

UNIVERSITÉ SORBONNE NOUVELLE PARIS 3

Ecole doctorale Langage et langues : ED 268

Langues, Textes, Traitements informatiques et Cognition: Lattice (UMR8094)

THÈSE DE DOCTORAT CIFRE

SPÉCIALITÉ DU DOCTORAT : SCIENCES DU LANGAGE

---

**Adaptation au domaine et  
combinaison de modèles pour  
l'annotation de textes multi-sources  
et multi-domaines**

**Résumé substantiel**

---

*Par :*

Tian TIAN

*Sous la direction de :*

Dr. Isabelle TELLIER<sup>†</sup>

Dr. Thierry POIBEAU

Dr. Marco DINARELLI

Composition du jury :

Iris ESHKOL, Professeure, Université Paris Nanterre, rapporteure

Anne-Laure LIGOZAT, MCF, HDR, ESIEE, rapporteure

Sophie PREVOT, DR-CNRS, Université Paris 3, examinatrice

Thierry POIBEAU, DR-CNRS, Université Paris 3, directeur de thèse

Marco DINARELLI, DR-CNRS, Université Grenoble Alpes, co-directeur

Patrick MARTY, Ingénieur de recherche, FNAC, examinateur

Soutenue le

# Adaptation au domaine et combinaison de modèles pour l’annotation de textes multi-sources et multi-domaines

## Résumé

Internet propose aujourd’hui aux utilisateurs de services en ligne de commenter, d’éditer et de partager leurs points de vue sur différents sujets de discussion. Ce type de contenu est maintenant devenu la ressource principale pour les analyses d’opinions sur Internet. Néanmoins, à cause des abréviations, du bruit, des fautes d’orthographe et toutes autres sortes de problèmes, les outils de traitements automatiques des langues, y compris les reconnaissseurs d’entités nommées et les étiqueteurs automatiques morpho-syntaxiques, ont des performances plus faibles que sur les textes bien-formés (RITTER et al., 2011).

Cette thèse a pour objet la reconnaissance d’entités nommées sur les contenus générés par les utilisateurs sur Internet. Nous avons établi un corpus d’évaluation avec des textes multi-sources et multi-domaines. Ensuite, nous avons développé un modèle de champs conditionnels aléatoires, entraîné sur un corpus annoté provenant des contenus générés par les utilisateurs.

Dans le but d’améliorer les résultats de la reconnaissance d’entités nommées, nous avons d’abord développé un étiqueteur morpho-syntaxique sur les contenus générés par les utilisateurs et nous avons utilisé les étiquettes prédites comme un attribut du modèle des champs conditionnels aléatoire. Enfin, pour transformer les contenus générés par les utilisateurs en textes bien-formés, nous avons développé un modèle de normalisation lexicale basé sur des réseaux de neurones pour proposer une forme correcte pour les mots non-standard.

**Mots-clés :** adaptation au domaine, reconnaissance des entités nommées, apprentissage automatique, champs aléatoires conditionnels, réseaux de neurones

# Domain adaptation and model combination for the annotation of multi-source, multi-domain texts

## Abstract

The increasing mass of User-Generated Content (UGC) on the Internet means that people are now willing to comment, edit or share their opinions on different topics. This content is now the main resource for sentiment analysis on the Internet. Due to abbreviations, noise, spelling errors and all other problems with UGC, traditional Natural Language Processing (NLP) tools, including Named Entity Recognizers and part-of-speech (POS) taggers, perform poorly when compared to their usual results on canonical text (RITTER et al., 2011).

This thesis deals with Named Entity Recognition (NER) on some User-Generated Content (UGC). We have created an evaluation dataset including multi-domain and multi-sources texts. We then developed a Conditional Random Fields (CRFs) model trained on User-Generated Content (UGC).

In order to improve NER results in this context, we first developed a POS-tagger on UGC and used the predicted POS tags as a feature in the CRFs model. To turn UGC into canonical text, we also developed a normalization model using neural networks to propose a correct form for Non-Standard Words (NSW) in the UGC.

**Keywords :** domain adaptation, named entity recognition, machine learning, conditional random fields, neural networks

# Table des matières

<b>1</b>	<b>Extraction de propriétés de produits</b>	<b>2</b>
1.1	Introduction	2
1.2	Tâche	3
1.2.1	Spécificités et difficultés de l'extraction	3
	Problèmes de forme	3
	Mots composés	5
	Problèmes syntaxiques	5
	Problèmes sémantiques et/ou pragmatiques	6
1.2.2	Etat de l'art	7
1.3	Constitution de corpus annotés	8
1.3.1	Les trois corpus : sélection, prétraitements et annotation	8
1.3.2	Variabilité des trois corpus	9
1.4	Extraction par dictionnaires	10
1.4.1	Constitution de dictionnaires	10
1.4.2	Résultats du reconnaiseur	11
1.4.3	Analyse des erreurs	11
1.5	Extraction par CRF	13
1.5.1	Conditions de nos expériences	13
1.5.2	Résultats des expériences	15
	Baseline	15
	Recherche des meilleurs patrons	15
	Autres expériences	16
1.6	Conclusion	17
<b>2</b>	<b>Etiquetage morpho-syntaxique de tweets avec des CRF</b>	<b>19</b>
2.1	Introduction	19
2.2	Tâche et état de l'art	20
2.3	Les corpus T-POS, PTB et le jeu d'étiquettes universel	22
2.4	Les CRF, les fonctions caractéristiques et les patrons	24
2.5	Expériences	25
2.5.1	Validation croisée avec T-POS uniquement	26

2.5.2	Modèles mixtes . . . . .	26
2.6	Conclusion et perspectives . . . . .	27
<b>3</b>	<b>Détection des mots non-standards dans les tweets avec des réseaux de neurones</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Etat de l'art . . . . .	30
3.2.1	Notion de mots non-standard . . . . .	30
3.2.2	Normalisation des tweets . . . . .	31
3.3	Les corpus annotés . . . . .	32
3.4	Les modèles de réseaux de neurones convolutifs . . . . .	33
3.5	Experiences et résultats . . . . .	34
3.6	Conclusion et perspectives du travail . . . . .	37

# Introduction

Le résumé long en français qui suit en fait fondé sur trois articles publiés qui rendent compte des principaux résultats décrits dans la thèse. Ces articles sont les suivants :

- Marty, Patrick, Tian Tian, and Isabelle Tellier (2014). “Extraction de propriétés de produits”. In : CONFérence en Recherche d’Information et Applications (CORIA 2014). CORIA 2014. Nancy, France, pp. 121–136. URL : <https://hal.archives-ouvertes.fr/hal-01473389>.
- Tian Tian, Dinarelli Marco, Tellier Isabelle and Cardoso, Pedro (2015). “Etiquetage morpho-syntaxique de tweets avec des CRF”. In : TALN 2015. Caen, France. URL : <https://hal.archives-ouvertes.fr/hal-01473383>.
- Tian Tian, Dinarelli Marco, Cardoso, Pedro and Tellier, Isabelle (2017). “Détection des mots non-standards dans les tweets avec des réseaux de neurones”. In : 24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), p. 174. URL : [http://taln2017.cnrs.fr/wp-content/uploads/2017/06/actes\\_TALN\\_2017-vol2.pdf](http://taln2017.cnrs.fr/wp-content/uploads/2017/06/actes_TALN_2017-vol2.pdf)

# Chapitre 1

## Extraction de propriétés de produits

Le travail présenté dans ce chapitre vise à extraire automatiquement certaines caractéristiques de produits à partir de descriptions textuelles fournies par un site marchand. La constitution d'un corpus de référence annoté révèle certains problèmes, provenant à la fois des textes et des particularités de la tâche. Pour l'aborder, nous avons testé deux approches : une méthode d'extraction fondée sur des dictionnaires et une méthode d'apprentissage automatique avec les CRF (Champs Aléatoires Conditionnels), pour lesquels nous avons essayé un grand nombre de modèles. Les résultats de nos expériences montrent les avantages et limites de ces deux méthodes.

### 1.1 Introduction

La pertinence d'un moteur de recherche repose beaucoup sur la qualité de son indexation. Dans le cas de sites marchands, les informations importantes à indexer, *i.e.* celles qui peuvent faire l'objet de requêtes d'internautes, figurent la plupart du temps dans des textes libres décrivant les offres de produits. L'objectif du travail présenté ici est d'extraire automatiquement de descriptions d'offres en texte libre des méta-données qui caractérisent le mieux leurs propriétés.

Notre source de données est [LeGuide.com](http://LeGuide.com), un comparateur de prix. Ses clients sont des sites marchands, dont le contenu est indexé offre par offre. Chaque offre est caractérisée par plusieurs champs parmi lesquels : un identifiant, l'URL du site marchand, un titre et une description en textes libres, une image, un prix et des méta-données précisant certaines caractéristiques de l'offre ou du produit (notamment sa (ou ses) couleur(s), sa (ou ses) matière(s) et sa marque). Cependant, peu de marchands remplissent correctement les informations de leur catalogue dans les champs dédiés. La qualité

des méta-données est en particulier très variable, alors que dans la plupart des cas les titres et/ou les descriptions des offres contiennent les informations pertinentes qui devraient y figurer. Pour améliorer la pertinence du moteur de recherche et l'expérience utilisateur, en lui proposant des possibilités de requête plus variées, il paraît donc naturel de chercher à extraire ces informations depuis les textes de description.

Pour étudier la faisabilité de cette tâche, abordée récemment dans GHANI et al., 2006; PUTTHIVIDHYA et HU, 2011 nous nous sommes concentrés sur les offres disponibles en langue française sur une catégorie particulière de produits : les "chaussures femmes", qui est une de celles générant le plus de trafic. Nous précisons d'abord les difficultés identifiées lors de la constitution d'un corpus de référence annoté issu de cette catégorie de produits. Puis nous essayons d'explorer deux approches possibles pour aborder cette tâche : une à base de dictionnaires construits manuellement, une autre faisant appel à l'apprentissage automatique supervisé (en l'occurrence les CRF). Nous montrons que, suivant la nature de la propriété à extraire, l'une ou l'autre de ces méthodes est plus performante.

## 1.2 Tâche

### 1.2.1 Spécificités et difficultés de l'extraction

Les offres en français de la catégorie "chaussures femmes" indexées par [LeGuide.com](http://LeGuide.com) pour la France sont au nombre de 271125. Les méta-données que nous cherchons à extraire des descriptions en textes libres sont de trois types : la marque, la couleur et la matière. La marque est un champ à valeur unique, mais ce n'est pas nécessairement le cas des deux autres propriétés. Par exemple : " Derbies Type/Description : Chaussure à lacet Tbs pour femme. Dessus/Tige : cuir Intérieur : cuir Semelle : Elastomère "

Dans cet exemple, il y a deux occurrences de "cuir" et une d'"Elastomère" qui, toutes, correspondent au champ "matière". Ces trois valeurs sont toutes des cibles de l'extraction. Nous présentons dans ce qui suit les différentes difficultés de la tâche, telles qu'elles sont apparues lors de la constitution de corpus de référence (décrits dans la partie suivante).

#### Problèmes de forme

Les textes originaux de la base de données ne sont pas toujours bien formés. Certains sont mal segmentés, mal encodés ou incomplets.



Ainsi, dans l'exemple de la Figure 1.1, l'espace manque dans le token "tissuintérieur" qui est donc mal segmenté, alors que "tissu" devrait être repéré comme une matière. L'unité d'extraction étant le token, il ne sera pas possible de l'extraire correctement à partir du texte. D'autres textes présentent des problèmes d'apostrophes absentes, qui induisent le même genre de difficultés.

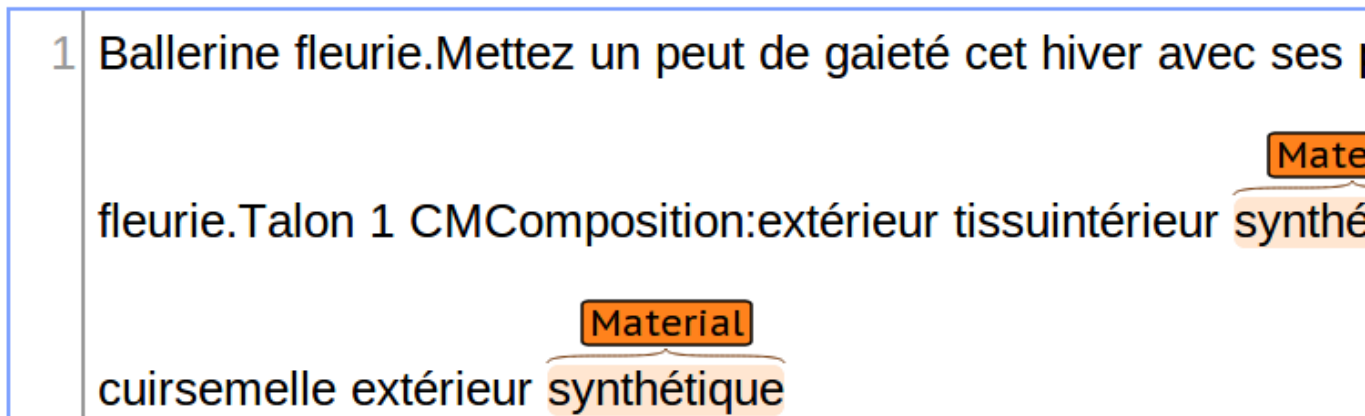


FIGURE 1.1 – Description avec mots collés

L'exemple de la Figure 1.2 est typique d'un problème de codage, dû à une mauvaise saisie de la part du marchand. Nous avons gardé ces textes comme des données dans notre base, mais sans en extraire les éventuelles valeurs qui y figurent.

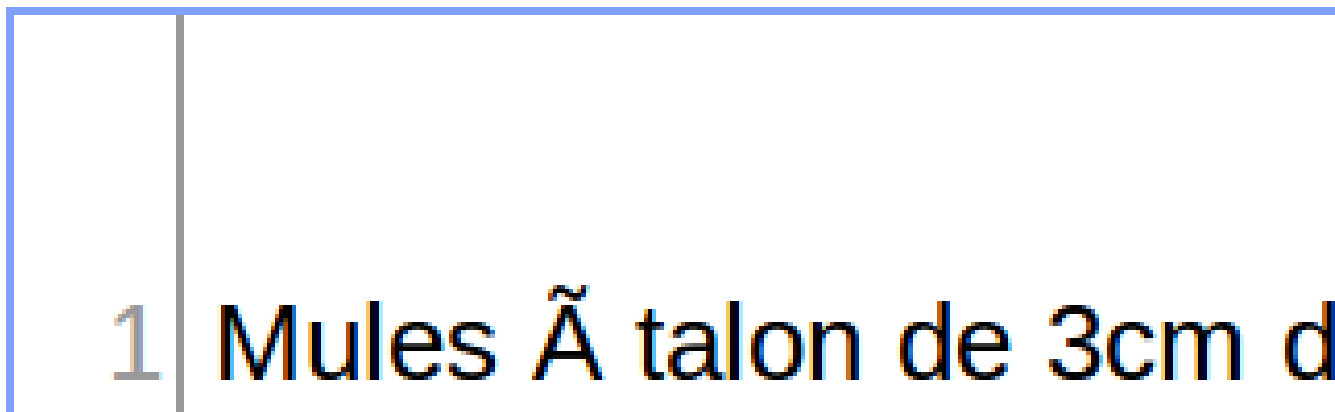


FIGURE 1.2 – Description avec problème de codage

Dans l'exemple de la Figure 1.3, un mot est écrit en abrégé. Un être humain pourrait compléter "synth" pour former "synthétique", qui est une matière. Mais nous avons fait le choix de ne chercher à extraire que les mots bien

formés, cette valeur du champ matière sera donc ignorée.

1	Tongs Mule entredoigt pour femme <span style="border: 1px solid black; padding: 2px;">Brand</span> Fitflop.
2	Sa technologie ergonomique permet d'augmenter votre activité musculaire.
3	Composition : Dessus <span style="border: 1px solid black; padding: 2px;">Material</span> cuir - Intérieur synth

FIGURE 1.3 – Description avec mot incomplet

### Mots composés

Certaines valeurs de champs sont formées de plusieurs tokens constituant un mot composé. Le tableau 1.1 montre que, suivant les cas, la sémantique du mot composé diffère fondamentalement -ou pas- de celle de sa tête : du "faux cuir" n'est pas du "cuir" mais "bleu ciel" est bien un type de "bleu". Un internaute ne s'offusquerait pas de se voir proposer un produit "bleu ciel" alors qu'il en a demandé un bleu, mais il pourrait légitimement se plaindre de voir apparaître un produit en "faux cuir" quand il en a demandé un en cuir.

A ne pas séparer	blanc cassé	faux cuir	velours côtelé
Qui peuvent être séparés	bleu ciel	rose petunia	bleu marine
Difficiles à décider	cuir vernis	daim stretch	mouton retourné

TABLE 1.1 – Exemples de mots composés

Pour les exemples de la dernière ligne du tableau, il est difficile de décider si "cuir vernis" est assimilable à "cuir" et "daim stretch" à "daim". Pour trancher cette question, nous avons essayé d'insérer des éléments (des adverbes par exemple) entre les mots composés. Ainsi "cuir très vernis"\* et "mouton très retourné" ne sont pas attestés en français mais "daim très stretch" est acceptable. Il faudrait donc extraire "cuir vernis" et "mouton retourné" comme une seule valeur, tandis que "daim" tout court sera une matière.

### Problèmes syntaxiques

Certaines descriptions d'offres ne sont pas syntaxiquement correctes. Dans l'exemple de la figure 1.4, la première phrase inclut le titre d'origine "Mules d'intérieur". D'autres textes se présentent comme de simples énumérations

de propriétés. Nous ne pourrions donc pas procéder à des analyses syntaxiques poussées de nos textes, et devons compter sur les contextes locaux pour identifier les valeurs des champs.

1	Mules d'intérieur Mule d'intérieur pour femme : motifs cousus sur la tige.
2	Hauteur du talon : 3.5 cm.
3	Composition : Dessus <span style="border: 1px solid black; padding: 2px;">Material</span> textile - Intérieur <span style="border: 1px solid black; padding: 2px;">Material</span> textile - Semel

FIGURE 1.4 – Exemple de texte syntaxiquement incorrect

### Problèmes sémantiques et/ou pragmatiques

Notre but est d'extraire la (ou les) couleur(s), la (ou les) matière(s) et la marque d'un produit dans les descriptions des offres, soit des chaussures pour notre corpus. Mais les descriptions peuvent aussi évoquer les caractéristiques d'autres objets liés au produit, comme les lacets ou les lanières : s'ils ne sont considérés que comme des accessoires, ils ne devraient pas être extraits. Il est en effet peu probable qu'ils fassent l'objet d'une requête spécifique par le biais du moteur de recherche.

Dans l'exemple suivant, les "clous métal" sont des accessoires, donc "métal" ne doit pas être repéré en tant que matière des chaussures. La "boucle" est aussi un accessoires, et "argent" ne doit pas non plus être extrait, ni comme matière ni comme couleur :

Escarpins à lanières cuir noir avec clous métal. Boucle argent avec inscription de la marque. Plateau de 2 cm et talon aiguille de 13 cm. Ces chaussures Guess sont livrées dans une boîte Guess avec une poche...

Il est parfois difficile de définir et distinguer entre les parties principales et accessoires de chaussures. Une partie sera considérée comme principale si, en son absence, le produit ne peut plus assurer la fonction à laquelle il est destiné. Ainsi, les matières des parties intérieures, extérieures et des semelles sont extraites. Pour les talons, il y a déjà matière à discussion. Une paire de "chaussures avec les talons en bois" ne sont en effet pas des "chaussures en bois". Et si les talons sont retirés, les chaussures restent portables alors que les tongs ou les sandales perdent leur fonctionnalité si on les prive

de leurs lanières ou de leurs "tiges". Ce raisonnement justifie d'extraire les parties soulignées dans les exemples suivants :

Merrell Qesbour Thong Dark Earth Man. Tongs artisanales à semelle plate conçues pour les "missions" de type repos et balade. L-assise plantaire en daim est douce et sa semelle légère. Caractéristiques :TIGE/DOUBLURE- Tige cuir- Dessus d-assise plantaire en...

Sandales Best Mountain Sandales légères à porter, Bride façon cuir à l'entredoigt, Ensemble de larges lanières en tissus dessus,

Dans ce dernier exemple, "bride façon cuir" n'est pas du "cuir", il ne faut donc pas extraire "cuir" seul. Dans certains cas extrêmes, la description du produit mentionne une couleur non associée au produit lui-même, comme dans l'exemple qui suit :

Bottes Indiennes grises cloutées Coloris : gris anthracite / doublé fourré. Composition : 100% cuir Conseil + : Des bottes très recherchées cette saison! Confortables, le cuir est doublé pour avoir bien chaud en tout saison. Le cuir est clouté sur le devant de la jambe d'un motif d'étoile en métal argent et bronze pour un effet so rock'. A porter avec un jean slim ou un leggings, un maxi gilet et un manteau de cachemire bleu ou noir pour un look working glam'!!!!

### 1.2.2 Etat de l'art

La tâche que nous venons de d'écrire, même si elle a des spécificités (à part les marques, les valeurs à extraire ne sont pas des entités nommées), relève de l'extraction d'information. C'est donc naturellement les techniques usuelles de ce domaine que nous allons utiliser pour la traiter. Nous allons en tester deux : une à base de ressources et de dictionnaires POIBEAU, 2003; EHRMANN, 2008, une autre fondée sur l'apprentissage automatique supervisé, en particulier les modèles CRF LAFFERTY, MCCALLUM et PEREIRA, 2001a; MCCALLUM et LI, 2003. C'est ce qu'ont fait par exemple RAYMOND et FAYOLLE, 2010 dans deux corpus pour l'extraction d'entités nommées. Dans leur cas, les CRF sont plus performants que les SVM, eux-mêmes plus efficaces que des transducteurs à états finis.

La reconnaissance automatique d'attributs de produits a tout de même fait l'objet de travaux spécifiques : ainsi, dans PUTTHIVIDHYA et HU, 2011 les CRF, les SVM, MaxEnt et des HMM sont mis à contribution pour l'extraction de propriétés de produits comme leur marque, leur style (décontracté, sexy, urbain, etc), leur taille et leur(s) couleur(s) dans les titres de descriptions de vêtements et chaussures pour eBay. Dans ce cas aussi, ce sont les CRF qui se comportent le mieux.

## 1.3 Constitution de corpus annotés

### 1.3.1 Les trois corpus : sélection, prétraitements et annotation

Nous avons construit trois corpus différents, tous issus de la catégorie "chaussures femmes". Il ne comprennent aucune offre avec une description vide, mais certaines se répètent. Corpus1 est construit avec des offres ne provenant que d'un seul marchand, tandis que Corpus2 complète Corpus1 avec presque la même quantité d'offres d'autres marchands. Corpus3 est créé à part, avec des offres qui n'ont jamais été sélectionnées précédemment (mais qui pourraient contenir les offres d'un marchand présent dans Corpus2). Le tableau 1.2 montre les principales propriétés de ces trois corpus et la figure 1.5 les relations qu'ils entretiennent les uns envers les autres.

Corpus	Nombre d'offres	Nombre de marchands	Nombre d'offres répétées	Nombre de tokens	Taille du vocabulaire
Corpus1	530	1	42	15493	462
Corpus2	1000	16	212	32251	1833
Corpus3	200	32	5	6750	1328

TABLE 1.2 – Corpus construits

Ces trois corpus ont été traités par le tokeniseur de NLTK (*Natural Language Toolkit* LOPER et BIRD, 2002) qui les réduit à des séquences de tokens (un token est un mot, une ponctuation, un chiffre ou un caractère spécial comme % ou &), mis en minuscule en conservant les accents. Les valeurs des trois champs à extraire ont été annotées à la main sur chacun des dits corpus. Le tableau 1.3 montre que les trois champs ne sont pas équitablement répartis dans les 1000 textes de Corpus2 (qui sera la principale source de nos

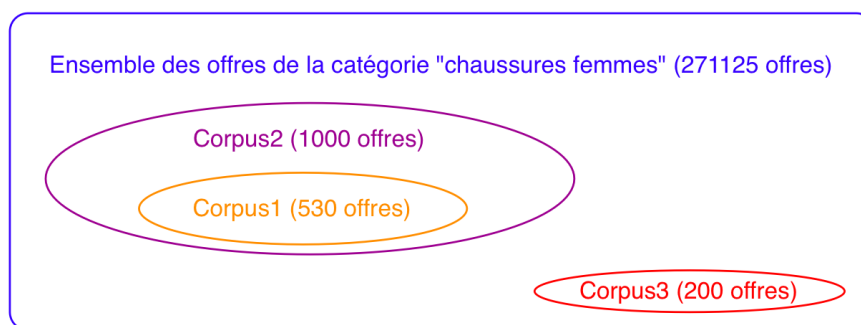


FIGURE 1.5 – Relations entre les corpus

expériences d'apprentissage), tandis que le tableau 1.4 donne le nombre d'entités de chaque type repérées dans chaque corpus, ainsi que la taille du vocabulaire concerné (nombre de tokens distincts). Le vocabulaire utilisé pour l'entité "marque" est le plus riche.

Corpus	Marque	Couleur	Matière	Aucune entité
Corpus2	552	165	827	93

TABLE 1.3 – Nombre d'offres contenant les entités

Corpus	Nombre d'entités extraites			Taille du vocabulaire		
	Marque	Couleur	Matière	Marque	Couleur	Matière
Corpus1	215	24	1121	47	15	13
Corpus2	590	757	2009	108	58	62
Corpus3	114	123	385	59	39	36

TABLE 1.4 – Nombre d'entités annotées et tailles des vocabulaires

### 1.3.2 Variabilité des trois corpus

Les trois corpus sont destinés à étudier dans quelle mesure un système d'apprentissage automatique appris sur des données plus ou moins homogènes s'applique à d'autres textes, autrement dit s'il parvient à généraliser. Corpus1 ne provenant que d'un seul marchand, c'est le plus homogène stylistiquement.

Les nombres de textes minimum, maximum et moyen issus d'un même marchand dans Corpus2 et Corpus3 figurent dans le tableau 1.5.

Les nombres minimum, maximum et moyen de tokens par offre sont donnés dans le tableau 1.6, tandis que les nombres d'occurrences minimum,

Corpus	Minimum	Maximum	Moyenne	Ecart type	Médiane
Corpus2	1	530	62,5	128,36	12,5
Corpus3	1	20	6,25	5,93	4,0

TABLE 1.5 – Nombre d'occurrence de tokens pour Corpus2 et Corpus3

maximum et moyen d'un même token dans l'ensemble d'un corpus figurent dans le tableau 1.7.

	Minimum	Maximum	Moyenne	Ecart type	Médian
Corpus1	19	38	29.2	3.76	29.5
Corpus2	2	375	32.3	38.09	28
Corpus3	2	122	33.8	18.40	31

TABLE 1.6 – Nombre de tokens par offre

	Minimum	Maximum	Moyenne	Ecart type	Médian
Corpus1	1	2204	33.5	131.86	2
Corpus2	1	2525	17.6	93.15	2
Corpus3	1	358	5.1	19.13	1

TABLE 1.7 – Nombre d'occurrences d'un même token par corpus

Chaque offre peut contenir de 0 à plusieurs entités annotées. Les nombres minimum, maximum et moyen d'entités par offre dans chacun des trois corpus sont assez similaires, comme le montre le tableau 1.8.

Corpus	Minimum	Maximum	Moyenne	Ecart type	Médiane
Corpus1	0	6	2,6	1,35	3
Corpus2	0	12	2,8	1,76	3
Corpus3	0	8	3,11	1,85	3

TABLE 1.8 – Nombre d'entités annotées par offre

## 1.4 Extraction par dictionnaires

### 1.4.1 Constitution de dictionnaires

La première méthode que nous avons testée sur notre tâche est l'application de dictionnaires (autant que de champs à extraire) construits manuellement. Pour le champ "marque", nous avons utilisé une base de marques (tous

produits confondus) disponible chez [LeGuide.com](http://LeGuide.com). Nous avons construit une liste de couleurs à partir d'informations disponibles sur Internet ainsi qu'une liste de matières. Le tableau 1.9 donne la taille de chaque liste.

Nom du dictionnaire	Taille du dictionnaire
marque	15391
couleur	596
matière	67

TABLE 1.9 – Tailles des vocabulaires pour chaque champ à extraire

## 1.4.2 Résultats du reconnaiseur

Pour chercher toutes les occurrences des entrées d'un dictionnaire dans un texte donné, nous utilisons l'algorithme de recherche de motif d'Aho-Corasick CROCHEMORE, HANCART et LECROQ, 2007, en raison de son efficacité (complexité linéaire en fonction de la taille du texte en entrée). Quelques pré-traitements supplémentaires ont été appliqués : les accents sont supprimés (contrainte d'implémentation) et un caractère espace est ajouté avant et après chaque token (afin d'indiquer le début et la fin du token). Le reconnaiseur est configuré pour extraire la séquence de tokens la plus longue correspondant à une valeur d'une liste (qui inclut des mots composés). Les résultats obtenus avec cette méthode pour chaque champ et chaque corpus sont fournis dans le tableau 1.10. Notons que, pour chaque offre, nous comparons les résultats du reconnaiseur avec l'intégralité des données annotées (un même champ peut donc prendre plusieurs valeurs) et que l'égalité stricte entre les valeurs des champs est requise. L'évaluation se fait en prenant en compte la localisation de l'entité, sa valeur et son type.

## 1.4.3 Analyse des erreurs

Les entités "couleur" et "matière" apparaissent clairement comme les plus faciles à extraire. La précision et le rappel du reconnaiseur sur le champ "marque" sont en revanche plus faibles. Cela est notamment dû à la non-exhaustivité du dictionnaire de marques qui, bien que volumineux, n'est pas spécialisé pour la catégorie chaussures. La faiblesse en précision s'explique aussi par le fait que cette liste de marques, une fois mise en minuscules et sans accents, contient beaucoup de mots de la langue courante, entraînant du bruit dans l'extraction. La liste de marques suivante en donne des exemples :



Champ	Corpus	Précision	Rappel	F-Mesure
marque	Corpus1	18,0%	66,5%	28,3%
	Corpus2	26%	74,1%	38,5%
	Corpus3	28,2%	81,6%	41,9%
couleur	Corpus1	100%	83,3%	90,9%
	Corpus2	82,9%	91,7%	87,1%
	Corpus3	89,1%	86,2%	87,6%
matière	Corpus1	98,9%	98,0%	98,4%
	Corpus2	95,6%	89,1%	92,2%
	Corpus3	90,7%	81,3%	85,7%

TABLE 1.10 – Evaluation de la méthode à base de dictionnaires par entité

- elle : un magazine
- tous : une marque de bijoux
- plus : se trouve dans la base de marque
- boots : une marque de chaussures
- talon : une marque de chaussures
- look : se trouve dans la base de marque
- chic : se trouve dans la base de marque

Le même phénomène peut concerner aussi, mais à moindre échelle, les autres champs : la couleur "maïs", transformée en "mais" par le retrait des accents, a ainsi entraîné beaucoup d'extractions fautives.

Le deuxième phénomène source d'erreurs que nous avons repéré est celui de l'ambiguïté de certains tokens, qui appartiennent à plusieurs listes. C'est le cas des exemples suivants :

- orange : marque d'un opérateur téléphonique ou couleur
- ciel : marque ou couleur
- chrome : couleur ou matière
- bronze : couleur ou matière
- mousse : couleur ou matière
- paille : couleur ou matière

Enfin, la non-prise en compte des contextes entraîne des extractions abusives comme celle de "cuir" dans des expressions comme "faux cuir" ou "façon cuir". Un vrai système à base de règles devrait bien sûr prendre en compte ces phénomènes. Mais, plutôt que d'approfondir cette approche, nous avons préféré la mettre en concurrence avec une méthode d'apprentissage automatique.

## 1.5 Extraction par CRF

L'alternative aux méthodes à base de règles écrites manuellement (dont nos dictionnaires sont une version minimale) est bien entendu l'utilisation de techniques d'apprentissage automatique. La plus efficace à l'heure actuelle pour la tâche d'extraction d'entités est celle mettant en œuvre les CRF : nous en explorons dans cette section toutes les possibilités.

### 1.5.1 Conditions de nos expériences

Les CRF sont des modèles graphiques probabilistes non dirigés et discriminants pour la prédiction d'étiquettes pour des données structurées LAFERTY, MCCALLUM et PEREIRA, 2001a ; TELLIER et TOMMASI, 2011. Pour la tâche de détection d'entités dans des données textuelles, les types de CRF employés sont des CRF linéaires SHA et PEREIRA, 2003. Ces derniers prédisent une séquence d'étiquettes à partir de la séquence textuelle segmentée (en tokens) en entrée.

Les prédictions d'un CRF sont basées sur la combinaison de fonctions caractéristiques qui modélisent des propriétés plus ou moins locales de la séquence de tokens observée. On distingue deux types de fonctions caractéristiques : les fonctions unigrammes qui prennent en entrée l'observation et l'étiquette du token courant, et les fonctions bigrammes qui prennent en entrée l'observation, l'étiquette du token courant et celle du token précédent. Un des intérêts des CRF linéaires est de pouvoir modéliser des dépendances longues car les fonctions caractéristiques peuvent accéder à toute la séquence d'observation. Cependant en pratique, les fonctions caractéristiques sont souvent utilisées pour modéliser des propriétés locales du texte. Ces fonctions sont générées à partir de patrons, qui définissent une conjonction de tests basiques sur les attributs de la séquence de tokens, et d'une fenêtre d'observation des données, qui contrôle la localité des fonctions caractéristiques.

L'apprentissage d'un CRF linéaire se fait de manière supervisée à partir d'un ensemble de couples (séquence de tokens , séquence d'étiquettes) et consiste à déterminer l'ensemble des poids des fonctions caractéristiques, généralement par maximum de vraisemblance et descente de gradient. Le calcul de la séquence d'étiquettes la plus probable pour une séquence de tokens se fait à l'aide de l'algorithme de Viterbi.

Pour nos expériences avec les CRF, nous avons utilisé le logiciel Wapiti LAVERGNE, CAPPÉ et YVON, 2010. Le codage classique d'étiquettes BIO pour

"Begin/In/Out" SARAWAGI, 2008 est associé aux noms des différents champs à trouver (marque, couleur et matière) pour annoter les textes, ce qui fait un total de 7 étiquettes distinctes (B et I associés à chacun des trois champs, plus O). Nous avons en outre essayé d'exploiter différents attributs, soit internes (propriétés des textes lisibles dans les données) soit issus de ressources linguistiques externes (listes, étiqueteur morpho-syntaxique). La liste de ces attributs est la suivante :

- la valeur du token en minuscule et/ou sans accent;
- la présence de majuscule(s) en début, au milieu ou partout dans le token (attributs booléens);
- le type de token (uniquement des lettres, présence de chiffres, de ponctuation ou de caractères spéciaux);
- la longueur en nombre de caractères du token;
- la position dans le texte du token : chaque texte est découpé en 6 blocs en fonction du nombre total de tokens, la position d'un token est un nombre entre 0 et 5 correspondant à l'index du bloc où il se trouve;
- la présence du token (en tant que B ou I) dans un de nos dictionnaires d'entités (celui de marque, de matière ou de couleur );
- la racine du token (obtenue avec NLTK LOPER et BIRD, 2002);
- l'étiquette morpho-syntaxique prédite par SEM, un analyseur morpho-syntaxique à base de CRF (TELLIER et DUPONT, 2013).

Cette liste d'attributs, combinée au choix de la fenêtre d'observation sur les données, donne une combinatoire très vaste de patrons possibles pour définir les fonctions caractéristiques du CRF. Nous avons exploré cet espace de façon progressive et systématique. Les Corpus 1 à 3 seront par ailleurs mis à contribution pour évaluer la capacité de généralisation des modèles sur des données plus ou moins homogènes. Nos expériences visent bien sûr à obtenir des résultats si possible au moins aussi bons que ceux atteints par les dictionnaires, mais elles poursuivent également différents autres objectifs :

- trouver les meilleurs attributs et les meilleurs patrons pour l'apprentissage d'un CRF sur cette tâche;
- déterminer s'il vaut mieux apprendre un modèle distinct pour chaque champ ou si un unique modèle est capable d'extraire efficacement tous les champs en même temps;
- déterminer s'il vaut mieux utiliser un corpus d'apprentissage très spécialisé (constitué par exemple d'un seul marchand, comme Corpus1) ou plus varié.

## 1.5.2 Résultats des expériences

### Baseline

A titre de baseline, nous avons défini un patron bigramme "minimal" ne faisant appel qu'à la valeur du token courant (sans exploiter d'autres attributs). Un seul modèle est appris pour reconnaître simultanément les trois types de champs. Ses résultats sur Corpus2 en validation croisée à 5 plis sont donnés dans le tableau 1.11.

Corpus	Champ	Précision	Rappel	F-mesure
Corpus2	Marque	86,18%	80,84%	83,41%
	Couleur	76,27%	63,71%	69,22%
	Matière	85,13%	85,02%	85,07%

TABLE 1.11 – Résultat de la baseline sur Corpus2

Nous constatons d'ores et déjà que les marques sont nettement mieux traitées par cette approche, mais les performances sur les autres champs sont en revanche inférieures à celles obtenues par les dictionnaires. Les différences de performance entre champs peuvent s'expliquer par leurs taux de présence dans les corpus : d'après le tableau 1.3, "couleur" est en effet moins présent que "marque", lui-même moins fréquent que "matière", dans les données de Corpus2.

### Recherche des meilleurs patrons

Nous cherchons maintenant à améliorer ces résultats en exploitant des propriétés plus fines dans les patrons définissant les fonctions caractéristiques des CRF. Les choix à faire sont la nature de l'attribut testé (parmi les 10 listés précédemment) et la taille de la fenêtre (nombre impair compris entre 1 et 33) prises en compte sur les données d'observation. Pour trouver la meilleure fenêtre adaptée à chaque attribut, nous avons tout d'abord testé indépendamment les 170 patrons bigrammes possibles, utilisés seuls, en validation croisée à 5 plis avec Corpus2, pour chacun des trois champs à extraire. Cette première série d'expériences sert à sélectionner, pour chaque attribut, le patron qui mène à la meilleure précision, et par ailleurs tous ceux qui permettent d'atteindre un meilleur rappel que celui-ci. Quand plus de deux patrons différents pour un même attribut étaient conservés, nous avons procédé à une sélection supplémentaire consistant à ne garder que les meilleurs en F-mesure. Une fois  $K$  patrons portant chacun sur une seule propriété choisis,

nous devons chercher la meilleure façon de les utiliser conjointement pour produire les fonctions caractéristiques du CRF. Nous avons le choix de garder entre 1 et  $K$  d'entre eux conjointement, ce qui fait une combinatoire de  $C_K^2 + C_K^3 + \dots + C_K^K$  ensemble de patrons possibles. Pour le champ "marque" nous avons ainsi testé 120 combinaisons de patrons possibles, 502 pour les couleurs et 2036 pour les matières.

Le tableau 1.12 montre les résultats des deux meilleures combinaisons de patrons trouvées (la meilleure en précision et la meilleure en F-mesure) pour chaque champ.

Champ	Modèle	Précision	Rappel	F-mesure
Marque (120 modèles testés)	Baseline	86,18%	80,84%	83,41
	Meilleur en précision	<b>88,9%</b>	81,6%	85,06%
	Meilleur en F-mesure	88,83%	83,43%	<b>86,03%</b>
Couleur (502 modèles testés)	Baseline	76,27%	63,71%	69,22%
	Meilleur en précision	<b>86,71%</b>	71,02%	77,91%
	Meilleur en F-mesure	85,59%	74,43%	<b>79,47%</b>
Matière (2036 modèles testés)	Baseline	85,13%	85,02%	85,07%
	Meilleur en précision	<b>86,51%</b>	84,67%	85,57%
	Meilleur en F-mesure	85,51%	86,31%	<b>85,9%</b>

TABLE 1.12 – Meilleures combinaisons de patrons

Nous constatons que les meilleurs modèles pour "couleur" augmentent significativement les scores de la baseline, tandis que l'extraction des champs "marque" et "matière" progresse plus difficilement de 1% à 3%. Les meilleurs patrons sélectionnés pour les trois champs contiennent des propriétés différentes, mais leur point commun est une taille de fenêtre presque toujours comprise entre 3 et 7. Nous avons aussi procédé à des expériences similaires pour les patrons unigrammes (que nous ne détaillons pas ici) : ils donnent des résultats souvent très proches tout en prenant moins de temps de calcul.

### Autres expériences

Comme nous l'avons vu en section 1.3.1, Corpus1 contient des offres en provenance d'un seul et unique marchand : est-ce avantageux de créer un modèle par marchand ou une certaine diversité est-elle préférable? Pour trancher cette question, nous avons fait des expériences où seul Corpus1 était utilisé, en validation croisée. Les résultats obtenus étaient moins bons que ceux de la baseline sur Corpus1, il ne semble donc pas avantageux de sur-spécialiser les modèles CRF marchand par marchand.

Nous avons également testé les meilleures combinaisons de patrons précédemment affinées sur Corpus2 sur Corpus3, pour vérifier que nos combinaisons ne sont pas trop spécifiques de l'ensemble d'apprentissage, et nous avons comparé systématiquement sur ce corpus les modèles spécialisés sur un seul champ et ceux cherchant à extraire tous les champs en même temps. Les meilleurs résultats obtenus pour la F-mesure figurent dans le tableau 1.13.

	<b>Précision</b>	<b>Rappel</b>	<b>F-mesure</b>
Meilleurs résultats sur marque	91,43%	26,45%	41,03%
Meilleurs résultats sur couleur	84,21%	65,04%	73,39%
Meilleurs résultats sur matière	86,32%	62,60%	72,57%

TABLE 1.13 – Meilleurs résultats obtenus sur Corpus3

Ces meilleurs résultats ont tous été obtenus à l'aide de modèles (pas les mêmes sur chaque ligne) qui extraient simultanément tous les champs. Si leurs précisions sont très proches de celles obtenues en validation croisée sur Corpus2 (et meilleures que celle de la baseline), les scores de rappel sont en revanche dégradés. Cela correspond sans doute à un effet de sur-apprentissage dû à la procédure de choix des patrons : les modèles n'ont pas réussi à bien généraliser sur un corpus nouveau.

## 1.6 Conclusion

Dans ce chapitre, nous nous sommes confrontés à une tâche d'extraction de caractéristiques de produits dans des textes de sites marchands. Cette tâche, potentiellement très utile, relève de l'extraction d'information, mais elle présente des caractéristiques spécifiques : la multiplicité des données à extraire est variable d'un texte à un autre, et toutes ne sont pas des entités nommées. Les textes sont écrits avec des styles très variés, pouvant aller de la simple énumération de propriétés à des descriptions riches et élaborées, évoquant même d'autres produits dont les propriétés ne devraient pas être prises en compte dans l'extraction.

Nous avons abordé cette tâche avec les deux familles d'outils traditionnels utilisés en extraction d'information : les règles écrites manuellement (réduites ici à des dictionnaires) et l'apprentissage automatique avec des CRF.

Les dictionnaires apparaissent particulièrement performants pour l'extraction de propriétés dont les valeurs peuvent être facilement listées (les couleurs et les matières), même si le traitement de tokens ambigus requiert certainement la prise en compte de contextes plus riches que ceux que nous avons utilisés, ainsi que des prétraitements pour éliminer les confusions possibles entre les champs (*i.e.* pas d'intersection entre les différents dictionnaires) ou avec des mots courants. Pour la reconnaissance des marques, en revanche, qui sont des entités nommées, la solution de l'apprentissage automatique semble la plus prometteuse. Les marques sont en effet les propriétés les plus évolutives et les plus spécifiques du type de produit décrit. Notre dictionnaire de marques qui, malgré sa taille, n'est ni spécialisé ni exhaustif, a favorisé le rappel mais s'est avéré très mauvais en précision. C'est au contraire sur ce champ que nos modèles CRF ont fourni leurs meilleurs résultats. Notons d'ailleurs que les dictionnaires utilisés comme règles étaient intégrés aux attributs des CRF appris, c'est pourtant seulement dans le cas des marques que les modèles en question ont dépassé la capacité d'extraction de ces dictionnaires. Les CRF n'ont pas réussi à tirer parti de cet attribut en conjonction avec les autres, comme on aurait pu l'espérer.

Cette première exploration suggère à l'avenir une combinaison de méthodes : pour certains champs "faciles à circonscrire" et relativement indépendants des produits décrits, les listes et les règles peuvent s'avérer suffisantes. Mais l'apprentissage automatique devient nécessaire pour ceux qui présentent une plus grande variabilité ou une plus grande spécificité. Il reste évidemment à valider ces intuitions sur des données nouvelles.

Ces travaux mettent aussi en évidence une difficulté d'utilisation des CRF : quelle stratégie employer pour sélectionner les caractéristiques produisant le meilleur modèle? Peu de travaux abordent ce point, l'approche couramment utilisée étant de déléguer cette sélection à l'algorithme d'apprentissage des CRF, qui va assigner un poids nul aux fonctions caractéristiques non pertinentes MCCALLUM et LI, 2003. Or nos expériences ont montré que cette approche n'est pas suffisante et qu'une exploration plus systématique des combinaisons possibles des fonctions caractéristiques permet d'obtenir de meilleurs résultats. Encore une fois, cette constatation empirique se doit d'être validée sur d'autres données et analysée sur le plan théorique.

## Chapitre 2

# Étiquetage morpho-syntaxique de tweets avec des CRF

Nous nous intéressons dans ce chapitre à l'apprentissage automatique d'un étiqueteur morpho-syntaxique pour les tweets en anglais. Nous proposons tout d'abord un jeu d'étiquettes réduit avec 17 étiquettes différentes, qui permet d'obtenir de meilleures performances en exactitude par rapport au jeu d'étiquettes traditionnel qui contient 45 étiquettes. Comme nous disposons de peu de tweets étiquetés, nous essayons ensuite de compenser ce handicap en ajoutant dans l'ensemble d'apprentissage des données issues de textes bien formés. Les modèles mixtes obtenus permettent d'améliorer les résultats par rapport aux modèles appris avec un seul corpus, qu'il soit issu de Twitter ou de textes journalistiques.

### 2.1 Introduction

Les réseaux sociaux sont devenus la principale source de textes générés par des utilisateurs sur Internet. Ces textes constituent des données massives potentiellement porteuses de beaucoup d'information, mais aussi difficiles à traiter automatiquement du fait de leur grande variété en genres (textes de blogs, forums, tweets...), domaines et styles. Après la phase préliminaire de tokenisation, l'étiquetage morpho-syntaxique de ces textes apparaît comme une étape fondamentale de traitement, permettant d'éventuelles analyses syntaxiques ultérieures.

Dans ce chapitre, nous nous intéressons à l'étiquetage morpho-syntaxique des tweets, qui présentent souvent le plus grand écart à la norme. Nous évoquons dans un premier temps la spécificité de ces données ainsi que les difficultés majeures qui en découlent pour la tâche d'étiquetage morpho-syntaxique. Puis nous introduisons les corpus (en anglais) utilisés dans nos




expériences : l'un d'eux est constitué de tweets, l'autre de textes journalistiques plus respectueux de la norme linguistique standard. Pour les traiter, nous proposons d'abord un jeu d'étiquettes morpho-syntaxiques réduit par rapport à ceux utilisés dans les corpus annotés habituellement disponibles. Nous décrivons ensuite l'approche que nous avons adoptée pour apprendre un étiqueteur morpho-syntaxique de tweets anglais avec des CRF. Nous essayons en particulier d'apprendre des modèles à partir de mélanges de textes issus des deux corpus. Les paramètres que nous faisons varier dans nos expériences sont donc : les propriétés prises en compte dans les textes et les patrons qui définissent les fonctions caractéristiques des CRF, les paramètres de régularisation de l'implémentation et les proportions des différents textes sources dans l'ensemble d'apprentissage. Les résultats de nos expériences montrent que notre jeu d'étiquettes permet d'améliorer les performances de l'étiqueteur et qu'un modèle appris avec un mélange de tweets et de textes de journaux donne de meilleurs résultats que ceux appris sur un seul type de textes.

## 2.2 Tâche et état de l'art

Le cadre général de ce travail est celui de l'analyse d'opinion portant sur des noms de produits ou de marques cités dans des tweets. L'étiquetage morpho-syntaxique est un préalable nécessaire car nous souhaitons identifier les produits/marques évoqués (présents sous la forme de noms communs ou de noms propres) ainsi que les mots éventuellement porteurs de sentiments (principalement les adjectifs, les verbes et les adverbes). Le but de notre étiqueteur morpho-syntaxique est donc de différencier les grandes classes de catégories grammaticales, pas de vérifier des propriétés morpho-syntaxiques fines comme les accords en genre et en nombre ni de préparer une analyse syntaxico-sémantique profonde.

Malgré cette simplification, l'étiquetage automatique des tweets reste difficile, surtout à cause de leur caractère mal formé, qu'illustre par exemple la figure 2.1.



```
Today wasz Fun cuzz anna Came juss for me <3( : hahaha
```

FIGURE 2.1 – Exemple 1 de tweet

Le tweet de la figure 2.1 est issu du corpus RITTER et al., 2011. La phrase “correcte” devrait être :

Today was fun because Anna came just for me <3( : hahaha

Dans cet exemple, les difficultés sont multiples :

- présence de fautes d'orthographe : wasz (was), cuz (because), juss (just)
- inversion majuscule/minuscule : Fun (fun), anna (Anna), Came (came)
- émoticon : <3( :
- interjection : hahaha

Dans un dictionnaire anglais, les mots comme "wasz", "cuz", "juss", "<3(:" et "hahaha" n'existent probablement pas. Les étiqueteurs à base de règles, fondées sur des listes de mots associés à leurs catégories (comme was : verbe) ne fonctionneront donc pas avec ce genre de tweets, à moins de mises à jour massives des ressources qu'elles exploitent, ou de pré-traitements permettant de "corriger" les textes initiaux. Plutôt que de chercher à réaliser ce genre de prétraitements, nous choisissons d'étiqueter ce type de textes comme s'ils étaient "bien écrits". Ceci revient à considérer que "wasz" est une variante possible du mot "was", etc. Pour cela, nous allons compter sur les capacités de généralisation de l'apprentissage automatique.

Eduardo Surita : your a freaking ... <http://tumblr.com/xmciuda0t>

FIGURE 2.2 – Exemple 2 de tweet

La figure 2.2 illustre une autre difficulté de l'étiquetage morpho-syntaxique des tweets. En anglais standard, il faudrait écrire "you're" au lieu de "your". Le mot "your" de ce tweet est ainsi porteur de deux catégories morpho-syntaxiques : pronom et verbe. Pour remédier à ce genre de problèmes, plusieurs solutions sont possibles :

- ajouter un pré-traitement de normalisation afin de substituer "you're" à "your" et traiter ensuite le tweet comme du texte écrit standard ;
- annoter le texte tel qu'il est, avec une seule étiquette syntaxique pour "your". Dans ce cas, on peut soit créer une étiquette nouvelle spéciale (pronom+verbe) comme dans GIMPEL et al., 2011, soit choisir une étiquette "traditionnelle" (par exemple "pronom") comme dans RITTER et al., 2011. Cette dernière solution entraînera la possibilité de séquences d'étiquettes en principe interdites pour certaines phrases, comme "pronom déterminant adjectif", signe d'une construction sans verbe.

Ces deux exemples expliquent que les performances des étiqueteurs appris sur des corpus "bien formés" comme le Penn TreeBank MARCUS, SANTORINI et MARCINKIEWICZ, 1993 (aussi appelé PTB par la suite) ou le French TreeBank ABEILLÉ, CLÉMENT et TOUSSENEL, 2003 chutent quand ils sont confrontés à des tweets. Le Maximum Entropy POS Tagger<sup>1</sup> appris avec le Penn TreeBank dans TOUTANOVA et MANNING, 2000 a obtenu 96.86% d'exactitude en validation croisée mais seulement 81.3% sur les tweets RITTER et al., 2011. Les expériences menées sur le français montrent des résultats similaires : l'étiqueteur appris sur le French TreeBank dans CONSTANT et al., 2011 atteint 97.3% d'exactitude en validation croisée, alors que celui utilisé dans FARHAD, CAROLINE et CLAUDE, 2014 a eu seulement un score de 91.7% sur le corpus French Social Media SEDDAH et al., 2012.

Beaucoup de travaux ont été consacrés à l'amélioration du traitement des tweets. FOSTER et al., 2011, par exemple, essaient d'apprendre un analyseur syntaxique qui leur est dédié. GIMPEL et al., 2011 propose d'utiliser un tokeniseur spécifique différent et un jeu d'étiquettes particulier (par exemple nom+verbe pour les tokens comme "I'll"). RITTER et al., 2011 cherchent aussi à construire un étiqueteur morpho-syntaxique avec un modèle appris sur un mélange de plusieurs corpus : le Penn TreeBank (MARCUS, SANTORINI et MARCINKIEWICZ, 1993), le NPS IRC Corpus (un corpus de *chatroom* introduit dans FORSYTH, 2007) et son propre corpus twitter (T-POS). Notre travail se situe dans la même lignée, consistant à mélanger des données issues de corpus de textes bien formés et mal formés pour l'apprentissage. Nous créons ainsi un modèle mixte que nous testons sur les données de Twitter. Notre contribution dans ce chapitre porte sur une proposition de jeu d'étiquettes réduit et sur l'étude des effets de diverses proportions de corpus "standard" et de corpus cible (Twitter) pour apprendre un modèle, associée à une optimisation des paramètres de régularisation du modèle d'apprentissage.

## 2.3 Les corpus T-POS, PTB et le jeu d'étiquettes universel

RITTER et al., 2011 ont mis à disposition un corpus Twitter annoté en étiquettes morpho-syntaxiques et en entités nommées. Les sujets de discussions y sont très variés : vie quodidienne, équipes sportives, films, groupes musicaux, etc. Nous avons choisi ce corpus comme référence pour Twitter. La

---

1. Stanford Pos Tagger : <http://nlp.stanford.edu/software/tagger.shtml>

partie annotée est toutefois très limitée : elle ne contient que 787 tweets, soit 15972 tokens. La taille de ce corpus ne permet donc pas d'apprendre un modèle complet.

Ce corpus Twitter contient de nombreux exemples d'une autre spécificité des tweets : la présence de caractères spéciaux désignant des "hashtags" (qui servent à l'indexation des tweets par mots clés), des "retweets" (pour une rediffusion de tweets), des URL et des "usernames" (comptes d'utilisateurs de tweets), aux catégories morpho-syntaxiques souvent ambiguës. Prenons l'exemple des hashtags, repérables à leur symbole "#": ils peuvent se substituer à un mot simple (un adjectif dans l'exemple 3), à un constituant syntaxique complet (exemple 4) ou être simplement un terme d'indexation sans rôle syntaxique précis (exemple 5).

3	My #twitter age is 458 days 0 hours 3 minutes 49 seconds
4	On Thanksgiving after you done eating its #TimeToGetOut unless you wanna help with the dishes
5	New book blogger @GennaSarnak launches weekly feature , Poetry Sunday : <a href="http://tinyurl.com/47vbdy5">http://tinyurl.com/47vbdy5</a> #Books #Poetry

TABLE 2.1 – Les hashtags dans les tweets

RITTER et al., 2011 ont choisi de ne pas traiter les hashtags et autres mots spéciaux utilisés dans Twitter comme des composants linguistiques comme les autres. Ils ont ajouté dans leur corpus de tweets quatre étiquettes spécifiques : *USR* pour "*at mention*", *HT* pour "*hashtag*", *URL* et *RT* pour "*retweet*" en plus des 45 étiquettes définies dans le Penn TreeBank. Ils n'ont donc pas pris en compte les éventuels rôles syntaxiques des hashtags, illustrés dans les exemples 3 et 4. Ce genre de tweets nécessiterait un étiqueteur morpho-syntaxique particulièrement flexible et robuste.

Notre étiqueteur morpho-syntaxique est construit dans un contexte d'analyse multi-langue. Un seul jeu d'étiquettes pour toutes les langues est dans ce cas nécessaire. Comme, de plus, l'objectif final de notre travail relève de la fouille d'opinion dans des données massives et devrait se passer d'une analyse syntaxique complète de ces données, nous n'avons pas besoin de certaines des distinctions du PennTreebank, comme verbes au passé / verbes au présent, nom commun singulier / pluriel, etc. Cela nous a amené à nous intéresser au jeu d'étiquettes proposé dans PETROV, DAS et MCDONALD, 2012, qui se veut universel tout en laissant la possibilité de réaliser une analyse syntaxique rudimentaire. Il comporte 12 étiquettes différentes et des correspondances (*mapping*) avec les jeux d'étiquettes utilisés dans 25 TreeBanks de 25 langues différentes sont disponibles. Il a été testé sur le PTB et permet d'obtenir des résultats légèrement meilleurs en exactitude (96.8% contre

96.7% avec les étiquettes originales). Nous l’avons étendu en lui adjoignant les quatre étiquettes dédiées à Twitter utilisées dans RITTER et al., 2011. Enfin, nous avons rétabli la distinction entre les noms propres et les noms communs (qui sont assimilés dans PETROV, DAS et MCDONALD, 2012), afin de préparer le terrain à la tâche ultérieure d’extraction des entités nommées. Le nombre total d’étiquettes que nous considérons est donc finalement de 17 : NUM (nombres), PUNCT, NN (nom commun), NP (nom propre), VB (verbe), ADJ (adjectif), ADV (adverb), DET (déterminant), PRON (pronom), CC (conjonction de coordination), PREPCS (préposition et conjonction de subordination), PRT (particule), X (mot inconnu, interjection, émotion), RT, URL, USR, et HT.

Le tableau 2.2 montre les correspondances entre ce jeu et les étiquettes du PTB, ainsi qu’avec celles de T-POS. Les différences d’étiquettes entre ces deux corpus sont marquées en gras.

Tag universel	Tag Penn TreeBank	Tag T-POS	Remarques
NUM	CD	CD	nombres
PUNCT	" , -LRB- -RRB- . : - "	" ( ) . , : NONE O LS	
NN	NN NNS	NN NNS	noms communs
NP	NNP NNPS	NNP NNPS	noms propres
VB	MD VB VBD VBG VBN VBP VBZ	MD VB VBD VBG VBN VBP VBZ <b>VPP</b>	verbes
ADJ	<b>AFX</b> JJ JJR JJS	JJ JJR JJS	AFX : Yes
ADV	RB RBR RBS WRB	RB RBR RBS WRB	WRB : <i>where, when</i>
DET	DET PDT <b>PRP\$</b>	PDT DT WDT <b>EX TD</b>	PDT : <i>half</i> , PRP\$ : <i>his</i>
PRON	PRP WP	PRP PRP\$ WP WP\$	pronoms
CC	CC	CC	
PREPCS	IN	IN	préposition et CS
PRT	POS TO RP	POS TO RP	<i>particule</i>
X	<b># \$ FW NIL SYM INTJ</b>	INTJ SYM FW	FW : mot inconnu
RT		RT	<i>Retweet</i>
HT		HT	<i>Hashtag</i>
URL		URL	Adresse d’un site web
USR		USR	Compte d’utilisateur

TABLE 2.2 – Correspondance entre les étiquettes du PTB et celles de Ritter

## 2.4 Les CRF, les fonctions caractéristiques et les patrons

Les CRF (Conditional Random Fields), introduits dans LAFFERTY, MCCALLUM et PEREIRA, 2001b, font désormais partie des méthodes d’apprentissage automatique supervisé standard, ils sont particulièrement efficaces pour l’annotation de séquences. Différents choix sont possibles pour définir les “patrons” qui leur seront utiles pour la tâche d’annotation morpho-syntaxique. LAVERGNE, CAPPÉ et YVON, 2010 et SUZUKI et ISOZAKI, 2008 ont ainsi chacun proposé un ensemble de patrons destinés à l’étiquetage

morpho-syntaxique de l’anglais, tandis que FARHAD, CAROLINE et CLAUDE, 2014 et CONSTANT et al., 2011 ont construit des modèles pour le français. Nos patrons sont inspirés de ceux de CONSTANT et al., 2011, définis pour des textes écrits. Etant donnée l’irrégularité des tweets, nous avons utilisé seulement des unigrammes d’étiquettes associés aux propriétés du tableau 2.3, ainsi que les bigrammes d’étiquettes seules, pour prendre en compte leurs transitions (comme dans un modèle de type *HMM*). Dans ce tableau, le chiffre 0 désigne le token courant, -1 le précédent, 1 le suivant, etc. Toutes nos expériences ont été menées avec le logiciel Wapiti<sup>2</sup>.

Type	Nom de propriété	Fenêtre
valeur de token	valeur de token	[-2, -1, 0, 1, 2]
valeur de token	valeur de token bigramme	[-1, 1], [-1, 0], [0, 1]
type de token en binaire	fstUpper, allUpper, hasDash, hasNumb	0
lowercase	lower	0
prefixe/suffixe	prefixe_n, suffixe_n (n = 1..5)	0
ressource externe en binaire	catétories dans PTB	[-2, -1, 0, 1, 2]

TABLE 2.3 – Patron des CRF pour les expériences

## 2.5 Expériences

Notre but est tout d’abord de mesurer l’effet sur l’apprentissage de passer des jeux d’étiquettes initiaux du T-POS et du PTB (45 étiquettes) à notre jeu d’étiquettes universel spécifique des tweets (17 étiquettes). Nous nous attendons à de meilleures performances en annotation morpho-syntaxique avec moins d’étiquettes. Ensuite, nous essayons d’évaluer les performances d’un modèle mixte appris en mélangeant, dans diverses proportions, des données du PTB et de T-POS.

Pour ce faire, nous voulions construire notre propre baseline en apprenant un modèle sur le Penn TreeBank entier et en le testant sur le corpus T-POS. Notre serveur ne s’est cependant pas avéré suffisamment puissant pour mener à bout ces expériences. Nous nous contentons donc de reproduire ici le résultat présenté dans RITTER et al., 2011, obtenu avec un modèle de Maximum d’Entropie : l’exactitude annoncée est de 81.3%.

Nous avons également procédé à une validation croisée en 10 blocs sur le corpus T-POS seul. L’évaluation en exactitude est dans ce cas une moyenne des 10 tests sur 1/10 de T-POS chacun.

2. wapiti 1.5.0, <https://wapiti.limsi.fr>

Enfin, notre dernière série d'expériences teste des modèles appris avec un mélange de corpus PTB et les 9 blocs de T-POS, avec la même répartition aléatoire du corpus Twitter T-POS que dans la partie précédente.

L'algorithme d'optimisation utilisé pour tous les apprentissages de wapiti est rprop+.

### 2.5.1 Validation croisée avec T-POS uniquement

Dans cette section, nous avons testé nos patrons en validation croisée à 10 blocs, en calculant l'exactitude moyenne obtenue sur l'ensemble du jeu d'étiquettes Ritter (49 étiquettes). Pour optimiser la régularisation de la log-vraisemblance du CRF, pondérée par les paramètres L1 et L2, nous avons d'abord fixé L2 à 0.00001 (valeur par défaut dans wapiti) et nous avons testé toutes les valeurs possibles de L1 dans l'ensemble suivant :

$$\{0.01, 0.03, 0.1, 0.3, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$$

. Ensuite, nous avons gardé la valeur de L1 qui donne le meilleur résultat et nous avons fait varier la valeur de L2 dans l'ensemble :

$$\{0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1\}$$

en gardant de nouveau celle qui donne le meilleur résultat.

Jeu d'étiquettes	L1	L2	Moyenne d'exactitude
Ritter	0.1	1	87.21%
Universal	0.01	1	89.25%

TABLE 2.4 – Validation croisée avec le corpus T-POS

D'après les résultats du tableau 2.4, nous remarquons que notre jeu d'étiquettes universel permet d'augmenter de 3% l'exactitude des étiquettes morpho-syntaxiques, bien que ce jeu entraîne de nouvelles séquences d'étiquettes auparavant impossibles. Ce résultat nous laisse espérer mieux analyser morpho-syntaxiquement les données de Twitter.

### 2.5.2 Modèles mixtes

Pour construire des modèles mixtes, trois différentes proportions de données issues du PTB ont été ajoutées à celles initialement disponibles en apprentissage (en validation croisée) dans T-POS : le même nombre de séquences, 4 fois plus et 9 fois plus. Les données de ces trois parties de PTB sont

disjointes. Les ensembles d'apprentissage des corpus mixtes sont construits de la manière suivante : pour chaque itération de la validation croisée, les données provenant du corpus PTB restent les mêmes, seule la partie du corpus T-POS change. Les résultats figurent dans le tableau 2.5. L'évaluation des modèles est calculée par la moyenne des 10 blocs. Ensuite, l'optimisation des paramètres L1 et L2 se fait de manière identique à la partie précédente.

Jeu d'étiquettes	Proportion	L1	L2	Moyenne d'exactitude
Ritter	1 :1	1.5	0.0001	85.40%
Ritter	4 :1	1.8	0.001	86.72%
Ritter	9 :1	1.9	0.0001	87.18%
Universal	1 :1	1.0	0.01	89.11%
Universal	4 :1	1.0	0.5	89.27%
Universal	9 :1	1.6	0.01	88.95%

TABLE 2.5

Ce résultat montre qu'avec le jeu d'étiquettes Ritter, le fait d'ajouter des textes issus du PTB n'augmente pas vraiment l'exactitude par rapport aux validations croisées (qui était de 87.21%), C'est sans doute dû au fait que les deux corpus ne se ressemblent pas suffisamment : les nouvelles données introduisent de nouveaux tokens et de nouvelles séquences d'étiquettes qui n'aident pas à mieux reconnaître celles de Twitter. L'effet semble légèrement moindre avec le jeu d'étiquettes universel, qui permet une très légère amélioration. Nous remarquons néanmoins qu'ajouter successivement plus de données du PTB dans l'ensemble d'apprentissage (mêlées avec celles du corpus T-POS) améliore les performances.

## 2.6 Conclusion et perspectives

Dans ce chapitre, nous proposons tout d'abord un jeu d'étiquettes réduit par rapport aux 45 étiquettes du PTB, qui sera facilement exploitable pour d'autres langues tout en préservant les spécificités de Twitter. Ce jeu d'étiquettes rend l'étiqueteur morpho-syntaxique plus fiable en regroupant entre elles les catégories similaires/proches. Mais sa limite est qu'il ne distingue pas bien certaines catégories. Une nouvelle version du jeu de Tags universel a d'ors et déjà été proposée dans <http://universaldependencies.github.io/docs/u/pos/index.html>. Cette version sépare les conjonctions de subordination des prépositions, les auxiliaires des verbes, les symboles et les interjections des X et les noms propres des noms communs. Nous



avons déjà pris en compte cette dernière distinction. Comme la prochaine étape de notre travail est d'analyser les opinions dans les tweets, nous allons tester l'impact des deux jeux d'étiquettes pour cette analyse d'opinions.

D'autre part, nous avons essayé d'apprendre un étiqueteur morpho-syntaxique à partir de textes écrits et de tweets, pour étiqueter des tweets. Les résultats de ces expériences montrent qu'ajouter des données éloignées de la cible dans la phase d'apprentissage permet d'améliorer l'exactitude. Cependant, pour apprendre un étiqueteur plus performant, rien ne vaut l'augmentation de la taille des données d'apprentissage proches de la cible. À défaut, il faudrait envisager l'utilisation de ressources linguistiques externes ou de données non étiquetées.

Une autre piste serait de trouver des patrons plus adaptés aux données de Twitter (nous avons vu que les bigrammes d'étiquettes avec les caractéristiques ne fonctionnent pas). Il peut être aussi intéressant d'ajouter d'autres caractéristiques pertinentes. L'exemple donné au début du chapitre dans la figure 2.1 suggère l'utilisation de transcriptions phonétiques pour normaliser des tokens comme "wasz" ou "cusz", comme proposé dans CLARK, 2003.

Des clusters de grandes quantités de tweets comme dans FARHAD, CAROLINE et CLAUDE, 2014, une optimisation des paramètres L1 et L2 à la façon de DINARELLI et ROSSET, 2012 sont aussi des pistes exploitables pour cette tâche. Enfin, un traitement spécifique des hashtags (s'ils jouent un vrai rôle syntaxique) pourrait aussi permettre un étiquetage plus fin et plus régulier des messages.

## Chapitre 3

# Détection des mots non-standards dans les tweets avec des réseaux de neurones

Dans ce chapitre, nous proposons un modèle pour détecter dans les textes générés par des utilisateurs (en particulier les tweets), les mots non-standards à corriger. Nous utilisons pour cela des réseaux de neurones convolutifs au niveau des caractères, associés à des “plongements” (embeddings) des mots présents dans le contexte du mot courant. Nous avons utilisé pour l’évaluation trois corpus de référence. Nous avons testé différents modèles qui varient suivant leurs plongements pré-entraînés, leurs configurations et leurs optimisations. Nous avons finalement obtenu une F1-mesure de 0.972 en validation croisée pour la classe des mots non-standards. Cette détection des mots à corriger est l’étape préliminaire pour la normalisation des textes non standards comme les tweets.

### 3.1 Introduction

Les contenus générés par des utilisateurs sur le web, par exemple via les tweets, sont devenus une source de données importante. Ces textes, sous la forme de petits messages d’au maximum 140 caractères, véhiculent souvent des opinions sur l’actualité ou sur des produits de consommation. Ils sont donc particulièrement intéressants à analyser automatiquement dans le cadre de la fouille d’opinion. Cependant, les outils traditionnels de Traitement Automatique des Langues (TAL) se comportent rarement bien face à ces textes, courts et souvent écrits dans une langue non standard. Par exemple, le Stanford POS Tagger, étiqueteur morpho-syntaxique appris avec le Penn

TreeBank<sup>1</sup> TOUTANOVA et MANNING, 2000 atteint 96.86% d'exactitude sur les textes journalistiques, mais seulement 81.3% sur les tweets (RITTER et al., 2011). Ce phénomène se retrouve aussi pour d'autres tâches de TAL, comme l'analyse syntaxique PLANK et al., 2014; KONG et al., 2014. La *normalisation de texte* est la tâche qui consiste à transformer *a minima* un texte "non standard" pour le faire se rapprocher le plus possible d'un texte "standard" sans modifier son sens. C'est devenu une étape de prétraitement nécessaire pour certains textes, avant d'essayer de leur appliquer d'autres étapes de TAL (HAN, COOK et BALDWIN, 2013). Elle opère pour l'instant essentiellement au niveau lexical, par l'identification et le remplacement des mots (ou "tokens") non standards. Beaucoup de travaux montrent qu'une telle normalisation appliquée sur des tweets peut améliorer le résultat de l'extraction des entités nommées qu'ils contiennent (NGUYEN, NGUYEN et SNAEL, 2016; LIU, WENG et JIANG, 2012). Dans certains cas, la détection des unités non-standards peut à elle seule améliorer ce résultat (LI et LIU, 2015).

Notre travail dans ce chapitre s'inspire de certaines expériences de LI et LIU, 2015. Il vise au final à améliorer l'extraction des entités nommées dans les tweets après une phase de normalisation. Nous nous focalisons en particulier sur le pré-traitement lexical permettant de rendre les tweets un peu mieux formés, en remplaçant les mots non-standards qui y figurent par une forme standard. Nous essayons ici, dans un premier temps, de détecter dans un tweet ces mots non-standards avant de proposer une liste de candidats pour les corriger.

Nous proposons pour cela un modèle de réseau de neurones convolutifs au niveau des caractères, associé à des "plongements" de mots, pour déterminer si un mot est non-standard (à corriger). Nous avons utilisé pour nous évaluer des corpus de tweets en anglais dans lesquels les mots non standards ont été manuellement corrigés et avons mené diverses expériences avec divers protocoles (en validation croisée, ou avec un autre corpus d'évaluation).

## 3.2 Etat de l'art

### 3.2.1 Notion de mots non-standard

Les mots non-standards (non-standard words, NSW) sont des mots à corriger dans le contexte où ils apparaissent par un mot présent dans un vocabulaire. Ils peuvent contenir des fautes d'orthographe (comme "dis" pour

---

1. Stanford Pos Tagger : <http://nlp.stanford.edu/software/tagger.shtml>

"this" en anglais), des acronymes (comme "lol" pour "laughing out loud"), des abréviations (comme "u" pour "you"), etc. Cette notion de NSW se distingue de celle de mots hors vocabulaire (out-of-vocabulary, OOV). En effet, les OOV sont des mots absents d'un dictionnaire de référence (par exemple celui de GNU Aspell pour l'anglais<sup>2</sup>), mais ne sont pas nécessairement pour autant mal écrits et à corriger : les entités nommées ou noms de produits sont souvent OOV. Ainsi, un mot comme "xanax" (nom d'un médicament contre l'anxiété), ne se trouve pas dans les dictionnaires. Ce n'est cependant pas un NSW parce qu'il n'y a pas moyen de le corriger, il est déjà sous sa forme correcte. Les NSW des exemples précédents sont des OOV, mais il y a aussi des NSW qui figurent dans un vocabulaire, mais ne conviennent pas dans leur contexte, par exemple "wit" dans la phrase "I'll go wit you" (la forme correcte est "with") HAN, 2014. Le contexte joue donc un rôle très important pour ranger les mots dans ces différentes classes. Enfin, du bruit peut aussi exister dans les tweets, comme "bahahahaha" à la fin d'un message, pour exprimer un sentiment. Ce token est OOV parce qu'il n'est pas présent dans les dictionnaires usuels, mais il n'existe pas non plus de forme correcte correspondante. Il ne s'agit donc pas d'un NSW. Certaines fois, il est difficile de juger si un token est NSW ou SW comme "footie", "y'all" et "yous" BALDWIN et al., 2015.

### 3.2.2 Normalisation des tweets

La normalisation lexicale de texte peut se décomposer en trois sous-tâches :

1. Trouver les NSW de ce texte. Dans un corpus de tweets, le taux de NSW est d'environ 10%. Cette sous-tâche se traduit souvent par une classification binaire (même si, comme nous venons de le voir, la situation est souvent plus compliquée à cause du bruit). Il s'agit donc à cette étape de décider, pour chaque token d'un texte, s'il est à corriger ou pas.
2. Pour un NSW dans un texte, proposer une liste de candidats pour le corriger. Si un token peut être considéré comme candidat à sa propre correction, alors cette sous-tâche peut être exécutée pour tous les tokens d'un texte, sans passer par la première sous-tâche comme dans JIN, 2015 et MAUSAM et al., 2012. HAN et BALDWIN, 2011 a proposé de générer les corrections des NSW en se basant sur les similarités morpho-phonétiques. SUPRANOVICH et PATSEPNIA, 2015 a, lui, fait

---

2. GNU Aspell : <http://aspell.net/>

appel à la distance d'édition (ou distance de Levenshtein) MILLER, VANDOME et MCBREWSTER, 2009.

3. Pour un NSW et une liste de candidats de correction proposée, trouver le candidat qui a la plus forte probabilité d'être la forme correcte du NSW. Cela peut être réalisé en calculant des mesures de similarité entre le NSW et chacun de ses candidats comme dans JIN, 2015, ou par la distance d'édition MILLER, VANDOME et MCBREWSTER, 2009 comme SUPRANOVICH et PATSEPNIA, 2015.

Pour la première sous-tâche, HAN et BALDWIN, 2011; SUPRANOVICH et PATSEPNIA, 2015; LEEMAN-MUNK, LESTER et COX, 2015 ont utilisé un classifieur pour détecter les NSW. SUPRANOVICH et PATSEPNIA, 2015 a utilisé les champs conditionnels aléatoires (Conditional Random Fields, CRF) tandis que LEEMAN-MUNK, LESTER et COX, 2015 ont utilisé un réseau de neurones pour le faire. BEREND et TASNÁDI, 2015 a quant à lui exploité les CRF pour la classification et la correction des NSW.

La correction de tweets a fait l'objet d'un challenge dans le workshop WNUT (Workshop on Noisy User-generated Text BALDWIN et al., 2015). Les meilleurs résultats ont été obtenus par LEEMAN-MUNK, LESTER et COX, 2015, qui ont atteint un résultat de 0,98 de F1-mesure pour la classe NSW sur le corpus de développement du workshop, et le même résultat sur le corpus d'évaluation. Dans ce chapitre, nous essayons de reprendre la méthode qu'ils ont utilisée pour détecter dans des tweets les mots non-standards (NSW), considéré comme une tâche de classification binaire.

### 3.3 Les corpus annotés

Nous avons utilisé trois corpus annotés : 2577 tweets provenant de LI et LIU, 2014 d'une part, les corpus de développement et d'évaluation du workshop WNUT d'autre part HAN et BALDWIN, 2011; BALDWIN et al., 2015. Le tableau 3.1 fournit les statistiques de base de ces trois corpus.

nom de corpus	nombre de tweets	# tokens	# NSW	# SW	taille de vocabulaire
2577 tweets	2577	44385	3942	40443	8270
WNUT dev	2950	40560	4274	36286	15108
WNUT eval	1698	29421	2776	26645	10765

TABLE 3.1 – Corpus statistiques

Ces trois corpus sont tokenisés puis annotés avec la correction manuelle de NSW. Pour la convenance des annotateurs, les annotations (les corrections

de tokens) sont en minuscules. Nous avons donc utilisé uniquement l’annotation pour distinguer les NSW des SW pour notre tâche de détection des NSW : si la forme de surface d’un token dans le corpus est différente de sa correction (la différence de majuscules et de minuscules sont ignorées), ce token est considéré comme un NSW.

D’après les statistiques du tableau 3.1, les mots non-standards sont minoritaires, ils constituent respectivement 8.9%, 10.5% et 9.4% du nombre total des tokens de ces trois corpus.

### 3.4 Les modèles de réseaux de neurones convolutifs

Les modèles de réseaux de neurones convolutifs (CNN) introduits dans COLLOBERT et al., 2011 sont utilisés pour extraire les propriétés de haut niveau des vecteurs de mots. Dans MA et HOVY, 2016, un modèle convolutif similaire au niveau des caractères est aussi utilisé pour représenter des propriétés plus fines. Dans ce chapitre, nous proposons un modèle de réseau de neurones convolutifs au niveau des caractères, associé à des plongements. La figure 3.1 montre le détail de la représentation du mot “Mercure” par la convolution des caractères comme dans MA et HOVY, 2016.

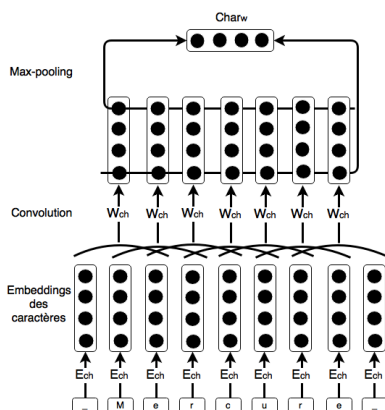


FIGURE 3.1 – structure du CNN sur les caractères utilisé

Dans cette figure, les  $E_{ch}$  désignent les plongements (embeddings) des caractères (un vecteur pour représenter un caractère, de taille quatre sur le dessin). Les symboles "\_" avant le "M" et après le "e" du mot servent à le compléter pour la fenêtre de taille trois autour d’un caractère. Les  $W_{ch}$  sont

les poids de la matrice correspondant à un triplet de caractères. Enfin le max-pooling prend uniquement la valeur la plus élevée de chaque composante du vecteur pour arriver à une représentation du mot "Mercure" ( $Char_w$ ). Dans notre modèle, la taille de ce vecteur est en réalité 300 (comme la configuration de modèle word2vec de Google).

Nous avons ensuite ajouté à la représentation du mot cible ainsi obtenue son contexte dans une fenêtre couvrant deux mots avant et deux mots après lui, représentés avec des plongements de mots classiques ( $Emb(W_i)$ , décrits plus loin). Les débuts et la fin de tweets sont marqués par  $\langle s \rangle \langle s \rangle$  et  $\langle /s \rangle \langle /s \rangle$  pour constituer une suite de mots de taille constante (5-grammes). Ainsi, un tweet de N mots sera présenté par N 5-grammes. Par exemple, pour un tweet avec six tokens : "I txd you last night." il y a six 5-grammes qui représentent cette séquence de texte : " $\langle s \rangle \langle s \rangle$  I txd you", " $\langle s \rangle$  I txd you last", "I txd you last night", "txd you last night .", "you last night .  $\langle /s \rangle$ " et "last night .  $\langle /s \rangle \langle /s \rangle$ ". La structure complète du modèle est illustrée dans le schéma 3.2.

Emb( $W_{-2}$ )	Emb( $W_{-1}$ )	Emb( $W_0$ )	Emb( $W_1$ )	Emb( $W_2$ )	représentation du mot
Dropout(0.2)					
Couche cachée (dimension = 128, activation = ReLU)					
Dropout(0.2)					
Couche de sortie (dimension = 2, activation = softmax), loss = categorical_crossentropy					

TABLE 3.2 – Structure du réseau

### 3.5 Experiences et résultats

Notre baseline1 est une méthode simple par dictionnaire induit sur les données d’entraînement : pour chaque token des données d’apprentissage, s’il est au moins une fois marqué comme NSW, il est stocké dans un ensemble (la casse est toujours ignorée comme convenu). Dans la phase de test, si le token est dans cet ensemble, il sera taggé comme NSW, et comme SW sinon. Cette baseline1 reconnaît donc uniquement les NSW déjà présents dans les données d’apprentissage ; les autres tokens des données de test seront taggés comme SW.

Notre baseline2 est une méthode similaire de la baseline1. La différence est que, au lieu d’utiliser les données de l’apprentissage, nous avons utilisé le dictionnaire Aspell et nous avons marqué tous les tokens présents du dictionnaire Aspell (in vocabulary, IV) comme les SW, et NSW sinon (out-of-vocabulary, OOV). Le résultat est présenté dans le tableau 3.4.

Nous avons aussi essayé d'implémenter avec Keras le réseau de neurones proposé dans LEEMAN-MUNK, LESTER et COX, 2015. Par manque de détail, nous avons essayé d'optimiser RMS prop (TIELEMAN et HINTON, 2012), Xavier initialisation (GLOROT et BENGIO, 2010), et l'entropie croisée catégorielle (categorical crossentropy) comme fonction de perte. Nous avons fait ensuite 150 itérations comme dans leur configuration mais en gardant uniquement le meilleur modèle qui a le score f1 mesure pour les NSW le plus élevé. Le résultat sur les données d'évaluation est montré dans le tableau 3.4.

Nous avons procédé à diverses expériences avec notre réseau de neurones convolutif. Le tableau 3.3 montre les résultats de la validation croisée en 5 blocs avec le corpus WNUT dev et avec le corpus WNUT dev plus le corpus de 2577 tweets. Le tableau 3.4 montre les résultats d'évaluation obtenus sur WNUT test par un modèle appris sur WNUT dev et sur WNUT dev plus 2577 tweets. Ces derniers résultats sont donc comparables avec les travaux de LEEMAN-MUNK, LESTER et COX, 2015.

Les conditions d'expérimentation sont les suivantes :

- Logiciel d'implémentation : Keras<sup>3</sup>, Theano<sup>4</sup>
- Initialisation des paramètres : Xavier initialisation GLOROT et BENGIO, 2010
- Plongements pré-entraînés des mots ( $Emb(W_i)$ ) : Google word2vec MIKOLOV et al., 2013 versus *interne*.

Le modèle *interne* est appris avec word2vec sur des textes plus longs et mieux formés que les tweets (news, forums, blogs, etc.) mais portant sur des domaines similaires (produits de luxe, automobiles et musique). La taille du vocabulaire de ce modèle est d'environ deux millions de tokens différents, tandis qu'il y a trois millions de tokens différents dans le modèle word2vec de Google. Certaines entités nommées comme les dates, heures, unités de longueur, de volumes, etc. sont remplacées par un token artificiel (par exemple "\_\_DATE\_\_" pour les dates). Les tokens qui ne varient que suivant la casse sont aussi normalisés par leur forme la plus courante. Par exemple, le token "nissan" apparaît sous trois formes : tout en minuscule, tout en majuscule et "Nissan". La forme "Nissan" a le plus grand nombre d'occurrences dans le corpus *interne*, les formes "NISSAN" et "nissan" sont donc remplacées par "Nissan". Le modèle de word2vec *interne* est ainsi appris sur un vocabulaire restreint. En revanche, le modèle word2vec de Google MIKOLOV et al., 2013 a gardé toutes les formes pour tous les tokens. Donc les tokens "NISSAN",

---

3. Keras : <https://keras.io/>

4. Theano : <http://deeplearning.net/software/theano/>



"Nissan" et "nissan" y ont des représentations vectorielles différentes (parce que leurs contextes sont aussi différents).

Nous avons testé les configurations suivantes :

- Corpus d'apprentissage : WNUT dev versus WNUT dev + 2577 tweets
- Optimiseurs : RMS prop TIELEMAN et HINTON, 2012, adadelta ZEILER, 2012, adagrad DUCHI, HAZAN et SINGER, 2011, sgd BERGSTRA et BENGIO, 2012

Dans le tableau 3.3, les expériences en validation croisée avec 5 plis sont effectuées d'abord avec seulement le corpus de WNUT dev en premières lignes, puis sur les corpus 2577 tweets et WNUT dev mélangés (les dernières lignes). Pour les expériences sur le corpus WNUT eval, le modèle est appris avec seulement le corpus de WNUT dev pour les premières lignes, puis avec les corpus 2577 tweets et WNUT dev ensemble.

Training data	Word2vec	Optimizer name	précision NSW	rappel NSW	f1 measure NSW	
WNUT dev	baseline1		0.6382	0.5733	0.6092	
	google	rmsprop	0.8859	0.7721	0.8242	
		adadelta	0.8813	0.7976	0.8372	
		adagrad	0.8941	0.7819	0.8341	
		sgd	0.8823	0.8127	0.8459	
	interne	rmsprop	0.9047	0.8013	0.8496	
		adadelta	0.911	0.8022	0.853	
		adagrad	0.9157	0.8062	0.8574	
		sgd	0.9022	0.824	0.8612	
	WNUT dev+2577	baseline1		0.3012	0.5388	0.3864
		google	rmsprop	0.9727	0.9501	0.9613
			adadelta	0.9813	0.9576	0.9693
adagrad			0.9791	0.9566	0.9677	
sgd			0.9812	0.9597	0.9703	
interne		rmsprop	0.9808	0.9545	0.9674	
		adadelta	0.9841	0.9620	<b>0.9729</b>	
		adagrad	0.9876	0.9583	0.9727	
		sgd	0.9853	0.9602	0.9725	

TABLE 3.3 – Résultat en validation croisée

Nous pouvons observer dans ces résultats :

1. sans surprise, plus le corpus d'entraînement est grand, meilleurs sont les résultats : WNUT dev seul produit des modèles moins performants que WNUT dev+2577 tweets.
2. plongements : les plongements de Google sont moins efficaces que ceux développés en *interne* (nous avons aussi testé sans plongements pré-entraînés, le résultat n'est pas meilleur)
3. optimiser : le meilleur optimiser varie selon les expériences. En général, RMS prop donne le résultat le moins bon. Le sgd donne souvent un meilleur résultat.

Training data	Word2vec	Optimizer name	précision NSW	rappel NSW	f1 measure NSW
WNUT dev	baseline1		0.3999	0.3542	0.3757
	baseline2		0.1304	0.8066	0.2245
	LEEMAN-MUNK, LESTER et COX, 2015		0.9237	0.3098	0.4640
	LEEMAN-MUNK, LESTER et COX, 2015 Keras		0.	0.	0.
	google	rmsprop	0.7385	0.7965	0.7664
		adadelta	0.6952	0.8307	0.7569
		adagrad	0.7835	0.7954	0.7894
		sgd	0.7964	0.8188	0.8075
	interne	rmsprop	0.6023	0.8260	0.6966
		adadelta	0.7961	0.7821	0.7890
adagrad		0.9157	0.8062	0.8574	
sgd		0.8931	0.7673	0.8254	
WNUT dev+2577	baseline1		0.3943	0.6459	0.4897
	google	rmsprop	0.6160	0.8606	0.7181
		adadelta	0.7000	0.8455	0.7659
		adagrad	0.8139	0.7893	0.8014
		sgd	0.7859	0.8184	0.8018
	interne	rmsprop	0.6254	0.8289	0.7129
		adadelta	0.8257	0.7849	0.8048
		adagrad	0.9038	0.7478	0.8145
		sgd	0.8769	0.7648	<b>0.8170</b>

TABLE 3.4 – Résultat d'évaluation

(nous avons testé aussi  $\text{decay} = \text{learn rate} / \text{nb d'epochs}$ , le résultat n'est pas meilleur)

### 3.6 Conclusion et perspectives du travail

Nous proposons dans ce chapitre un modèle de réseau de neurones convolutif pour détecter dans des tweets les mots non-standards (NSW) à corriger. Nos expériences montrent que la quantité de données d'apprentissage influe sensiblement sur le résultat. Le modèle word2vec *interne* et l'optimiser SGD produisent le meilleur classifieur de NSW, avec une F1-mesure d'environ 0,83 pour cette classe sur le corpus d'évaluation. Ce travail servira de base pour une étape de normalisation ultérieure des tweets.

# Bibliographie

- ABEILLÉ, A., L. CLÉMENT et F. TOUSSENEL (2003). « Building a treebank for French ». In : *Treebanks*. Sous la dir. d'A. ABEILLÉ. Dordrecht : Kluwer.
- BALDWIN, Timothy et al. (2015). « Shared Tasks of the 2015 Workshop on Noisy User-generated Text : Twitter Lexical Normalization and Named Entity Recognition ». In : *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China : Association for Computational Linguistics, p. 126–135. URL : <http://www.aclweb.org/anthology/W15-4319>.
- BEREND, Gábor et Ervin TASNÁDI (2015). « USZEGED : Correction Type-sensitive Normalization of English Tweets Using Efficiently Indexed n-gram Statistics ». In : *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China : Association for Computational Linguistics, p. 120–125. URL : <http://www.aclweb.org/anthology/W15-4318>.
- BERGSTRA, James et Yoshua BENGIO (2012). « Random Search for Hyperparameter Optimization ». In : *J. Mach. Learn. Res.* 13, p. 281–305. ISSN : 1532-4435. URL : <http://dl.acm.org/citation.cfm?id=2188385.2188395>.
- CLARK, Alexander (2003). « Combining Distributional and Morphological Information for Part of Speech Induction ». In : *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*. EAACL '03. Budapest, Hungary : Association for Computational Linguistics, p. 59–66. ISBN : 1-333-56789-0. DOI : 10.3115/1067807.1067817. URL : <http://dx.doi.org/10.3115/1067807.1067817>.
- COLLOBERT, Ronan et al. (2011). « Natural Language Processing (Almost) from Scratch ». In : *J. Mach. Learn. Res.* 12, p. 2493–2537. ISSN : 1532-4435. URL : <http://dl.acm.org/citation.cfm?id=1953048.2078186>.
- CONSTANT, Matthieu et al. (2011). « Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français ». In : *TALN*. T. 1. Montpellier, France, p. 321. URL : <https://hal.archives-ouvertes.fr/hal-00620923>.
- CROCHEMORE, Maxime, Christophe HANCART et Thierry LECROQ (2007). *Algorithms on strings*. Cambridge University Press.

- DINARELLI, Marco et Sophie ROSSET (2012). *Tree-Structured Named Entity Recognition on OCR Data : Analysis, Processing and Results*. Anglais. Sous la dir. de Nicoletta Calzolari (Conference CHAIR) et al. Istanbul, Turkey.
- DUCHI, John, Elad HAZAN et Yoram SINGER (2011). « Adaptive Subgradient Methods for Online Learning and Stochastic Optimization ». In : *J. Mach. Learn. Res.* 12, p. 2121–2159. ISSN : 1532-4435. URL : <http://dl.acm.org/citation.cfm?id=1953048.2021068>.
- EHRMANN, Maud (2008). « Les Entités nommées, de la linguistique au TAL : statut théorique et méthodes de désambiguïsation ». Université Paris 7 – Denis Diderot.
- FARHAD, Nooralahzadeh, Brun CAROLINE et Roux CLAUDE (2014). « Part of Speech Tagging for French Social Media Data ». In : *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference : Technical Papers, August 23-29, 2014, Dublin, Ireland*, p. 1764–1772. URL : <http://aclweb.org/anthology/C/C14/C14-1166.pdf>.
- FORSYTH, Eric Nielsen (2007). *Improving Automated Lexical and Discourse Analysis of Online Chat Dialog*. Los Alamitos, CA, USA. DOI : <http://doi.ieeecomputersociety.org/10.1109/ICSC.2007.55>.
- FOSTER, J. et al. (2011). « # hardtoparse : POS Tagging and Parsing the Twitverse ». In : *proceedings of the Workshop On Analyzing Microtext (AAAI 2011)*, p. 20–25. URL : [http://scholar.google.co.uk/scholar.bib?q=info:e2OU-YlfaKkJ:scholar.google.com/&output=citation&scisig=AAGBfm0AAAAAUQKz2MvFN5rEWNVP\\_Dz19DRLnajmBI&scisf=4&hl=en&as\\_sdt=0,5&scfhb=1](http://scholar.google.co.uk/scholar.bib?q=info:e2OU-YlfaKkJ:scholar.google.com/&output=citation&scisig=AAGBfm0AAAAAUQKz2MvFN5rEWNVP_Dz19DRLnajmBI&scisf=4&hl=en&as_sdt=0,5&scfhb=1).
- GHANI, Rayid et al. (2006). « Text mining for product attribute extraction ». In : *SIGKDD Explor. Newsl.* 8.1, p. 41–48. ISSN : 1931-0145. DOI : [10.1145/1147234.1147241](http://doi.acm.org/10.1145/1147234.1147241). URL : <http://doi.acm.org/10.1145/1147234.1147241>.
- GIMPEL, Kevin et al. (2011). « Part-of-speech Tagging for Twitter : Annotation, Features, and Experiments ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : Short Papers - Volume 2*. HLT '11. Portland, Oregon : Association for Computational Linguistics, p. 42–47. ISBN : 978-1-932432-88-6. URL : <http://dl.acm.org/citation.cfm?id=2002736.2002747>.
- GLOROT, Xavier et Yoshua BENGIO (2010). « Understanding the difficulty of training deep feedforward neural networks ». In : *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. Society for Artificial Intelligence and Statistics.

- HAN, Bo (2014). « Improving the utility of social media with Natural Language Processing ». Thèse de doct. The University of Melbourne.
- HAN, Bo et Timothy BALDWIN (2011). « Lexical Normalisation of Short Text Messages : Makn Sens a #Twitter ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies - Volume 1. HLT '11*. Portland, Oregon : Association for Computational Linguistics, p. 368–378. ISBN : 978-1-932432-87-9. URL : <http://dl.acm.org/citation.cfm?id=2002472.2002520>.
- HAN, Bo, Paul COOK et Timothy BALDWIN (2013). « Lexical Normalization for Social Media Text ». In : *ACM Trans. Intell. Syst. Technol.* 4.1, 5 :1–5 :27. ISSN : 2157-6904. DOI : [10.1145/2414425.2414430](https://doi.org/10.1145/2414425.2414430). URL : <http://doi.acm.org/10.1145/2414425.2414430>.
- JIN, Ning (2015). « NCSU-SAS-Ning : Candidate Generation and Feature Engineering for Supervised Lexical Normalization ». In : *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China : Association for Computational Linguistics, p. 87–92. URL : <http://www.aclweb.org/anthology/W15-4313>.
- KONG, Lingpeng et al. (2014). « A Dependency Parser for Tweets ». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, p. 1001–1012. URL : <http://www.aclweb.org/anthology/D14-1108>.
- LAFFERTY, J., A. MCCALLUM et F. PEREIRA (2001a). « Conditional random Fields : Probabilistic models for segmenting and labeling sequence data ». In : *Proceedings of ICML 2001*, p. 282–289.
- LAFFERTY, John D., Andrew MCCALLUM et Fernando C. N. PEREIRA (2001b). « Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data ». In : *Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., p. 282–289. ISBN : 1-55860-778-1. URL : <http://dl.acm.org/citation.cfm?id=645530.655813>.
- LAVERGNE, Thomas, Olivier CAPPÉ et François YVON (2010). « Practical Very Large Scale CRFs ». In : *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Uppsala, Sweden : Association for Computational Linguistics, p. 504–513. URL : <http://www.aclweb.org/anthology/P10-1052>.

- LAVERGNE, Thomas, Olivier CAPPÉ et François YVON (2010). « Practical Very Large Scale CRFs ». In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden : Association for Computational Linguistics, p. 504–513. URL : <http://dl.acm.org/citation.cfm?id=1858681.1858733>.
- LEEMAN-MUNK, Samuel, James LESTER et James COX (2015). « NCSU\_SAS\_SAM: Deep Encoding and Reconstruction for Normalization of Noisy Text ». In : *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China : Association for Computational Linguistics, p. 154–161. URL : <http://www.aclweb.org/anthology/W15-4323>.
- LI, Chen et Yang LIU (2014). « Improving Text Normalization via Unsupervised Model and Discriminative Reranking. » In : *ACL (Student Research Workshop)*, p. 86–93.
- (2015). « Improving named entity recognition in tweets via detecting non-standard words ». In : *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*. T. 1. Association for Computational Linguistics (ACL), p. 929–938. ISBN : 9781941643723.
- LIU, Fei, Fuliang WENG et Xiao JIANG (2012). « A Broad-coverage Normalization System for Social Media Language ». In : *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers - Volume 1*. ACL '12. Jeju Island, Korea : Association for Computational Linguistics, p. 1035–1044. URL : <http://dl.acm.org/citation.cfm?id=2390524.2390662>.
- LOPER, Edward et Steven BIRD (2002). « NLTK : The Natural Language Toolkit ». In : *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP '02. Philadelphia, Pennsylvania : Association for Computational Linguistics, p. 63–70. DOI : [10.3115/1118108.1118117](https://doi.org/10.3115/1118108.1118117). URL : <https://doi.org/10.3115/1118108.1118117>.
- MA, Xuezhe et Eduard HOVY (2016). « End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF ». In : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.
- MARCUS, Mitchell P., Beatrice SANTORINI et Mary Ann MARCINKIEWICZ (1993). « Building a Large Annotated Corpus of English : The Penn Treebank ». In : *COMPUTATIONAL LINGUISTICS* 19.2, p. 313–330.

- MAUSAM et al. (2012). « Open Language Learning for Information Extraction ». In : *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. EMNLP-CoNLL '12*. Jeju Island, Korea : Association for Computational Linguistics, p. 523–534. URL : <http://dl.acm.org/citation.cfm?id=2390948.2391009>.
- MCCALLUM, A. et W. LI (2003). « Early results for named entity recognition with conditional random fields ». In : *CoNLL'2003 : Proceedings of The Seventh Conference on Natural Language Learning*.
- MIKOLOV, Tomas et al. (2013). « Efficient Estimation of Word Representations in Vector Space ». In : *CoRR abs/1301.3781*. URL : <http://arxiv.org/abs/1301.3781>.
- MILLER, Frederic P., Agnes F. VANDOME et John MCBREWSTER (2009). *Levenshtein Distance : Information Theory, Computer Science, String (Computer Science), String Metric, Damerau-Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press. ISBN : 6130216904, 9786130216900.
- NGUYEN, Vu H., Hien T. NGUYEN et Vaclav SNASEL (2016). « Text normalization for named entity recognition in Vietnamese tweets ». In : *Computational Social Networks 3.1*, p. 10. ISSN : 2197-4314. DOI : [10.1186/s40649-016-0032-0](https://doi.org/10.1186/s40649-016-0032-0). URL : <http://dx.doi.org/10.1186/s40649-016-0032-0>.
- PETROV, Slav, Dipanjan DAS et Ryan MCDONALD (2012). « A Universal Part-of-Speech Tagset ». Anglais. In : *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Sous la dir. de Nicoletta Calzolari (Conference CHAIR) et al. Istanbul, Turkey : European Language Resources Association (ELRA). ISBN : 978-2-9517408-7-7.
- PLANK, Barbara et al. (2014). « Adapting taggers to Twitter with not-so-distant supervision ». In : *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference : Technical Papers, August 23-29, 2014, Dublin, Ireland*, p. 1783–1792. URL : <http://aclweb.org/anthology/C/C14/C14-1168.pdf>.
- POIBEAU, T. (2003). *Extraction automatique d'information*. Paris : Hermès.
- PUTTHIVIDHYA, Duangmanee (Pew) et Junling HU (2011). « Bootstrapped named entity recognition for product attribute extraction ». In : *Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11*. Edinburgh, United Kingdom : Association for Computational Linguistics, p. 1557–1567. ISBN : 978-1-937284-11-4. URL : <http://dl.acm.org/citation.cfm?id=2145432.2145598>.

- RAYMOND, Christian et Julien FAYOLLE (2010). « Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement ». Français. In : *TALN'10*. ATALA. Montréal, Québec, Canada. URL : <http://hal.inria.fr/inria-00561732>.
- RITTER, Alan et al. (2011). « Named Entity Recognition in Tweets : An Experimental Study ». In : *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom : Association for Computational Linguistics, p. 1524–1534. ISBN : 978-1-937284-11-4. URL : <http://dl.acm.org/citation.cfm?id=2145432.2145595>.
- SARAWAGI, Sunita (2008). « Information extraction ». In : *Foundations and trends in databases* 1.3, p. 261–377.
- SEDDAH, Djamé et al. (2012). *The French Social Media Bank : a Treebank of Noisy User Generated Content*. Mumbai, India : Kay, Martin and Boitet, Christian. URL : <https://hal.inria.fr/hal-00780895>.
- SHA, F. et F. PEREIRA (2003). « Shallow parsing with conditional random fields ». In : *Proceedings of HLT-NAACL 2003*, p. 213 –220.
- SUPRANOVICH, Dmitry et Viachaslau PATSEPNIA (2015). « IHS\_RD : Lexical Normalization for English Tweets ». In : *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China : Association for Computational Linguistics, p. 78–81. URL : <http://www.aclweb.org/anthology/W15-4311>.
- SUZUKI, Jun et Hideki ISOZAKI (2008). « Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data ». In : *In ACL*.
- TELLIER, Isabelle et Yoann DUPONT (2013). « Apprentissage symbolique et statistique pour le chunking : comparaison et combinaison ». In : *actes de TALN*.
- TELLIER, Isabelle et Marc TOMMASI (2011). « Champs Markoviens Conditionnels pour l'extraction d'information ». Français. In : *Modèles probabilistes pour l'accès à l'information textuelle*. Sous la dir. d'Eric GAUSSIER et François YVON. Hermès.
- TIELEMAN, T. et G. HINTON (2012). *Lecture 6.5—RmsProp : Divide the gradient by a running average of its recent magnitude*. COURSERA : Neural Networks for Machine Learning.
- TOUTANOVA, Kristina et Christopher D. MANNING (2000). « Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger ». In : *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora : Held in Conjunction*



*with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13. EMNLP '00. Hong Kong : Association for Computational Linguistics, p. 63–70. DOI : [10.3115/1117794.1117802](https://doi.org/10.3115/1117794.1117802). URL : <http://dx.doi.org/10.3115/1117794.1117802>.*

ZEILER, Matthew D. (2012). « ADADELTA : An Adaptive Learning Rate Method ». In : *CoRR* abs/1212.5701. URL : <http://arxiv.org/abs/1212.5701>.

## **Adaptation au domaine et combinaison de modèles pour l'annotation de textes multi-sources et multi-domaines**

Internet propose aujourd'hui aux utilisateurs de services en ligne de commenter, d'éditer et de partager leurs points de vue sur différents sujets de discussion. Ce type de contenu est maintenant devenu la ressource principale pour les analyses d'opinions sur Internet. Néanmoins, à cause des abréviations, du bruit, des fautes d'orthographe et toutes autres sortes de problèmes, les outils de traitements automatiques des langues, y compris les reconnaisseurs d'entités nommées et les étiqueteurs automatiques morpho-syntaxiques, ont des performances plus faibles que sur les textes bien-formés (RITTER et al., 2011).

Cette thèse a pour objet la reconnaissance d'entités nommées sur les contenus générés par les utilisateurs sur Internet. Nous avons établi un corpus d'évaluation avec des textes multi-sources et multi-domaines. Ensuite, nous avons développé un modèle de champs conditionnels aléatoires, entraîné sur un corpus annoté provenant des contenus générés par les utilisateurs.

Dans le but d'améliorer les résultats de la reconnaissance d'entités nommées, nous avons d'abord développé un étiqueteur morpho-syntaxique sur les contenus générés par les utilisateurs et nous avons utilisé les étiquettes prédites comme un attribut du modèle des champs conditionnels aléatoire. Enfin, pour transformer les contenus générés par les utilisateurs en textes bien-formés, nous avons développé un modèle de normalisation lexicale basé sur des réseaux de neurones pour proposer une forme correcte pour les mots non-standard.

**Mots-clés :** adaptation au domaine, reconnaissance des entités nommées, apprentissage automatique, champs aléatoires conditionnels, réseaux de neurones

### **Domain adaptation and model combination for the annotation of multi-source, multi-domain texts**

The increasing mass of User-Generated Content (UGC) on the Internet means that people are now willing to comment, edit or share their opinions on different topics. This content is now the main resource for sentiment analysis on the Internet. Due to abbreviations, noise, spelling errors and all other problems with UGC, traditional Natural Language Processing (NLP) tools, including Named Entity Recognizers and part-of-speech (POS) taggers, perform poorly when compared to their usual results on canonical text (RITTER et al., 2011).

This thesis deals with Named Entity Recognition (NER) on some User-Generated Content (UGC). We have created an evaluation dataset including multi-domain and multi-sources texts. We then developed a Conditional Random Fields (CRFs) model trained on User-Generated Content (UGC).

In order to improve NER results in this context, we first developed a POS-tagger on UGC and used the predicted POS tags as a feature in the CRFs model. To turn UGC into canonical text, we also developed a normalization model using neural networks to propose a correct form for Non-Standard Words (NSW) in the UGC.

**Keywords :** domain adaptation, named entity recognition, machine learning, conditional random fields, neural networks