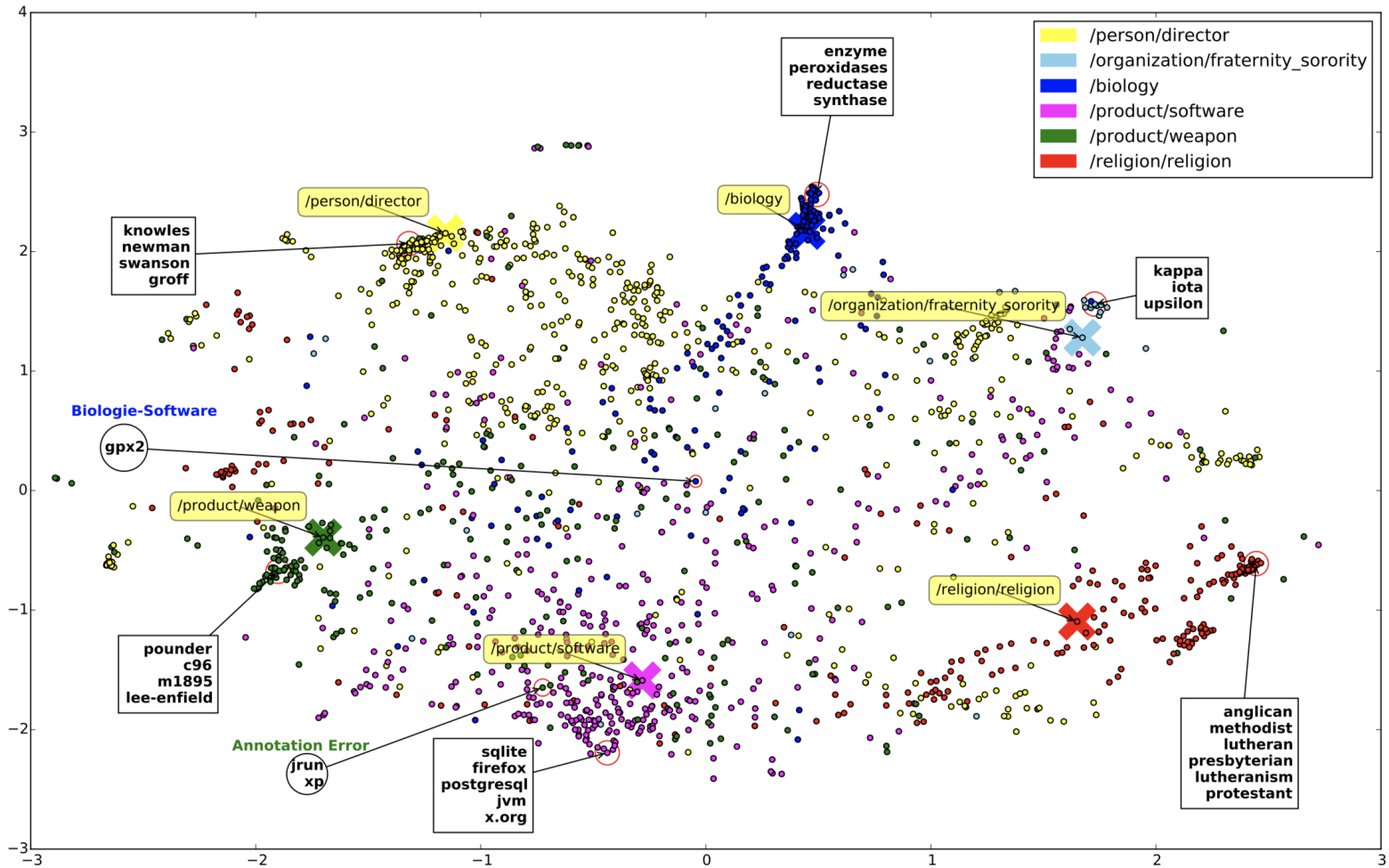

Figure 2: Two-dimensional representation of the vector space which embeds both words and entity typ
Big Xs indicate entity types, while circles refer to words (i.e. named-entities, here).

(Ghaddar et Langlais, 2018a) : représentation LS

Word	Entity Type	Sim	Word	Entity Type	Sim
hilton	/building/hotel	0.58	located	/location	0.47
	/building/restaurant	0.46		/location/city	0.44
	/person/actor	0.37		/building	0.40
gpx2	/biology	0.69	directed	/person/director	0.60
	/product/software	0.56		/art/film	0.55
jrun	/product/software	0.64	in	/date	0.58
	/product/weapon	0.23		/location/city	0.54
dammstadt	/location/city	0.45	won	/award	0.53
	/location/railway	0.44		/event/sports_event	0.53

Table 1: Topmost similar entity types to a few single-word mentions (left table) and non-entity words (right table).

- chaque mot est associée à un vecteur de dim. 120 (le nombre de types dans WiFiNE), chaque valeur (entre -1 et 1) est le cosine entre la représentation du mot et le type associé
- dammstadt (qui n'a pas sa page Wikipedia) et qui est vu 5 fois dans WiFiNE

(Ghaddar et Langlais, 2018a) : CONLL 2003

Model	LEX	GAZ	CAP	EMB	CHE	LME	LS	**F1**
(Finkel et al., 2005)	+	+	+	•	•	•	•	86.86
(Ratinov and Roth, 2009)	+	+	+	•	•	•	•	90.88
(Lin and Wu, 2009)	+	+	+	•	•	•	•	90.90
(Luo et al., 2015)	+	+	+	•	•	•	•	91.20
(Collobert et al., 2011)	•	+	+	+	•	•	•	89.56
(Huang et al., 2015)	•	•	+	+	+	•	•	90.10
(Lample et al., 2016)	•	•	+	+	+	•	•	90.94
(Ma and Hovy, 2016)	•	•	+	+	+	•	•	91.21
(Shen et al., 2017)	•	•	+	+	•	•	•	90.89
(Strubell et al., 2017)	•	•	+	+	•	•	•	90.54
(Tran et al., 2017)	•	•	+	+	+	+	•	91.69
(Liu et al., 2017)	•	•	+	+	+	+	•	91.71
This work	•	+	+	+	+	•	+	**91.73**

Table 4: F1 scores on the CoNLL test set. The first four systems are feature-based, the others are neuronal. The feature configuration of each system is encoded with: LEX which stands for LEXical feature, GAZ for GAZetteers, CAP for CAPitalization, EMB for pre-trained EMBeddings, CHE for CHaracter Embeddings, LME for Language Model Embeddings, and LS for the proposed LS feature representation. + indicates that the model uses the feature set.

(Ghaddar et Langlais, 2018a) : Ontonotes 5.0

Model	LEX	GAZ	CAP	EMB	CHE	LME	LS	**F1**
(Finkel and Manning, 2009)	+	+	+	•	•	•	•	82.42
(Ratinov and Roth, 2009)	+	+	+	•	•	•	•	84.88
(Passos et al., 2014)	+	+	+	•	•	•	•	82.24
(Durrett and Klein, 2014)	+	+	+	•	•	•	•	84.04
(Chiu and Nichols, 2016)	•	+	+	+	+	•	•	86.28
(Shen et al., 2017)	•	•	+	+	+	•	•	86.52
(Strubell et al., 2017)	•	•	+	+	+	•	•	86.99
This work	•	+	+	+	+	•	+	**87.95**

Table 5: F1 scores on the ONTONOTES test set. The first four systems are feature-based, the following ones are neuronal. See Table 4 for an explanation of the column of features.

rali

(Ghaddar et Langlais, 2018a) : Ontonotes 5.0

Model	BC	BN	MZ	NW	TC	WB
(Finkel and Manning, 2009)	78.66	87.29	82.45	85.50	67.27	72.56
(Durrett and Klein, 2014)	78.88	87.39	82.46	87.60	72.68	76.17
(Chiu and Nichols, 2016)	85.23	89.93	84.45	88.39	72.39	78.38
This work	**86.33**	**90.46**	**85.91**	**89.75**	**75.41**	**80.39**

Table 6: Per-genre F1 scores on ONTONOTES (numbers taken from Chiu and Nichols (2016)). F = broadcast conversation, BN = broadcast news, MZ = magazine, NW = newswire, TC = telepho conversation, WB = blogs and newsgroups.

- des gains marqués sur les corpus les plus bruités

Plan

Spacy

```
import spacy

nlp = spacy.load('en_core_web_sm')
doc = nlp(u'Apple is looking at buying U.K. startup
    for $1 billion')

for token in doc:
    print("token=%-10s\tlemme=%-10s\tpos=%-10s\ttag
        =%-10s\tdep=%-10s\tshape=%-10s\tis_alpha=%d\
        tis_stop=%d" %
        (token.text, token.lemma_, token.pos_, token.
            tag_,
         token.dep_, token.shape_, token.is_alpha,
             token.is_stop))
```

```
token=Apple     lemme=apple     pos=PROPN   tag=NNP   dep=nsubj      shape=Xxxxx   is_alpha=1   is_stop=0
token=is        lemme=be        pos=VERB    tag=VBZ   dep=aux        shape=xx      is_alpha=1   is_stop=1
token=looking   lemme=look      pos=VERB    tag=VBG   dep=ROOT       shape=xxxx    is_alpha=1   is_stop=0
token=at        lemme=at        pos=ADP     tag=IN    dep=prep       shape=xx      is_alpha=1   is_stop=1
token=buying    lemme=buy       pos=VERB    tag=VBG   dep=pcomp      shape=xxxx    is_alpha=1   is_stop=0
token=U.K.      lemme=u.k.      pos=PROPN   tag=NNP   dep=compound   shape=X.X.    is_alpha=0   is_stop=0
token=startup   lemme=startup   pos=NOUN    tag=NN    dep=dobj       shape=xxxx    is_alpha=1   is_stop=0
token=for       lemme=for       pos=ADP     tag=IN    dep=prep       shape=xxx     is_alpha=1   is_stop=1
token=$         lemme=$         pos=SYM     tag=$     dep=quantmod   shape=$       is_alpha=0   is_stop=0
token=1         lemme=1         pos=NUM     tag=CD    dep=compound   shape=d       is_alpha=0   is_stop=0
token=billion   lemme=billion   pos=NUM     tag=CD    dep=pobj       shape=xxxx    is_alpha=1   is_stop=0
```

Bibliography I

- Abney, Steven (1991). **Parsing By Chunks**. Robert Berwick and Steven Abney and Carol Tenny, "Principle-Based Parsing", Kluwer Academic.
- Adda, Gilles et al. (1999). "The GRACE evaluation for POS tagging for French language". In : **Cahiers/Langues**. T. Vol. 2, Issue 2. http ://www.john-libbey-eurotext.fr/en/revues/lan/index.htm.
- Augenstein, Isabelle, Leon Derczynski et Kalina Bontcheva (2017). "Generalisation in named entity recognition : A quantitative analysis". In : **Computer Speech & Language** 44, p. 61–83.
- Balasuriya, Dominic et al. (2009). "Named Entity Recognition in Wikipedia". In : **Proceedings of the 2009 Workshop on The People's Web Meets NLP : Collaboratively Constructed Semantic Resources**. Association for Computational Linguistics, p. 10–18.

rali

Bibliography II

- Brill, Eric (1992). "A simple rule-based part-of-speech tagger". In : **Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing**. Trento, IT, p. 152–155.
- Brill, Eric (1995). "Transformation-Based Error-Driven Learning and Natural Language Processing : A Case Study in Part-of-Speech Tagging". In : **Computational Linguistics** 21.4, p. 543–565.
- Charniak, Eugene (1993). **Statistical Language Learning**. MIT Press.
- Chiu, Jason PC et Eric Nichols (2016). "Named entity recognition with bidirectional LSTM-CNNs". In : **Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics**.

rali

Bibliography III

- Collobert, R. et al. (2011). "Natural Language Processing (Almost) from Scratch". In : **Journal of Machine Learning Research** 12, p. 2493–2537.
- Finkel, Jenny Rose, Trond Grenager et Christopher Manning (2005). "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". In : **Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics**. ACL '05. Ann Arbor, Michigan : Association for Computational Linguistics, p. 363–370.
- Ghaddar, Abbas et Philippe Langlais (2017). "WiNER : A Wikipedia Annotated Corpus for Named Entity Recognition". In : **Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)**, p. 413–422.

rali

Bibliography IV

Ghaddar, Abbas et Philippe Langlais (2018a). "Robust Lexical Features for Improved Neural Network named-Entity Recognition". In : **27th International Conference on Computational Linguistics (COLING 2018)**. Santa Fe, New Mexico, USA, p. 1896–1907.

– (2018b). "Transforming Wikipedia into a Large-Scale Fine-Grained Entity Type Corpus". In : **11th edition of the Language Resources and Evaluation Conference (LREC 2018)**. Miyazaki,Japan.

Lample, Guillaume et al. (2016). "Neural Architectures for Named Entity Recognition". In : **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies**. San Diego, California : Association for Computational Linguistics, p. 260–270.

rali

Bibliography V

- Manning, Christopher D. et Hinrich Schütze (1999). **Foundations of Statistical Natural Language Processing**. MIT Press.
- Merialdo, Bernard (1994). "Tagging English text with a probabilistic model". In : **Computational Linguistics** 20(2), p. 155–172.
- Onal, Kezban Dilek et Pinar Karagoz (2015). "Named entity recognition from scratch on social media". In : **ECML-PKDD, MUSE Workshop**.
- Osborne, Miles. "Shallow Parsing as Part-of-Speech Tagging". In :
- Peters, Matthew et al. (2017). "Semi-supervised sequence tagging with bidirectional language models". In : **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)**. Vancouver, Canada : Association for Computational Linguistics, p. 1756–1765.

Bibliography VI

- Plank, Barbara, Anders Søgaard et Yoav Goldberg (2016). "Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss". In : **CoRR** abs/1604.05529.
- Ratinov, Lev et Dan Roth (2009). "Design Challenges and Misconceptions in Named Entity Recognition". In : **Proceedings of the Thirteenth Conference on Computational Natural Language Learning**. CoNLL '09. Boulder, Colorado : Association for Computational Linguistics, p. 147–155.
- Ristad, E. et R. Thomas (1997a). **Hierarchical Non-Emitting Markov models**. Rapp. tech. CS-TR-544-97. Department of Computer Science, Princeton University.

rali

Bibliography VII

- Ristad, E. S. et R. G. Thomas (1997b). "Nonuniform Markov Models". In : **Proc. ICASSP '97**. Munich, Germany, p. 791–794.
- Ritter, Alan et al. (2011). "Named Entity Recognition in Tweets : An Experimental Study". In : **EMNLP**.
- Sang, Erik F. Tjong Kim (2002). "Introduction to the CoNLL-2002 Shared Task : Language-Independent Named Entity Recognition". In : **COLING-02 : The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)**.
- Schütze, Hinrich et Yoram Singer (1994). "Part-of-Speech tagging using a variable memory Markov model". In :

rali

Bibliography VIII

Tjong Kim Sang, Erik F. et Fien De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task : Language-independent Named Entity Recognition". In : **Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4**. CONLL '03. Edmonton, Canada : Association for Computational Linguistics, p. 142–147.

Yasunaga, Michihiro, Jungo Kasai et Dragomir R. Radev (2017). "Robust Multilingual Part-of-Speech Tagging via Adversarial Training". In : **CoRR** abs/1711.04903.

rali