INSTITUT NATIONAL DES LANGUES ET CIVILISATIONS ORIENTALES

Département Textes, Informatique, Multilinguisme

Anonymisation des adresses postales dans des documents non-structurés : comparaison des méthodes symboliques et statistiques

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Chloé LECOINTE

Directeur de mémoire :

Damien Nouvel

Encadrant :

Jugurtha Aït-Hamlat

Année universitaire 2017/2018

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

Remerciements

Je tiens à remercier en premier lieu toute l'équipe pédagogique de PluriTAL qui m'a offert une formation de qualité et m'a aidée au cours de mes études pendant quatre ans.

Je remercie particulièrement Damien Nouvel, maître de conférences à l'Inalco, pour avoir accepté de me guider tout au long de la rédaction de ce mémoire. Sa disponibilité et ses nombreux conseils m'ont été d'une aide précieuse.

Mes remerciements s'adressent également à mon tuteur de stage Jugurtha Aït-Hamlat, Chef de projet R&D Informatique et Sémantique, qui a toujours su trouver le temps de m'encadrer pendant ces six mois de stage.

Enfin, je remercie mes camarades de Master ainsi que mes collègues de travail, pour leur aide et leur bonne humeur.

RESUME

Le règlement général sur la protection des données est applicable dans l'ensemble des Etats membres de l'Union depuis le 25 mai 2018. Le principal objectif de ce règlement est d'accroitre la protection des personnes concernées par un traitement de leurs données à caractère personnel. Afin de les protéger, il convient donc d'anonymiser toutes données sensibles. La tâche d'anonymisation, qui est souvent liée à la tâche de reconnaissance des entités nommées, est le fil conducteur de ce mémoire. Notre travail se concentre essentiellement sur la comparaison de deux méthodes, une méthode symbolique et une méthode statistique, pour améliorer l'anonymisation des adresses physiques dans des courriels rédigés en anglais.

MOTS-CLES

Anonymisation, reconnaissance d'entité nommée, apprentissage automatique, transducteurs, adresse physique

PARTIE 1

I. INTRODUCTION

1.1 CONTEXTE DE L'ETUDE

SOMMAIRE

1.1.1 PRESENTATION DU PROJET

LE CONTEXTE

L'avènement du numérique conjugué à la croissance constante du nombre des innovations technologiques facilite aujourd'hui la collecte et le stockage de données. Que ce soit dans un domaine juridique ou médical, d'ordre public ou privé, chaque organisme est confronté au risque de divulgation de données sensibles, et plus particulièrement, à la question du risque de violation de la vie privée via l'utilisation de données personnelles. Ces risques sont d'autant plus réels qu'il existe de plus en plus de données partageables, favorisés par l'engagement des pays sur la voie de l'ouverture et du partage des données publiques[1].

Depuis le 25 mai 2018, la loi européenne du 27 avril 2016 concernant le Règlement général sur la protection des données (RGPD), est applicable dans tous les Etats membres de l'Union. Cette loi s'adresse à toutes les sociétés ayant des activités en Europe qui utiliseraient des données à caractère personnel, autrement dit, toutes informations pouvant se référer à une personne physique identifiée ou identifiable [RÈGLEMENT (UE) 2016/679]. Une donnée personnelle peut donc être : un nom, un prénom, une date de naissance, un numéro de sécurité sociale, une photo, des données

[1] Ou « *open data* ».

biométriques telles que des empreintes digitales, un numéro de téléphone, une carte de paiement ou bien encore une adresse.

Pour être conforme à la RGPD, les entreprises doivent anonymiser les données personnelles. La norme ISO/TS 25237 :2008 définit l'anonymisation comme « un processus qui supprime l'association entre l'ensemble de données identifiant et le sujet des données ». Cela signifie que, suite à un processus d'anonymisation, les données sont transformées de telle sorte que ces dernières ne puissent être ré-identifiées après traitement. Le procédé doit donc être irréversible.

SémantiWeb est une société qui se spécialise dans l'étude de la voix des consommateurs sur le web, grâce à des outils sémantiques. Elle possède, parmi ses données, des mails de consommateurs communiquant avec les clients (qui sont ici des entreprises de produits de beauté). Les raisons de ces mails sont diverses : ce sont des personnes qui se plaignent d'un produit ou qui le complimentent, qui témoignent d'un dysfonctionnement, qui proposent des partenariats, qui recherchent diverses informations, ou bien encore qui font des suggestions. De ce fait, il n'est pas rare de voir dans ces correspondances des informations personnelles qui doivent être "anonymisées" conformément à la RGPD.

Pour illustrer le contexte de l'étude, voici un exemple de mail normalisé[2] (*Figure 1*) que l'on pourrait trouver dans notre base de données (toutes les informations personnelles qui apparaissent dans ce mémoire ont été modifiées) :

Dear Urban Decay, My name is Mary and I'm an avid product tester, who immensely enjoys trying new products and spread the word with friends, family and online. I'm a budding product review blogger, currently using Instagram as my main platform. My Insta page is @tester256. Please feel free to check my page and explore how I'm'a trying to launch my own opinions and advise to others. I'd very much love to sample some of your products in exchange for a glowing review (I like to be as honest as possible). Should you feel this is something you'd be interested in my postal address is below. Mary 251 Green Hill Road Ashford Middle RT56 2TL I look forward to hearing from you Kind Regards Mary Allison

Figure 1 – Exemple de verbatim à anonymiser

LE PROCESSUS D'ANONYMISATION MIS EN PLACE PAR SEMANTIWEB

La méthode utilisée actuellement par Sémantiweb consiste à mettre au point des ressources linguistiques, essentiellement à base d'expressions régulières

[2] Avant de masquer les données sensibles contenues dans le mail, il est nécessaire de procéder à quelques prétraitements. Les verbatim sont donc tout d'abord normalisés : les sauts de lignes ou certaines balises html sont supprimés et les conflits d'encodages sont résolus.

(transducteurs), afin d'appliquer une transformation sur les données sensibles. Ces dernières, une fois reconnues par le système, sont substituées par des X.

Information	*Remplacement dans le verbatim*
Nom, prénom, adresse postale	XXXX
E-mail	xxxx@xxx.xx
Urls	www.xxxx.xx
Numéro de téléphone	xx-xx-xx-xx-xx ou xxx-xxxx-xxxx

Tableau 1- Exemple de transformations

Les transducteurs sont appliqués en cascade : l'ordre déclaré des expressions est primordial. Ainsi, certains transducteurs sont prévus pour réaliser des reconnaissances partielles afin de faciliter la reconnaissance des segments à anonymiser ou, au contraire, afin de les préserver de toute altération.

Pour chaque langue détectée est associé un ou plusieurs fichiers de transducteurs. Les langues couvertes par la société sont l'anglais, le français, l'espagnol, le portugais, l'italien, l'allemand et le néerlandais.

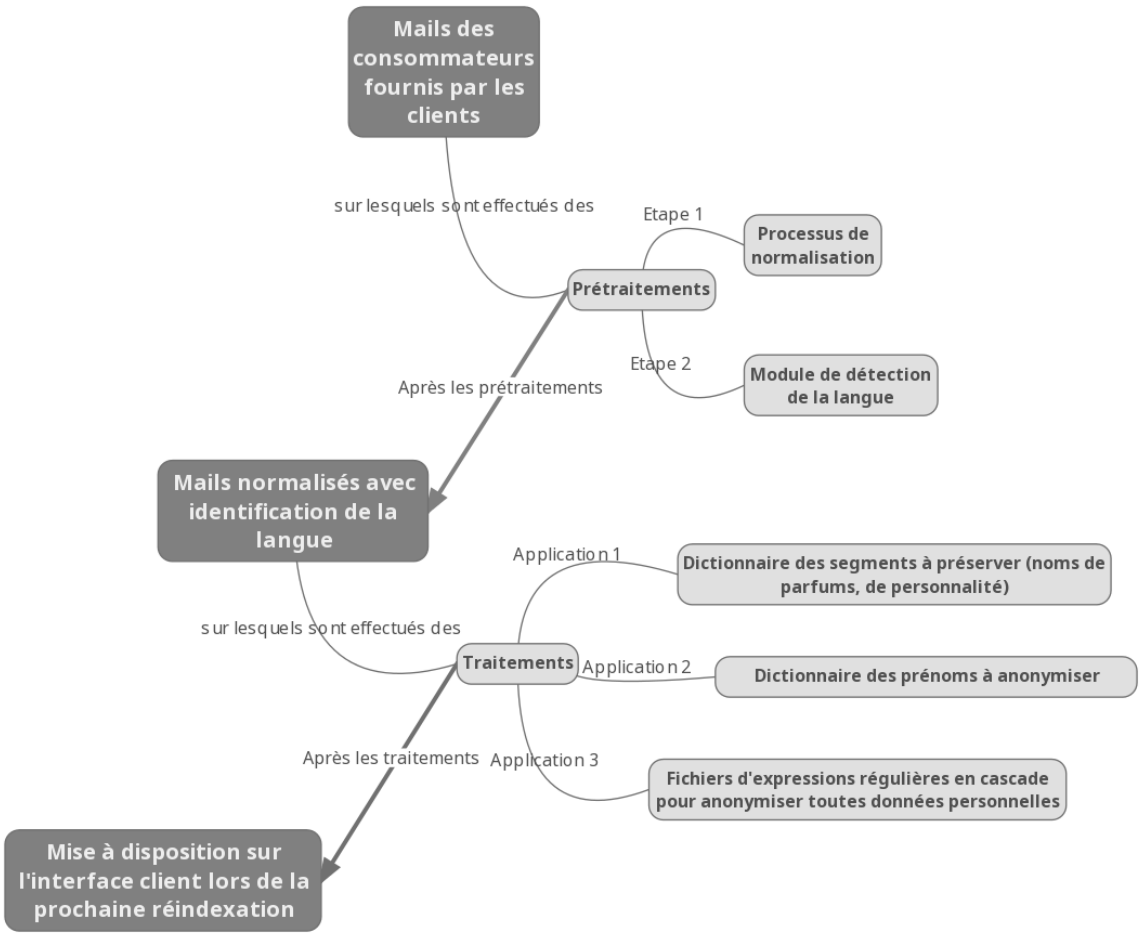


Figure 2 - Chaines de traitements de l'anonymisation actuelle

Tous les prétraitements et les traitements (*Figure 2*) sont implémentés en java. Les fichiers d'expressions régulières ainsi que les dictionnaires sont quant à eux des fichiers de textes bruts, chargés et exécutés par des modules Java.

1.1.2 L'EVALUATION DE LA METHODE ACTUELLE

LES CORPUS

Les langues qui ont été évaluées sont l'anglais, le français et l'allemand, en raison de leur forte occurrence et de la demande.

Langue	Français	Anglais	Allemand
Nombre de documents	965	813	792

Figure 3 - Les corpus d'évaluations

Chaque document correspond à un mail de 15 à 300 tokens. Une fois les modules d'anonymisations déjà mis en place par la société (méthode symbolique) appliqués sur les corpus, nous procédons à une évaluation humaine pour évaluer les résultats (voire *Partie 2 Chapitre 3 Sous-chapitre 2 – L'évaluation humaine*).

LES RESULTATS

Document	**Français**	**Anglais**	**Allemand**
Correctement anonymisé	86%	74%	71%
Partiellement anonymisé	5%	13%	21%
Non anonymisé	9%	13%	8%

Tableau 2 - Pourcentage d'anonymisation du français, anglais et allemand

On remarque que les erreurs sont de trois types : *(i)* une omission totale de l'anonymisation (exemple *a*), *(ii)* une omission partielle de l'anonymisation (exemple *b*), ou bien encore, *(iii)* une sur-anonymisation, qui masqueraient des éléments textuels ne faisant pas parties du segment à anonymiser (exemple *c*). L'omission totale de l'anonymisation est l'erreur la plus grave, tandis que la sur-anonymisation est jugée comme l'erreur étant la moins grave. Dans la plupart des cas, ces erreurs peuvent être caractérisées comme une mauvaise détection des frontières des éléments à anonymiser dans le courriel.

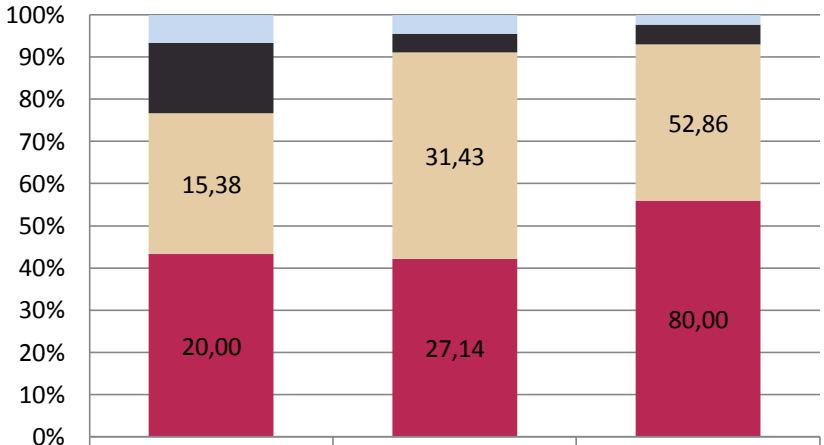
a) "I believe that a lot of people say that, because you really are the best. And i know that you will be the best forever, It was a leasure to slem like Giorgio Armani Si. Your faithfully, Sofia Lhuaer Valtia, Riga Strueh praspekts 87-65 TL-1023 sofia@fakeemail.uk"

b) « […] Voici mon adresse comme convenu : mme XXXX,07,boulevard XXXX maillol.95270.montigny les cormeilles .Mon numéro de téléphone :xx-xx-xx-xx-xx. Cdt »

c) « Bonjour, J'aimerais savoir si vous commercialisez toujours le déodorant Bocage en spray parce que je ne le vois plus actuellement. Merci à XXXX »

Les données sensibles de notre corpus de courriels sont généralement : le nom, le prénom, le numéro de téléphone ou de portable, l'adresse physique ou l'e-mail, les liens vers des sites/blogs personnels ainsi que les liens vers des réseaux sociaux tels que twitter, instagram ou facebook. Ces données à anonymiser correspondent souvent aux entités nommées[3]. La tâche de reconnaissance des entités nommées (REN) est donc une tâche primordiale pour l'anonymisation.

Si, pour chaque langue, on répertorie le pourcentage d'erreurs d'anonymisation des différentes catégories, on peut établir le graphique suivant (*Figure 4*):



	Français	Anglais	Allemand
Portable / Téléphone	3,08	2,86	3,33
Url / Réseaux sociaux / mails	7,69	2,86	6,67
Adresse physique	15,38	31,43	52,86
Nom / Prénom	20,00	27,14	80,00

Figure 4 – Pourcentage d'erreur de reconnaissance d'entité pour chaque catégorie

La catégorie ayant le taux d'erreur le plus élevé est, toutes langues confondues, les noms et les prénoms avec respectivement un taux de 20% pour le français, 27% pour l'anglais et 80% pour l'allemand. Cette difficulté à anonymiser les noms et prénoms peut provenir de plusieurs facteurs :

- Un manque de règle pour l'allemand qui est une langue nouvellement traitée par l'entreprise.

[3] Les entités nommées (EN) apparaissent comme un élément essentiel dans plusieurs domaines du TAL. Elles répondent à un besoin de structurer des connaissances afin de les rendre exploitables. L'entité nommée est définie comme une unité textuelle correspondant à l'origine à des noms de personnes, de lieux et d'organisations.

- La difficulté de reconnaitre l'entité <personne>, qui se confond avec les marques (« *Helena Rubinstein* », «*Yves Saint Laurent* ») ou les noms de parfums (« *Chloé* », « *Jeanne Lanvin Couture* ») et les noms de produit de beauté.
- La typographie des noms et prénoms ; la reconnaissance d'entité des personnes se révèle difficile lorsque toutes les majuscules ne sont pas renseignées, ce qui est le cas de beaucoup de documents évalués.

La deuxième catégorie qui possède le taux d'erreur le plus élevé est la catégorie des adresses physiques avec un taux d'erreur de 15% pour le français, 31% pour l'anglais et 53% pour l'allemand. Cette dernière catégorie est difficile à anonymiser car le nombre de segment à reconnaître varie énormément selon l'adresse.

CONCLUSION

Le système actuellement mis en place éprouve des difficultés à anonymiser les adresses physiques dans le cas de l'anglais. Ces dernières posent d'importantes erreurs de frontières compte tenu de la longueur du segment à anonymiser. C'est donc suite à ce constat que le présent mémoire se concentrera uniquement sur l'amélioration de l'anonymisation des adresses physiques, dans des verbatim écrits uniquement en anglais.

1.2 PROBLEMATIQUE

SOMMAIRE

1.2.1 L'ANONYMISATION

Si le substantif « anonymisation » ou le verbe «anonymiser » n'apparaissaient pas dans les dictionnaires tels que *Le nouveau Petit Robert* en 1993, ces néologismes ont depuis su parfaitement s'intégrer dans la langue française d'aujourd'hui.

D'après le dictionnaire en ligne *Le Larousse*, le nom « anonymisation » est tout simplement « le fait d'anonymiser », c'est-à-dire « de rendre anonyme ». On remarque que ce processus ne s'applique pas à des personnes mais à des documents (exemple : « *Rendre anonyme un CV* »).

Appliquée au règlement général sur la protection des données (RGPD) qui vise à protéger les personnes concernées par un traitement de leurs données à caractère personnel, l'anonymisation des adresses physiques consiste à masquer les informations identifiant un individu, tout en laissant intacts les autres informations présentes dans le document. C'est autour de cette problématique que sera construit ce mémoire.

1.2.2 LES DIFFICULTES

FORMAT DES ADRESSES

Nous l'avons vu, les erreurs d'anonymisation peuvent être multiples, et de répercutions différentes (l'omission totale de l'anonymisation étant l'erreur la plus grave). Ces difficultés peuvent être liées au format même des adresses.

Pour rappel, ce mémoire se concentre uniquement sur les mails rédigés en anglais, suite à la première évaluation du modèle mis en place. Cependant, rien n'exclut la possibilité des consommateurs à écrire une adresse étrangère dans leur mail. Ainsi, la difficulté est de pouvoir reconnaître plusieurs types de format d'adresses postales.

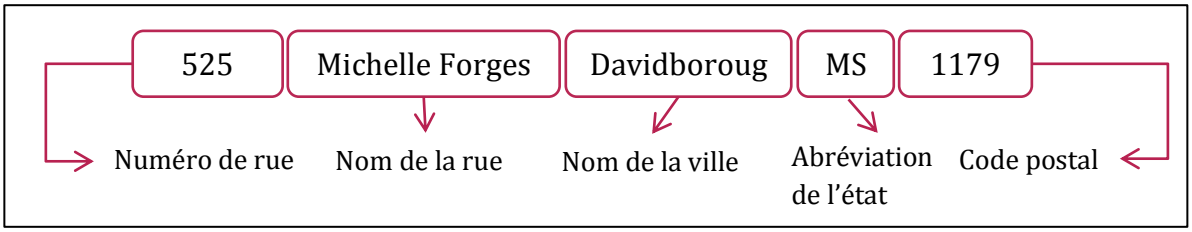


Figure 5 - Format standard d'une adresse américaine

Dans un format standardisé, les adresses américaines (*Figure 5*) comportent : un numéro de rue composé de un à quatre chiffres, puis un nom de rue qui peut être également composé d'un type (*road*, *avenue*, *street*), suivi du nom de la ville, puis de deux lettres pour l'abréviation des états, et enfin un code postal à cinq chiffres. Ce schéma d'adresse est le format d'adresse le plus simple. Il peut également y avoir d'autres informations comme le numéro d'un appartement (*exemple 1*), le nom d'un immeuble ou d'un bâtiment, ou encore, une adresse militaire (*exemple 2)* qui comporterait un numéro d'unité ainsi qu'un numéro de box. L'adresse militaire ne possède pas le même format de code postal.

(1) 025 Kimberly Oval Apt. 383, Davidborough MS 11791

(2) Unit 3104 Box 4037 DPO AA 57731-4570

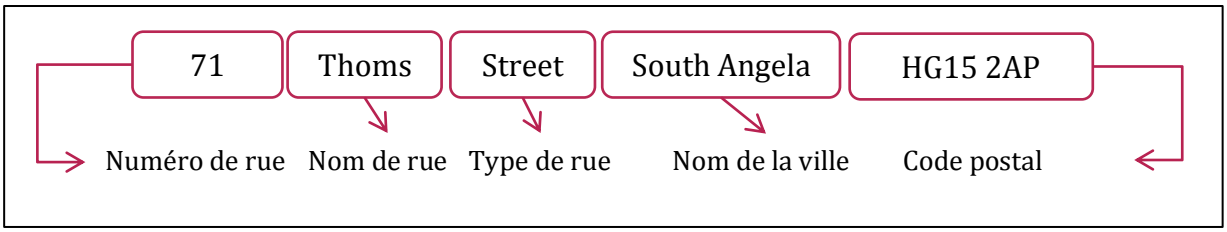


Figure 6 - Format standard d'une adresse anglaise

Les adresses anglaises standards (*Figure 6*) se constituent à peu près de la même manière que celles des adresses américaines, à ceci près que leur numéro de rue ne se compose plus que de deux chiffres et que leur code postal est constitué d'une première

partie de deux lettres et de deux chiffres, puis d'une seconde partie d'un chiffre et de deux lettres.

On retrouve également dans les verbatim anglais des adresses françaises (*Figure 7*) : le format de ce type d'adresse comporte cette fois-ci des différences dans l'ordre des éléments qui la compose. En effet, le type de rue, s'il existe, sera placé avant le nom de la rue, de même que le code postal sera placé avant le nom de la ville.

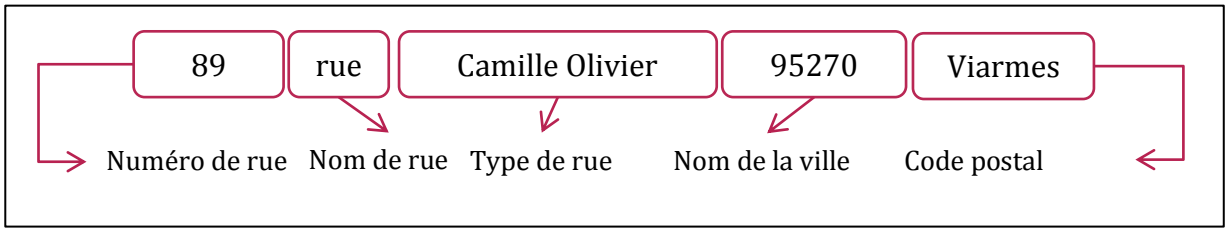


Figure 7 - Format standard d'une adresse française

Les adresses italiennes (*Figure 8*) ont, de même que les adresses américaines ou anglaises, le type de rue renseigné avant le nom de la rue. Par contre, le numéro de la rue se place cette fois-ci derrière le nom de la rue. Le nom de la ville est écrit après le code postal, comme en France. Cependant, les adresses italiennes ont la particularité de rajouter un code province, souvent mis entre parenthèses.

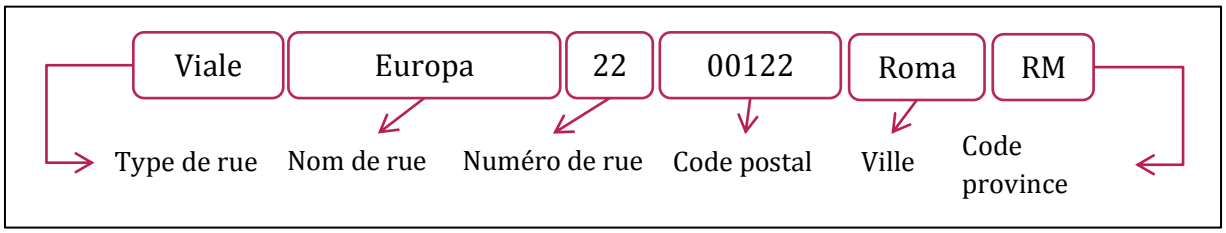


Figure 8 - Format standard d'une adresse italienne

En résumé, les différences syntaxiques peuvent s'illustrer ainsi (*Figure 9*) :

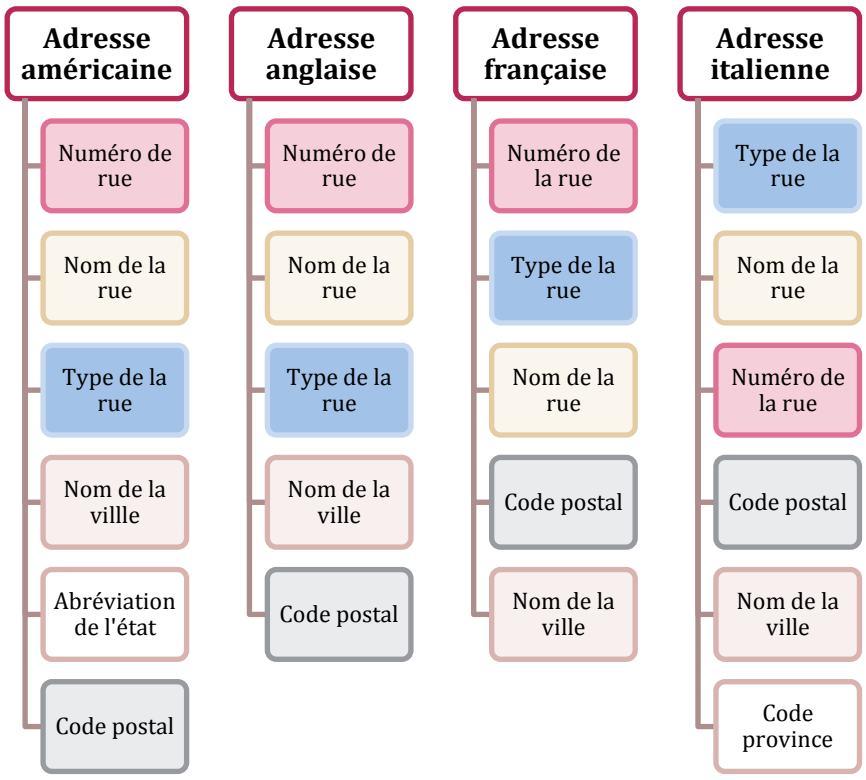


Figure 9 - Différences syntaxiques entre les adresses

Outre la possibilité d'avoir une adresse secondaire renseignée (numéro d'appartement etc.), il est également courant d'avoir dans la première partie de l'adresse postale le prénom ainsi que le nom d'une personne, ce qui peut rajouter quelques difficultés supplémentaires dans la tâche de reconnaissance des entités nommées (REN).

TYPOGRAPHIES DES ADRESSES

[Moncla et al., 2017], dans leur étude d'extraction automatique des noms des rues de Paris, soulignent plusieurs difficultés qu'ils ont pu rencontrer. Ainsi, la ponctuation, les tirets, ou même les apostrophes sont souvent sources d'erreurs liées à la typographie. Les encadrés dans les exemples suivants sont les portions reconnues par leur système.

a) *street of Mrs. Arnoux*
b) *rue de la Tour-d'Auvergne*
c) *rue Notre-Dame-des-Champs*
d) *rue de la Barrière des Gobelins*

Dans les exemples b) à d), les erreurs sont dues à la présence des mots de connexion entre deux mots en majuscules.

1.2.3 CONCLUSION

L'objectif de ce mémoire consistera à trouver parmi les méthodes (présentées Partie 2 Chapitre 2) la méthode la plus pertinente pour repérer les unités significatives d'une adresse postale, anglaise ou étrangère, dans des mails rédigés en anglais, à l'issue des expérimentations (Partie 3). Les difficultés de cette tâche seront inhérentes aux difficultés de REN : il faudra tout d'abord repérer correctement les portions des entités, délimitées par des frontières, pour ensuite attribuer la bonne catégorie aux portions précédemment repérées.

PARTIE 2

II. ETAT DE L'ART

CHAPITRE 1

2.1 TRAVAUX PRECEDENTS

SOMMAIRE

Plusieurs travaux ont été réalisés dans le cadre de l'anonymisation des données personnelles, que ce soit appliqué à un domaine médical [Grouin, 2013], ou spécifiquement pour des courriels [De Mazancourt et al., 2014]. Dans notre cas, on s'intéresse plus spécifiquement à l'anonymisation des entités nommées de localisation qui constitueraient une adresse physique.

2.1.1 GUIDE D'ANNOTATION QUAERO

Le projet Quaero possède nombre d'entités nommées structurées définies et spécifiées pour annoter des corpus de presse audio et de presse ancienne (environ trois millions de mots). La partie qui nous intéresse est la structuration choisie pour définir les entités catégorisant *les localisations*. Dans [Rosset et al., 2011], il est expliqué que l'entité <loc> couvre les localisations, les lieux et les entités spatiales. Cette catégorie regroupe six sous-catégories :

1. les localisations administratives <loc.adm>
2. les localisations géographiques <loc.phys>
3. les voies <loc.oro>
4. les bâtiments <loc.fac>
5. les adresses <loc.add>
6. et les autres localisations qui ne font pas partie des précédentes sous-catégories <loc.other>

Pour couvrir les adresses postales, plusieurs sous-catégories devront être appliquées ; la sous-catégorie <loc.add> ne se suffit pas à elle seule, puisqu'elle ne désigne dans ce projet qu'un « point » dans l'espace, par exemple un point dans une voie (rue). C'est dans <loc.oro> que sont catégorisées les rues, les places, les routes ou les

autoroutes. Cette dernière sous-catégorie se compose d'un genre <kind> qui constitue le type de voie (autoroute, rue, place) et d'un nom <name> qui regroupe le nom de ce type. Le nom peut être un nom de personne ou de ville.

La sous-catégorie <loc.add> se décompose quant à elle en deux sous-sous-catégories : les physiques <loc.add.phys> et les électroniques <loc.add.elec>. Cette dernière catégorie désigne les coordonnées électroniques que ce soit des numéros de téléphone, des urls, des adresses de messageries ou bien encore des identifiants de réseaux sociaux. Nous ne détaillerons donc pas cette sous-catégorie ici.

Pour la phrase « *J'habite au 2 rue de Lille escalier B* », les entités seront donc taguées de la manière suivante :

```
J'habite au
<loc.add.phys>
    <adress-number> 2 </adress-number>
    <loc.oro>
        <kind> rue </kind>
        de
        <name><loc.adm.town> Lille </loc.adm.town></name>
    </loc.oro>
    <other-address-component> escalier B </other-address-component>
<loc.add.phys>
```

Figure 6 – Exemple d'annotation d'une adresse suivant le projet Quaero

2.1.2 LES OUTILS

METHODE A BASE DE REGLES

Plusieurs outils ont été développés dans des buts de REN. Dans leur projet d'extraction de rues parisiennes, [Moncla et al., 2017] ont utilisé le PERDIDO NER *processing chain* (PPC). Celui-ci transforme et étiquette du texte brut via différents processus : segmenteur de phrase, tokenisation, lemmatisation, et étiqueteur morpho-syntaxiques. Pour les adresses, le PPC utilise des transducteurs à états finis[4] en cascade. Ces derniers ont été développés en utilisant les programmes CasSys disponible dans la plateforme Unitex. L'outil s'est révélé très efficace, puisque leurs résultats montrent une F_1 mesure[5] de 99,3 appliqué à leur corpus de test.

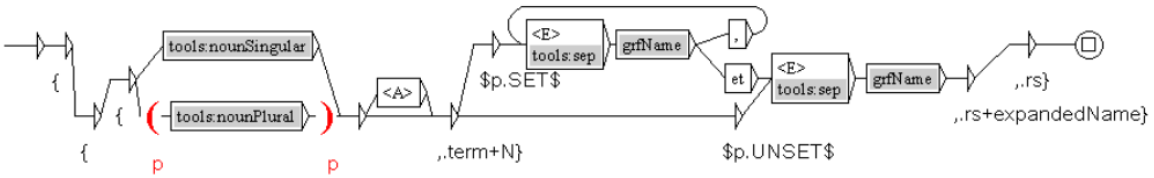


Figure 10 - Exemple de transducteur issu du projet [Moncla et al., 2017]

[4] Un transducteur à états finis est un automate fini avec remplacements
[5] Voire Partie 2 Chapitre 3 – Les mesures d'évaluation

De même que pour PERDIDO, CasEN utilise des graphes de transducteurs pour la reconnaissance d'entité nommée. La version Quaero de CaseEN, utilisée lors de la campagne ETAPE[6] a été classé premier sur la tâche de REN.

Dans [Grouin, 2013], plusieurs outils à base de règles ont été spécifiquement créés pour répondre à la problématique d'anonymisation de documents cliniques : «Stomato », « De-ID » et « Medina ». Ces deux premiers outils ont respectivement eu une F mesure de 0,85 et de 0,87 sur leur corpus d'application.

METHODE STATISTIQUE

Il existe plusieurs méthodes dites statistiques qui répondent à la résolution d'une tâche de REN. Les **arbres de décisions**, appelés ainsi par analogie avec un arbre naturel pour ses notions de nœuds et de branches qui correspondraient respectivement aux décisions et aux différents chemins pour parvenir aux feuilles (phases finales), ont été utilisés par exemple par [Palouras et al., 2017] pour repérer plus précisément les organisations et les personnes.

Les **séparateurs à vaste marge** (SVM), introduit par [Vapnik, 1998] calculent l'hyperplan qui sépare le mieux un espace en classes. C'est la méthode choisie par [Guo et al., 2006] pour leur système d'anonymisation. Plutôt qu'une tâche de REN, les chercheurs ont réalisé à l'issu de leur recherche une tâche de classification entre les éléments à anonymiser ou non.

Enfin, les **champs aléatoires conditionnels** (CRF) [Lafferty et al., 2001] sont largement employés dans ce domaine de recherche. Il existe de nombreux outils utilisant les CRF, comme Wapiti (2009) ou bien CRFSuite [Okazaki et al., 2013]. C'est ce dernier outil qui sera utilisé dans les expérimentations pour la méthode statistique.

[6] Evaluations in Automatic Speech Processing

CHAPITRE 2

2.2 METHODOLOGIES

SOMMAIRE

2.2.1 L'APPROCHE SYMBOLIQUE

INTRODUCTION

L'approche symbolique, ou approche à base de règles, est une méthode coûteuse à implémenter en termes de temps. En effet, elle est souvent construite manuellement [Han et al., 2004]. De plus, il faut non seulement connaître le domaine d'expertise mais aussi étendre de nouvelles règles tout en faisant une maintenance des anciennes. Cependant, c'est une méthode qui peut offrir de très bons résultats au niveau qualitatif car les règles sont établies par rapport aux documents possédés. De plus, cette méthode ne nécessite pas obligatoirement de corpus annoté. Elle permet notamment d'avoir le contrôle sur ce que font les règles et de pouvoir les organiser, par exemple par modules.

Néanmoins, ce type d'approche ne permet pas une bonne réutilisation de règles appliquées à de nouveaux genres de documents. En effet, les règles sont trop souvent spécifiques, et ne permettent donc pas l'application à un autre domaine plus général. D'après [Grouin, 2013], le fonctionnement d'un système symbolique repose sur deux éléments : des ressources externes et des règles.

Pour les ressources externes, il s'agit essentiellement de liste de termes (par exemple des listes de noms, prénoms, de villes ou de régions), des listes de déclencheurs ou bien des dictionnaires. Il peut aussi s'agir d'outils externes qui ajouteraient des

informations supplémentaires au corpus (étiqueteur morpho-syntaxique, analyseur syntaxique).

Pour les règles, il s'agit principalement d'expressions régulières (transducteurs). Ces dernières constituent un système puissant et rapide pour effectuer des recherches dans des chaines de caractères. Elles sont particulièrement efficaces pour traiter des cas numériques (par exemple des numéros de rues ou des codes postaux) car il est simple et rapide d'en modéliser toutes les formes. En revanche, lorsqu'il n'y a pas de prétraitements linguistiques (segmentation, lemmatisation), il est bien plus difficile de les utiliser. Elles devront être verbeuses, et donc plus sujettes à des erreurs de bornes.

LES ETAPES PRINCIPALES DE L'APPROCHE SYMBOLIQUE

Selon [Meystre et al., 2010], l'approche symbolique s'effectue en deux étapes. La première étape consisterait à appliquer des patrons syntaxiques implémentés sous la forme de regex, et des déclencheurs. On peut également établir des ordres de priorités pour l'exécution de certaines règles, pour accorder une plus grande importance aux règles les plus larges au détriment des règles plus spécifiques. La deuxième étape projette quant à elle des listes (de noms, prénoms ou lieux, selon l'usage). Ces listes peuvent également être utilisées par des patrons.

2.2.2 L'APPROCHE STATISTIQUE

INTRODUCTION

Dans le cadre d'une anonymisation automatique, les méthodes à base d'apprentissages statistiques peuvent proposer deux types d'approches [Grouin, 2013] : une première qui consisterait à repérer une entité nommée avec ses bornes bien délimitées, et une seconde, qui consisterait à classifier chaque token parmi une ou plusieurs catégories déjà définies au préalable. Autrement dit, soit une tâche de segmentation, soit une tâche de classification.

L'approche statistique est utilisée pour plusieurs raisons. La principale est inhérente à la définition que l'on prête à l'apprentissage automatique, -champ d'étude de l'intelligence artificielle-, qui permet d'apprendre des modèles à partir d'exemples. Ces modèles, guidés par les données (exemples d'entrées et de sorties attendues), doivent être capables d'ajuster des paramètres numériques en fonction de ce qu'ils ont déjà appris, pour ne prendre que les décisions les plus probables ou vraisemblables sur de nouvelles données [Nouvel et al., 2015]. Dans notre cas d'anonymisation d'adresses postales, cette approche permettrait de découvrir automatiquement les combinaisons les plus optimales des nombreux paramètres associés aux indices repérés dans les textes. L'avantage de cette méthode consiste donc à fournir des caractéristiques au modèle

pour laisser le soin à l'outil de déterminer les plus pertinentes selon les données fournies.

Autre avantage, cette approche peut traiter rapidement un grand nombre de données pour une efficacité reconnue pour la tâche de REN [McCallum et Li, 2003]. Cependant, pour une méthode supervisée, cela implique de préparer un corpus annoté qui contienne, pour chaque classe à repérer, d'assez nombreuses variations, afin que le modèle soit le plus robuste possible. Cette approche a donc un coût conséquent.

Il existe dans les approches statistiques deux types de modèles : *(i)* les modèles génératifs, et *(ii)* les modèles discriminants. Parmi ces derniers se trouve les champs markoviens conditionnels (ou CRF de chaînes linéaires) introduit par [Lafferty et al., 2001], et souvent utilisés dans le domaine du TAL, en particulier pour des tâches d'étiquetage. C'est sur la base de ce modèle que les expériences sur l'approche statistique de la deuxième partie seront réalisées. Ce modèle est dit discriminant car il modélise directement la probabilité conditionnelle $P(y|x)$ qui permet la prédiction (c'est à dire le choix du label ou étiquette à utiliser selon le token donné). Les modèles génératifs, comme les réseaux bayésiens ou les modèles de Markov Cachés (HMM), quant à eux, ne modélise pas directement la probabilité $P(y|x)$ puisque qu'ils doivent calculer également la probabilité d'un vecteur caractéristique $P(x)$.

Modèle génératif	*Modèle discriminant*
$P(y\|x) = \frac{P(y\|x)}{P(x)}$	$P(y\|x)$

Tableau 3 - Formalismes des modèles

FORMALISATION D'UNE SEQUENCE D'ETIQUETAGE

Soit la séquence de token suivante : "*I live at 2 rue de Lille*". Cette phrase de sept tokens est la réalisation de $x = (x_1, \dots, x_7)$ d'un champ de sept variables aléatoires $X = (X_1, \dots, X_7)$. On appelle ces variables X_i des variables d'observation [Gaussier et Yvon, 2011] ou *features*. Prenons pour l'étiquetage $y = (y_1, \dots, y_7)$ comme réalisation de variables aléatoires $Y = (Y_1, \dots, Y_7)$. La phrase ci-dessus pourrait ainsi être modélisée de la façon suivante (*Tableau 8*) :

x	I	live	at	2	rue	de	Lille
	↓	↓	↓	↓	↓	↓	↓
y	O	O	O	U-NUM_P	U-TYPE_P	B-NAME_P	L-NAME_P

Tableau 4 - Modélisation d'une séquence d'étiquetage

Soit la probabilité conditionnelle $P(Y = y\,|X = x)$. On suppose que P appartient à une classe qui résoudra deux types de problèmes : *(i)* l'identification de P à partir des

exemples étiquetés pour entrainer le modèle et *(ii)* la détermination des étiquettes (ici O, U-NUM_P, U-TYPE_P, B-NAME_P et L-NAME_P) à associer avec une nouvelle séquence de variables d'observation.

Si l'on applique cette modélisation aux CRF (champs markoviens conditionnelles linéaires), la séquence ("*I live at 2 rue de Lille*") sera représentée ainsi (*Figure 11*) :

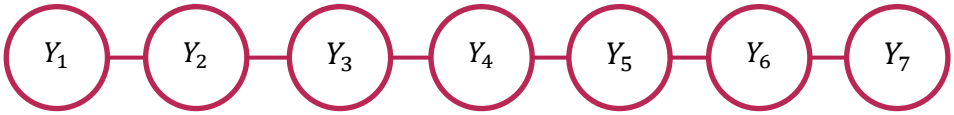


Figure 11 – Classes Y d'un CRF

Les caractéristiques de la valeur d'une annotation d'un token peuvent être retrouvées pour l'annotation du token suivant ou précédant (puisque le graphe est non-dirigé). Les CRF prennent donc en compte le contexte pour étiqueter une séquence de token.

La formule qui modélise les CRF linéaires d'ordre 1 est donnée par [Gaussier et Yvon, 2011] :

$$P(Y = y|X = x) = \frac{1}{Z(x)} \exp\left(\sum_{i=2}^{n} \sum_{k=1}^{K} \theta_k \, f_k(y_{i-1}, y_i, x, i) \right)$$

Avec :

$$Z(x) = \sum_{y \in Y} \exp\left(\sum_{i=2}^{n} \sum_{k=1}^{K} \theta \, f_k(y_{i-1} y_i, x, i) \right)$$

Équation 1 - Formule des CRF

y_{i-1} et y_i sont les arguments des fonctions, n la somme de la taille des variables d'observations, K la somme de toutes les fonctions de caractéristiques, θ le poids pour une fonction de caractéristique donnée, f les fonctions des caractéristiques et enfin Y la somme de tous les labels possibles.

LES CARACTERISTIQUES (FEATURES)

On appelle "caractéristique" (*feature* en anglais) toute information associée à un token. Généralement, pour un token donné, plusieurs caractéristiques sont attribuées. Selon [Grouin, 2013] il existe trois grandes familles de caractéristiques :

1. Les **caractéristiques de surface** comme la casse du token, la présence de ponctuation, la longueur du mot, la présence de caractère spéciaux, la présence de nombres

2. Les **caractéristiques profondes** ou caractéristiques morpho-syntaxiques, syntaxiques ou sémantiques

3. Les **caractéristiques externes** comme la position du token dans le document ou la fréquence globale du token dans le document

Bien sûr, cette liste n'est pas exhaustive ; d'autres caractéristiques peuvent être envisagées. Par exemple, [Tkachenko et Simanovsky, 2012] utilise comme caractéristique la forme des tokens[7], les préfixes ou encore les suffixes.

LA CREATION DU MODELE D'APPRENTISSAGE

Nous utilisons comme outil d'apprentissage *CRFsuite*, qui est une implémentation des CRF développé par [Okazaki et al., 2013]. Cet outil prend en entrée deux types de fichiers : *(i)* un premier qui comporte toute la configuration du modèle avec le choix de l'algorithme (L-BFGS, OWL-QN, SGD, Perceptron, AROW), le choix des caractéristiques, le nombre d'itérations ou encore le choix des coefficients de pénalité et *(ii)* un second fichier, qui est le corpus d'apprentissage.

Dans notre cas, nous voulons anonymiser les adresses postales contenues dans les mails des consommateurs. Ce corpus annoté comportera donc un ensemble de mails (documents), dont chaque token sera *(i)* segmenté, *(ii)* caractérisé par une étiquette morphosyntaxique, et enfin *(iii)* étiqueté manuellement selon nos règles d'annotations.

En python, cela correspond donc à structurer le corpus comme un dictionnaire (ensemble de documents) de dictionnaires (documents) de tuple de trois valeurs (le token, l'étiquette morphosyntaxque et l'étiquette attendue de l'entité nommée).

```
data =
[
     [('I', 'PRON', 'O'),
     ('live', 'VERB', 'O'),
     ('at', 'ADP', 'O'),
     ('2', 'NUM', 'U-NUM_P'),
     ('rue', 'NOUN', 'U-TYPE_P'),
     ('de', 'ADP', 'B-NAME_P'),
     ('Lille', 'PROPN', 'L-NAME_P')
     ]
]
```

Exemple de structuration des données sous Python

Ce choix de structuration s'inspire du format utilisé lors de la campagne d'évaluation CoNLL de 2003 [Jong Kim Sange et De Meulder, 2003]. Dans cette compagne, les données sont formatées en fichiers tabulaires de quatre colonnes : une colonne pour le mot, une autre pour sa partie du discours, une encore pour son *chunk tag* [8]et enfin une dernière pour son étiquette d'entité nommée. Devant l'entité est spécifié son schéma

[7] La caractéristique de forme des tokens (*shape feature*) est le résultat de mappages tels que Bill → Xxxx, Moscow-based → Xxxxxx-xxxxx

[8] On désigne par chunk la plus petite séquence d'unités linguistiques possible.

BIO, introduit par [Ramshaw et Marcus, 1995]. Ce schéma d'annotation indique les frontières de l'entité nommée : B(egin) pour le début d'une entité, I(n) pour renseigner que l'entité est à l'intérieur de la portion annotée, et enfin O(ut) pour signifier que le token courant ne fait pas partie d'une entité.

Il n'y a pas d'indication de fin d'entité nommée avec ce schéma d'annotation, au contraire du format BILOU, plus précis et récent [Ratinov et Roth, 2009]. Ce dernier possède, en plus du sigle BIO, le L de *last* pour indiquer la fin d'une entité et le U de *unit* pour indiquer que l'entité nommé se compose d'un seul token isolé. Ce dernier format sera utilisé lors de nos expérimentations.

2.3 L'EVALUATION

SOMMAIRE

2.3.1 LES MESURES D'EVALUATION

Évaluer un système de REN consiste à comparer une référence qui est produite manuellement par des experts humains à une hypothèse qui est produite automatiquement par un système. Cela consiste donc à mesurer la distance entre le résultat attendu et le résultat obtenu. Pour cela, on utilise en général les mesures classiques que sont le rappel, la précision et la F-mesure. Il s'agit d'indicateurs compris entre 0 et 1 pour en faciliter l'interprétation, toujours largement employés dans les évaluations en TAL. Le *Slot Error Rate* (SER) peut venir compléter ces mesures.

La matrice de confusion adaptée au Traitement Automatique des Langues est la suivante (*Tableau 12*) :

		REFERENCE	
		Etiqueté	**Non étiqueté**
HYPOTHESE	**Etiqueté**	Vrais positifs	Faux positifs
	Non étiqueté	Faux négatifs	Vrais négatifs

Tableau 12 - Matrice de confusion

Pour expliciter les termes de "vrais positifs" et de "faux négatifs", on peut dire que ces premiers correspondent à des éléments étiquetés de la même manière dans l'hypothèse et la référence, et que ces seconds, les faux négatifs, se réfèrent à des

éléments qui sont étiquetés dans la référence mais qui sont absents de l'hypothèse.

Les faux positifs se rapportent à des éléments qui, étiquetés dans l'hypothèse, ne le sont pas dans la référence. Les vrais négatifs correspondent à l'ensemble des éléments qui sont absents de l'hypothèse et de la référence.

LE RAPPEL

Le rappel est donné par le ratio entre le nombre de réponses correctement étiqueté par le système (les vrais positifs) et le nombre de réponses attendues, donc contenues dans la référence (les vrais positifs et les faux négatifs). C'est une mesure de quantité qui permet d'estimer la capacité d'un système à couvrir l'ensemble des réponses se trouvant dans un corpus de test. Un mauvais rappel correspond à du silence, cas où des réponses pertinentes ne sont pas proposées par le système alors qu'elles existent (faux négatifs).

$$R = \frac{vrais\ positifs}{vrais\ positifs + faux\ négatifs}$$

Équation 2 - Formule du rappel

LA PRECISION

La précision, quant à elle, est donnée par le ratio entre le nombre de réponses correctes (vrais positifs) et toutes les réponses données par un système (vrais et faux positifs). C'est une mesure de qualité qui permet d'évaluer la fiabilité des réponses fournies par le système. Une mauvaise précision correspond à du bruit, autrement dit, des réponses non-pertinentes qui seraient présentes dans l'hypothèse (faux positifs).

$$P = \frac{vrais\ positifs}{vrais\ positifs + faux\ positifs}$$

Équation 3 - Formule de la précision

LA F-MESURE

La F-mesure représente la moyenne harmonique pondérée du rappel et de la précision.

$$F - mesure = \frac{(1 + \beta^2)\ \times P \times R}{\beta^2 \times P + R}$$

Équation 4 - Formule de la F-mesure

β est une valeur qui permet soit d'équilibrer le rappel et la précision (β = 1), soit de mettre en avant l'une ou l'autre des mesures. Si le rappel est privilégié, on aura $\beta > 1$

et inversement, si la précision est considérée comme plus importante, on donnera comme valeur $\beta < 1$.

Dans le cas d'une anonymisation, il n'est finalement guère important qu'une étiquette soit mal attribuée, du moment qu'elle en a tout de même une. De plus, on peut accepter dans une certaine mesure que certains éléments soient étiquetés alors qu'ils ne l'auraient pas dû, du moment que tous les éléments à anonymiser soient bel et bien repérés. Ainsi, on peut tolérer du bruit (faux positifs), mais guère le silence (faux négatifs). Nous avons choisi d'accorder plus d'importance au rappel avec un β à 2 ($\beta = 2$). On utilisera donc dans nos évaluations la F2-mesure.

$$F_2 - mesure = \frac{(1 + 2^2) \times P \times R}{2^2 \times P + R}$$

Équation 5 - Formule de la F2-mesure

SLOT ERROR RATE

Pour prendre en compte dans l'évaluation les erreurs de frontières et de typage, on peut compléter les mesures vues précédemment par le SER. Cette dernière mesure repose sur une énumération de différentes erreurs :

1. Les délétions "D" : le nombre d'entités qui figurent dans la référence mais qui ont été manquées par le système (faux négatifs)
2. Les insertions "I" : le nombre d'entités détectées dans l'hypothèse mais qui ne figurent pas dans la référence (faux positifs)
3. Les erreurs de typages seules "T" : le nombre d'entités détectées dans l'hypothèse avec des frontières correctes mais avec une catégorie incorrecte
4. Les erreurs de frontières "F" : le nombre d'entités détectées dans l'hypothèse avec une catégorie correcte mais avec des frontières incorrectes
5. Les erreurs de type et de frontières "TF" : le nombre d'entités détectées dans l'hypothèse avec une catégorie et des frontières incorrectes

Plus le résultat est proche de zéro, plus le système est performant.

2.3.2 L'EVALUATION HUMAINE

L'EVALUATION PAR DES HUMAINS EN ANONYMISATION

L'évaluation humaine, appliquée lors d'une tâche d'anonymisation, peut être plus tolérante que les mesures d'évaluations classiques, générées automatiquement. Dans les tâches d'anonymisation, la règle impose de masquer tous les éléments qui permettraient d'identifier un individu. Si les mesures d'évaluations présentées dans la première partie n'acceptaient pas l'anonymisation partielle des données, l'humain pourrait, quant à lui,

décider arbitrairement que les quelques informations laissées dans le document ne sont de toute façon pas suffisantes pour remonter à l'individu. Il pourrait ainsi évaluer différemment les mesures classiques d'évaluation.

Evalué comme inacceptable par les mesures d'évaluations classiques	Evalué comme acceptable par les mesures d'évaluations humaines
[…] My adress is : XXXX, Washington DC *[…]Thank you, XXXX AB51-XXXX*	

Tableau 5 - Exemple de différence d'évaluation entre la machine et l'humain

Dans ce mémoire, les résultats ne seront toutefois pas évalués par l'humain mais bien par les mesures d'évaluations classiques, car les décisions peuvent parfois être complexes.

PARTIE 3

III. EXPERIMENTATIONS

CHAPITRE 1

3.1 LE CORPUS

SOMMAIRE

3.1.1 PARTICULARITE DU CORPUS

Nos données sont des messages de consommateurs s'adressant au service client d'enseignes de produit de beauté, rédigés en anglais. Les messages, fournis par les enseignes, nous parviennent donc directement dans nos bases de données. Cependant, seuls les messages comportant des adresses physiques nous intéressent ici. C'est donc à l'aide de requête SQL sur la base de données que notre corpus sera constitué.

En raison de leur faible représentation, seuls 100 mails comportant des adresses physiques ont été récupérés directement. A ces données sont rajoutés des adresses créées artificiellement à l'aide de scripts[9] sur la base de ces mails et d'autres ressources externes afin d'étoffer le corpus. Ces 400 nouvelles adresses sont des adresses anglaises, américaines, françaises ou italiennes. Elles sont insérées dans des verbatim variant en longueur de 20 à 300 tokens, créées à l'aide des chaines de Markov sur le modèle des mails issus des clients. Le corpus d'apprentissage se constitue ainsi de 500 mails : 100 provenant de la base de données du client, et 400 créés artificiellement.

Verbatim récupéré (l'adresse a été modifiée pour qu'elle puisse apparaître ici)	"Dear Sirs/Madam. I am 60 and I have medium/sensitive skin. Would you be able to send me some free samples to try out if they suit my skin so I can make the right choice. Thank you for your help. 2 Walson Way, Green Hill, Danson, Telford, RT5 6TT"
Verbatim généré à l'aide de script	"I buy flower bomb perfume, body lotion, but i want to try a Si parfume. Can I have sample at James Levine Acceso de Daniela Mendoza 56 Alicante, 17465."

Tableau 6 - Exemple de verbatim

[9] Voire Annexe – create_data_en.py

3.1.2 LES PRETRAITEMENTS NECESSAIRES

Avant tout traitement, les messages ont été normalisés : les sauts de lignes ou certaines balises html ont été supprimés. Ensuite, afin de préparer les corpus d'entrainement et de test pour l'approche statistique, chaque token de chaque verbatim a été segmenté puis disposé dans des listes renseignant : sa forme, son étiquette morphosyntaxique, et son étiquette d'entité nommé.

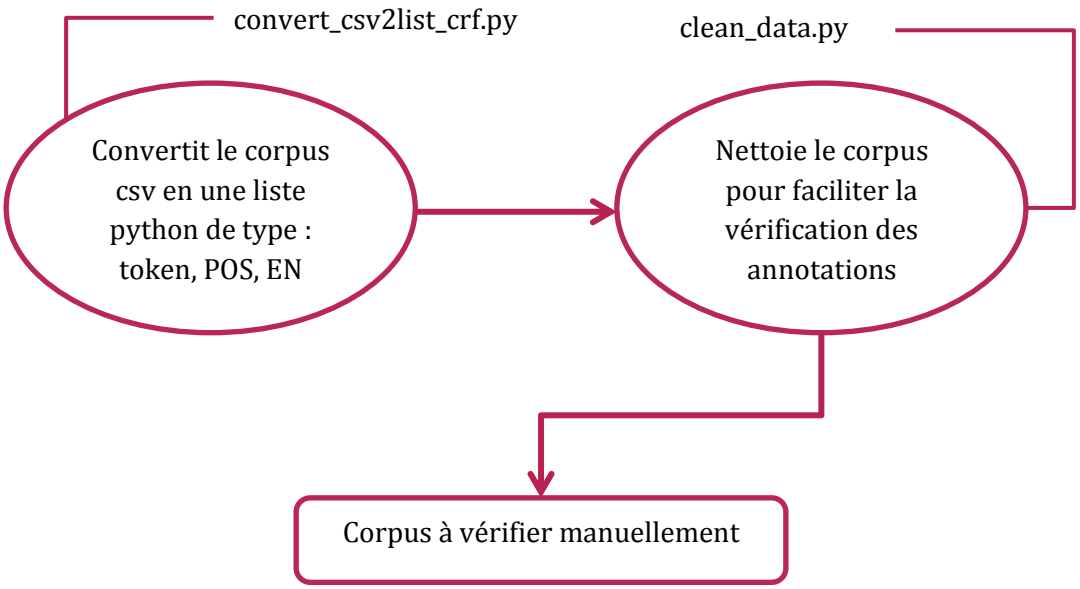


Figure 13 - Chaines de prétraitements pour l'approche statistique

Seuls les verbatim générés aléatoirement peuvent prétendre à une génération des étiquettes d'entité nommés issus du guide d'annotation. Mais même pour ces derniers, une vérification manuelle est nécessaire pour corriger les nombreuses erreurs générées. De plus, pour éviter l'apprentissage sur des adresses trop monotones du fait de leur format respectif irréprochable, de nombreuses modifications ont été appliquées afin de les rendre plus intéressantes :

- des ajouts d'abréviation (« rd » pour « road », « st » pour « street », etc.)
- des modifications de casse
- des ajouts ou suppression de ponctuations
- des suppressions de certaines portions de l'adresse

3.1.3 LES REGLES D'ANNOTATION

Plusieurs catégories et sous-catégories ont été créées pour expliciter une adresse physique :

Nom de la catégorie	Nom de la sous-catégorie	Explications
<NUM>	<NUM_P>	Numéro de rue de l'adresse principale
	<NUM_S>	Numéro de rue de l'adresse secondaire
<NAME>	<NAME_P>	Nom de la rue de l'adresse principale
	<NAME_S>	Nom de la rue de l'adresse secondaire
<TYPE>	<TYPE_P>	Type de la rue (ex : *street*, *road*, *avenue*) de l'adresse principale
	<TYPE_S>	Type de la rue de l'adresse secondaire (ex : *flat*, *building*)
<STATE_ABR>	-	Abréviation de l'état pour les adresses américaines
<DIR>	-	Direction (ex : *South*, *North*) pour les adresses anglaises
<CITY>	-	Nom de la ville
<ZIP>	-	Code postale ou code zip
<PERSON>	-	Nom et prénom devant l'adresse

Tableau 7 - Les différentes catégories du guide d'annotation

On peut retrouver dans les adresses secondaires, si elles sont renseignées, un type, un numéro ainsi qu'un nom. La catégorie <PERSON> ne peut être utilisée dans les noms de rue, bien que certains noms de rues désignent des personnes. Cette catégorie est utilisée dans le cas où les consommateurs renseignent leur patronyme avant l'adresse.

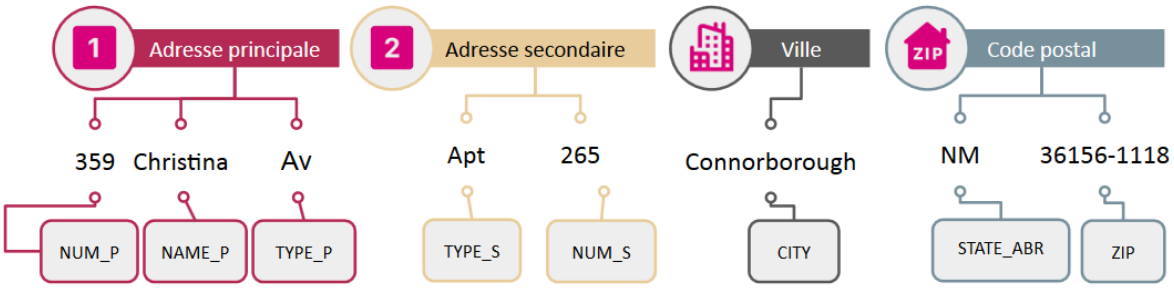


Figure 14 - Exemple d'adresse annotée

Le format BILOU a également été accolé dans un premier temps aux catégories, ce qui nous donne un total de 26 entités différentes pour l'annotation.

CHAPITRE 2

3.2 EXPERIENCES ET RESULTATS

SOMMAIRE

3.2.1 METHODE SYMBOLIQUE

Toutes les expressions régulières destinées à anonymiser les verbatim sont rédigées dans des fichiers de texte brut, un fichier pour chaque langue. Lors du processus d'anonymisation, les programmes Java chargeront ces fichiers en appliquant ces règles une à une, dans leur ordre d'apparition.

LES REGLES

REGLES CREEES A PARTIR DU FORMAT EPISTOLAIRE

La forme épistolaire répond à certaines règles qui peuvent nous être utiles dans notre tâche d'anonymisation. En effet, on retrouve parfois dans les e-mails un respect des codes précis de la lettre : les formules d'appel, les formules finales, la phrase d'introduction et enfin la signature, qui peut comporter une adresse postale. Comme la mise en page des mails n'existe plus avec la normalisation, les expressions régulières qui vont récupérer les signatures s'aident des formules finales telles que « *yours faithfully* », « *best* », « *kind regards* », etc.

\b([Ff]aithfully)\b([\.,]).*$

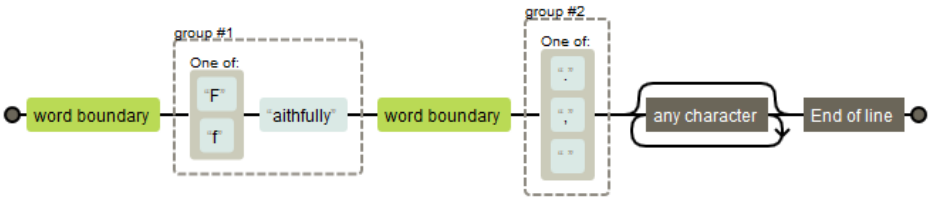


Tableau 8 - Exemple de règle autour de l'expression « faithfully »

REGLES CREEES AUTOUR DU TYPE DE RUE OU DU CODE POSTAL D'UNE ADRESSE

Que ce soit pour le français, l'anglais ou l'italien, les types de rue tels que « road » ou « street », « rue » ou « avenue », « via » ou « strada », sont généralement un bon moyen de récupérer une adresse.

Type d'adresse	Regex
anglaise ou américaine	([09]{1,4}?[\p{L}]+?([\p{L}])?)?(([09]{1,4})? ?([\p{L}]+ ?)?)?([09]{1,3}?([Bb](is)?|[Tt](er)?|BIS|TER)?,?)?([AZ])?(\b(apartment|apt)\b|\b(boulevard|blv?d)\b|\b(avenue|ave)\b|\b(building|bldg)\b|\bcenters?\b|\bcircles?\b|\bhill\b|\b(court|ct)\b|\b(drive)\b|\beast\b|\b(exp(resway)?y\b)|\b([Ee]xt(ension)?)\b|\bf(ort)?t\b|\bf(ree)?w(a)?y\b|\bh(igh)?w(a)?y\b|\bisland\b|\bj(un)?ct(ion)?\b|\bl(a)?n(e)?\b|\bnorth(east|west)\b|\bparkwaypky\b|\bpl\b|\broad\b|\br(ural)?d(elivery)?\b|\br(ural)?r(oute)?\b|\bs(ain)t?\b|\b(sw)\b|\bsq(uares?)?\b|\bst(reet)?\b|\bs(ui)?te\b|\bter(race)?\b|\bt(urn)?p(i)?ke\b|\bw(est)?\b|\bhedge\b) ?(([\p{L}'-,]+|[0-9]{0,2}) ?){1,10},?

Tableau 9 - Exemple de regex créée autour du type

Le code postal peut également servir de noyau pour tenter récupérer le contexte gauche et droit d'une adresse.

REGLES CREEES AUTOUR DE MOTS DECLENCHEURS

D'autres règles s'aident de mots déclencheurs tels que « *my following address is* », « *my address:* », « *my contacts details are*» pour récupérer et masquer de possibles coordonnées. Cependant, avec les expressions régulières, la frontière d'une adresse est difficilement repérable avec ce genre de règle. C'est pourquoi ces règles doivent être souvent combinées à d'autres, chargées de récupérer des portions d'adresse qui auraient été à moitié anonymisées.

Au total, **62 règles** ont été créées spécifiquement pour les adresses (*Tableau 10*).

	Règles liées au format épistolaire	Règles se basant sur le type	Règles se basant sur le code postal	Règles se basant sur les déclencheurs	Règles corrigeant les adresses à moitié anonymisées
Nombre	12	7	10	14	19

Tableau 10 - Répartition du nombre de régle

EVALUATION

Toutes les règles ont été conçues à partir d'exemples existants du corpus extraits directement à partir de la base de données. Pour l'évaluation, les règles ont été testées sur 100 autres verbatim inconnus comportant des adresses physiques. Les tableaux de résultats sont les suivants (*Tableau 11*) :

Vrais positifs	Faux positifs	Faux négatifs	Rappel	Précision	F-mesure
56	22	44	56%	72%	63%

Tableau 11 - Evaluation globale de la méthode symbolique

3.2.2 METHODE STATISTIQUE

La deuxième méthode choisie pour nos expériences d'anonymisation repose exclusivement sur le formalisme des champs aléatoires conditionnels (CRF) de chaîne linéaire présenté au chapitre 2 de la partie 2.

CONFIGURATIONS

Parmi les nombreux outils, *CRFSuite* a été sélectionné. Le choix de cet outil a été dicté par sa simplicité d'utilisation d'une part, et par sa gestion des paramètres et des fichiers à traiter d'une seconde part.

Couplé avec *Sklearn*, il est possible de déterminer l'optimisation des paramètres. Parmi les algorithmes L-BFGS, L2SGD, AP, PA ou bien encore AROW, nous avons retenu L-BFGS. [Shanno et al., 1970].

La pénalité $l1$ qui est utilisée comme paramètre de sélection des caractéristiques est celle par défaut, laissée à 0. Le coefficient pour la $l2$ est à 0,01.

Le corpus d'entrainement et de développement est construit à partir des 500 verbatim. La séparation entre les deux corpus a été réalisée avec la fonction *train_test_split* de *Scikit Learn*. Cette fonction divise par défaut le corpus en un corpus d'entrainement et de test avec 75% des verbatim pour le premier et 25% des autres verbatim pour le second.

EXPERIMENTATION

Les caractéristiques que nous avons fournies à *CRFSuite* afin de construire le modèle sont diverses. Il y a tout d'abord les **caractéristiques de surface** qui renseignent les propriétés morphologiques du token avec la présence ou non :

- de casse typographique (on distingue si le token commence par une majuscule, s'il est en capital ou s'il est en minuscule)

- de ponctuation
- de chiffre
- sa taille
- son code
- sa terminaison (les trois ou les deux dernières lettres)

Ces caractéristiques ne requièrent pas l'utilisation de ressources ni d'outils externes. La casse typographique est généralement utilisée pour distinguer les noms propres des autres mots. La ponctuation est, elle, un bon moyen de repérer un séparateur. De plus, savoir si le token étudié est un chiffre ou non, permet de reconnaître les numéros de rues ou de codes postaux. Quant à la taille des tokens, on utilise généralement cette caractéristique pour distinguer les mots outils des entités. Le code est ici une fonction créée pour convertir la chaine de caractères composant le token par des correspondances codées : le *a* pour un caractère en minuscule, le *A* pour la majuscule, le *p* s'il s'agit d'une ponctuation, et enfin le *c* s'il s'agit d'un chiffre. Si une même typologie de caractère est trouvée, elle n'est pas répétée.

Token	Token codé
TNR75	Ac
Road	Aa
,	p

Tableau 12 - Exemple de token codé

Les **caractéristiques profondes données** sont ici l'étiquette en partie du discours qui correspond au token (la POS). Celle-ci avait déjà était renseignée durant les prétraitements, à l'aide de SpaCy. L'utilisation des parties du discours permet de distinguer les catégories faisant rarement l'objet d'une anonymisation (pronom, adjectif, prépositions), des catégories beaucoup plus susceptibles d'en faire l'objet (noms propres).

Enfin, les **caractéristiques externes** utilisées dans les expériences sont essentiellement des lexiques. La présence de chaque token est ainsi vérifiée dans des listes qui comportent :

- les noms de types de rue tels que « *road* » ou « *street* »
- des expressions telles que «*my following address* », « *my address* »

Nous donnons dans la figure suivante un exemple de la mise en forme des caractéristiques dans notre programme Python.

```
# Common features for all words

features = [
    'bias',
    'word.lower=' + word.lower(),
    'word[-3:]=' + word[-3:],
    'word[-2:]=' + word[-2:],
    'word.isupper={}'.format(word.isupper()),
    'word.istitle={}'.format(word.istitle()),
```

```
        'word.isdigit={}'.format(word.isdigit()),
        'word.ispunct={}'.format(punct),
        'postag=' + postag,
        'len=' + str(len(word)),
        'partofstreet={}'.format(word in street),
        'partofd={}'.format(word in declencheur),
        'code={}'.format(code(word))
]
```

Code – Les caractéristiques communes à tous les mots

RESULTATS

Dans cette partie est présentée les résultats obtenus par les CRF sur le même corpus de test que la méthode symbolique.

Vrais positifs	Faux positifs	Faux négatifs	Rappel	Précision	F1-mesure	F2-mesure	D	I	T	F	TF
175	57	124	59%	75%	66%	41%	124	57	16	7	217

Tableau 13 - Evaluation globale de la méthode statistique

Lorsque l'on s'intéresse à la reconnaissance des différentes catégories trouvées par l'hypothèse par rapport à la référence, on remarque que le modèle a eu ici du mal à reconnaitre les codes postaux (ZIP), ainsi que les noms des villes (CITY).

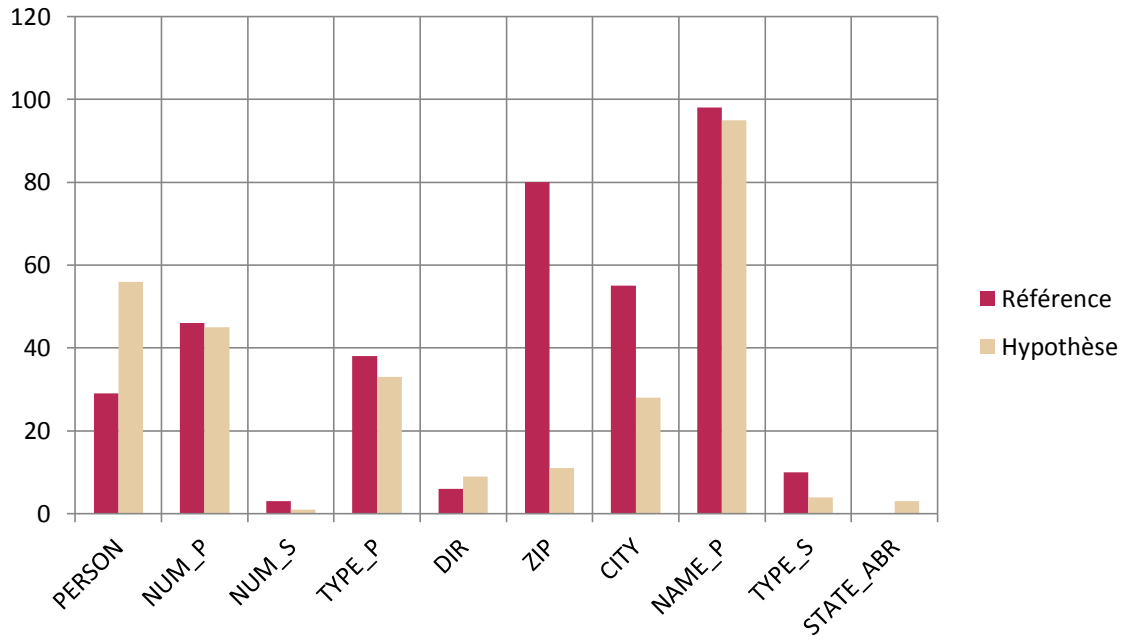


Figure 15 - Fréquence des catégories reconnues dans l'hypothèse et la référence

Cette difficulté peut s'expliquer par le fait que la segmentation des codes postaux des adresses anglaises a été différente du modèle d'apprentissage (présence d'un espace qui a segmenté le token en deux).

Le tableau des cinq meilleures transitions ainsi que des cinq moins bonnes sont données par le tableau suivant (*Tableau 14*) :

Meilleures transitions (poids)		Pires transitions (poids)	
O -> O	(7.880848)	B-NAME_P -> U-TYPE_P	(-0.362100)
B-NAME_P -> L-NAME_P	(7.767755)	L-NAME_P -> U-NUM_P	(-0.595233)
B-TYPE_P -> L-TYPE_P	(6.406943)	U-TYPE_S -> B-ZIP	(-0.609393)
B-CITY -> L-CITY	(6.393796)	U-NAME_P -> U-NUM_S	(-0.619890)
B-ZIP -> L-ZIP	(5.779750)	B-CITY -> O	(-0.675794)

Tableau 14 - Les meilleures et pires transitions

L'étiquette « O » signifie « *Other* », et concerne donc tous les tokens qui ne font pas partie d'une adresse. La meilleure transition est le passage d'une étiquette « O » à une autre étiquette « O », avec un poids de 7,88. Les autres transitions ayant un poids élevé sont les noms, les types, les villes et les codes postaux. Chacune de ces transitions montre le passage d'un premier segment de l'entité (B) vers le dernier segment de cette même entité (L). On remarque que si l'évaluation du système peinait à reconnaitre les villes et les codes postaux, ces derniers ont cependant un poids élevé, ce qui contribue à augmenter la précision de ces catégories.

La pire transition est le passage hypothétique d'un premier segment (B) d'un nom de rue (NAME_P) avec un type de rue (TYPE_P) en segment unique (U). Cette transition est reconnue comme n'étant pas bonne et pour cause, <B-NAME_P> doit toujours être suivie de <I-NAME_P> ou de <L-NAME_P>.

3.2.3 COMPARAISON DES METHODES

S'il l'on compare les résultats entre la méthode symbolique et la méthode statistique, les différences ne sont pas flagrantes, mais c'est tout de même la méthode statistique qui obtient sur toutes les mesures les meilleures performances.

	Rappel	Précision	F-mesure
Méthode symbolique	56%	72%	63%
Méthode statistique	59%	75%	66%

Tableau 15 - Comparaison de la performance des méthodes

Le corpus de test, strictement le même pour les deux méthodes afin de pouvoir effectuer les comparaisons, comportait des schémas d'adresse jusqu'alors inconnus des deux méthodes : des adresses australiennes et coréennes. Ces dernières n'ont été reconnues par aucun des deux systèmes, tandis que les adresses australiennes ont été partiellement anonymisées par la méthode statistique et totalement par la méthode symbolique (avec une règle portant sur une formulation finale).

Nous remarquons que beaucoup d'adresses restent partiellement anonymisées, ce qui explique le faible rappel dans les deux méthodes.

CHAPITRE 3

3.3 CONCLUSION

A l'issu des expériences, les méthodes symboliques et statistiques ont montré des performances quasiment similaires. La précision est plus élevée que le rappel pour les deux systèmes, ce qui n'est pas très satisfaisant pour une tâche d'anonymisation. Le modèle symbolique n'a su anonymiser de nouvelles adresses physiques uniquement grâce à ses règles portant sur le format épistolaire, et non grâce à ses règles portant sur l'adresse, tandis que le système statistique a su partiellement s'adapter. Nous pouvons en conclure que le système statistique d'anonymisation d'adresses s'exporterait beaucoup plus facilement que le système symbolique sur un autre projet avec un format de documents autre que le courriel, du moment que l'on dispose d'un corpus annoté.

Il reste beaucoup de marge de progression pour aboutir à un système d'anonymisation correct. En effet, le système symbolique gagnerait en efficacité avec plus de règles prenant en compte des adresses étrangères, et le système statistique s'améliorait grandement avec un corpus d'entrainement plus fourni. La plus grande difficulté pour l'anonymisation des adresses postales est la reconnaissance de tous ses éléments : comme l'entité nommée est trop longue, elle est trop souvent partiellement reconnue.

Nous avons remarqué que pour beaucoup de catégories, la première caractéristique retenue par les CRF était souvent d'ordre morphosyntaxique (caractéristique sur les parties du discours). Le système statistique gagnerait peut-être donc à voir ses caractéristiques profondes plus détaillées et plus nombreuses. Autre perspective d'amélioration : l'utilisation d'une méthode hybride, combinant les meilleurs aspects des deux méthodes.

BIBLIOGRAPHIE

[Feten Ben Fredj, 2017] Feten Ben Fredj (2017). *Méthode et outil d'anonymisation des données sensibles*, Cryptographie et sécurité. Conservatoire national des arts et métiers - CNAM

[De Mazancourt et al., 2014] De Mazancourt, H., Couillault, A., Recourcé, G. (2014). *L'anonymisation, pierre d'achoppement pour le traitement automatique des couriels*. Journée d'Etude ATALA Ethique et TAL, Paris

[Gaussier et Yvon, 2011] Gaussier, E. et Yvon, F. (2011). *Modèles statistiques pour l'accès à l'information textuelle*. Lavoisier

[Grouin, 2013] Grouin, C. (2013). *Anonymisation de documents cliniques : performances et limites des méthodes symboliques et par apprentissage statistique*. Université Pierre et Marie Curie – Paris VI

[Guo et al., 2006] Guo, Y. Gaizauskas, R., Roberts, I., Demetriou, G., et Hepple, M. (2006). *Identifying personal health information using support vector machines. In Proc of i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, Washington, DC

[Han et al., 2004] Tzong-Han, T., Shih-Hung, W., Cheng-Wei, L., Cheng-Wei, S., et Wen-Lian, H. (2004). *Mencius : A Chinise Named Entity Recognizer Using Maximum Entropy-based hybrid Model. In Computational Linguistics and Chinese Language Processing*, Vol. 9, No. 1

[Jong Kim Sange et De Meulder, 2003] Jong Kim Sange et De Meulder (2003). *Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition*, Proceedings of CoNLL-2003. Edmonton, Canada

[Lafferty et al., 2001] Lafferty, J., McCallum, A., et Pareira, F. (2001). *Conditional Random Fields : Probabilistic models for segmenting and labeling sequence data*. In *Proc of ICML*.

[McCallum et Li, 2003] McCallum, A., Li W. (2003). *Early Result for Named Entity Recognition with Conditional Random Fields*, Feature Induction and Web-Enhanced Lexicons, In *CONLL'03*

[Meystre et al., 2010] Meystre, S.M., Friedlin, F. J., South, B. R., Shen, S., et Samore M. H. (2010*). Automatic de-identification of textual documents in the electronic health record: a review of recent research*. BMC Med Res Methodol, 10(70)

[Moncla et al., 2017] Moncla L., Gaio M., Joliveau T, Le Lay Y-F. (2017*). Automated Geoparsing of Paris Street Names in 19th Century Novels*, Redondo Beach, CA, United States

[Moncla et al., 2018] Moncla, L., Gaio, M., Egorova, E., Claramunt, C. (2018). *An automatic extraction method of static and dynamic spatial contexts from texts*, <hal-01694376>

[Nouvel et al., 2015] Nouvel, D., Ehrmann, M., Rosset, S. (2015). *Les entités nommées pour le traitement automatique des langues*, ISTE editions.

[Okazaki et al., 2013] Cho, H.C., Okazaki, N., Miwa, M., et Tsujii J. (2013). *Named entity recognition with multiple segment representations*, In *Information Processing & Management*

[Palouras et al., 2017] Paliouras G., Kaekaletsis, V., Petasis, G., Spyropoulos C.S. (2017) *Learning Decision Trees for Named-Entity Recognition and Classification*

[Ramshaw et Marcus, 1995] Ramshaw, A., Marcus, M.P. (1995). *Text Chunking using Transformation-Based Learning*

[Ratinov et Roth, 2009] Ratinov, L., Roth, D. (2009). *Design Challenges and Misconceptions in Named Entity Recognition, Urbana*, IL 61801, USA

[RÈGLEMENT (UE) 2016/679] Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données) (Texte présentant de l'intérêt pour l'EEE)

[Rosset et al., 2011] Rosset, S., Grouin, C., Zweigenbaum, P. (2011*). Entités nommées structurées : guide d'annotation Quaero*, Quaero, T3.2, presse écrite et orale, version 1.25

[Shanno et al., 1970] Shanno, David, F., Kettler, Paul C. (1970). *Optimal conditioning of quasi-Newton methods*, Mathematics of Computation, 24 (111)

[Tkachenko et Simanovsky, 2012] Tkachenko, M. et Simanovsky, A. (2012). *Named entity recognition : Exploring features,* In *KONVENS*

[Vapnik, 1998] Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

ANNEXES

EXTRAITS DE CODE

CREATE_DATA_EN.PY

```
#!/usr/bin/env python
# Chloe Lecointe

# Modules a installer :
# -- pip install Faker
# -- https://github.com/joke2k/faker
# -- pip install markovify
# -- https://github.com/jsvine/markovify

import random
import markovify
import plac
from faker import Faker

def main():
    """
    Create random corpus of addresses (us, uk, it, de, es, nl, pt and fr)
    """
[…]

#+-+-+-+-+-+-+-+-+-+-+-+-+-
# Generator Address
#+-+-+-+-+-+-+-+-+-+-+-+-+-

     us = Faker()
     it = Faker('it_IT')
     uk = Faker('en_GB')
     fr = Faker('fr_FR')
     us_address = [(us.address()) for i in range(50)]
     it_address = [(it.address()) for i in range(50)]
     uk_address = [(uk.address()) for i in range(50)]
     fr_address = [(fr.address()) for i in range(50)]
     us_address = [add.replace('\n', ' ') for add in us_address]
     it_address = [add.replace('\n', ' ') for add in it_address]
     uk_address = [add.replace('\n', ' ') for add in uk_address]
     fr_address = [add.replace('\n', ' ') for add in fr_address]
     address_all = us_address + it_address + de_address + uk_address +
     es_address + nl_address + pt_address + fr_address

#+-+-+-+-+-+-+-+-+-+-+-+-+-
# Generator Verbatim
#+-+-+-+-+-+-+-+-+-+-+-+-+-

# Get raw text as string.
     with open("../data/verbatim_exemple.txt") as f:
         text = f.read()

# Build the model.
     text_model = markovify.Text(text, state_size=2)
```

```
# Print three randomly-generated sentences
     verbatim = []
     for i in range(400):
         verbatim.append(text_model.make_sentence())
[…]

#+-+-+-+-+-+-+-+-+-+-+-+-+-
# Generator All
#+-+-+-+-+-+-+-+-+-+-+-+-+-

     with open("../data/generated_data_en.csv", "w") as f:

         f.write('verbatim\n')
         for verbatim, rand_d, name_all, address_all in zip(verbatim,
     rand_d, name_all, address_all):
             f.write("\"{} {}{} {}.\"\n\n".format(verbatim, rand_d,
     name_all, address_all))

if __name__ == '__main__':
     plac.call(main)
```

TRAINING_CRF.PY

```
#!/usr/bin/env python
# Chloe Lecointe
# usage : python crf_training.py crf_all.model
# doc: https://python-crfsuite.readthedocs.io/en/0.4/pycrfsuite.html
# https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html

import plac
import nltk
import re
import pycrfsuite
import numpy as np
from corpus_crf_cleaned_hyp3 import data
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from collections import Counter
from dictionary import street
from dictionary import declencheur

def word2features(doc, i):

    """
    doc = list of tuple
    i = index
    """
     word = doc[i][0]
     postag = doc[i][1]
     punct = [".", ",", "!", "-", "_", "(", ")", ";", ":", "!", "?"]

def code(word):

    """
    Replace character caracteristic by a code
    """
     rsl = ""

     for w in word:
```

```
        if w.isupper():
            rsl += "A"
        if w.islower():
            rsl += "a"
        if w in punct:
            rsl += "p"
        if w.isdigit():
            rsl += "c"
        for i in rsl:
            dup = i + i
            rsl = re.sub(dup, i, rsl)
        return rsl


# Common features for all words
features = [

        'bias',
        'word.lower=' + word.lower(),
        'word[-3:]=' + word[-3:],
        'word[-2:]=' + word[-2:],
        'word.isupper={}'.format(word.isupper()),
        'word.istitle={}'.format(word.istitle()),
        'word.isdigit={}'.format(word.isdigit()),
        'word.ispunct={}'.format(punct),
        'postag=' + postag,
        'len=' + str(len(word)),
        'partofstreet={}'.format(word in street),
        'partofd={}'.format(word in declencheur),
        'code={}'.format(code(word))
        ]

# Features for words that are not at the beginning of a document

if i > 0:
        word1 = doc[i-1][0]
        postag1 = doc[i-1][1]
        features.extend([
            '-1:word.lower=' + word1.lower(),
            'word.isupper={}'.format(word.isupper()),
            'word.istitle={}'.format(word.istitle()),
            'word.isdigit={}'.format(word.isdigit()),
            'word.ispunct={}'.format(punct),
            '-1:postag=' + postag1,
            'len=' + str(len(word)),
            'partofstreet={}'.format(word in street),
            'partofd={}'.format(word in declencheur),
            'code={}'.format(code(word))
])

else:

# Indicate that it is the 'beginning of a document'
features.append('BOS')

# Features for words that are not at the end of a document

if i < len(doc)-1:
        word1 = doc[i+1][0]
        postag1 = doc[i+1][1]
```

```
        features.extend([

        '+1:word.lower=' + word1.lower(),
        'word.isupper={}'.format(word.isupper()),
        'word.istitle={}'.format(word.istitle()),
        'word.isdigit={}'.format(word.isdigit()),
        'word.ispunct={}'.format(punct),
        '+1:postag=' + postag1,
        'len=' + str(len(word)),
        'partofstreet={}'.format(word in street),
        'partofd={}'.format(word in declencheur),
        'code={}'.format(code(word))
])

        else:
        # Indicate that it is the 'end of a document'
        features.append('EOS')
        return features

def extract_features(doc):
        return [word2features(doc, i) for i in range(len(doc))]

def get_labels(doc):
        return [label for (token, postag, label) in doc]

def print_transitions(trans_features):
        for (label_from, label_to), weight in trans_features:
        print("%-6s -> %-7s %0.6f" % (label_from, label_to, weight))

def print_state_features(state_features):
        for (attr, label), weight in state_features:
        print("%0.6f %-6s %s" % (weight, label, attr))

@plac.annotations(

model=("Name of the crf model\n Exemple : crf_en.model"))

def main(model):
        """
        Generate features / model training

        """
        # X_train: feature (ex: '-1:word.istitle=False')
        # y_train: tag (ex: B-ZIP)

        X = [extract_features(doc) for doc in data]
        y = [get_labels(doc) for doc in data]
        X_train, X_test, y_train, y_test = train_test_split(X, y,
        test_size=0.1)
        trainer = pycrfsuite.Trainer(verbose=True)

        # Submit training data to the trainer
        for xseq, yseq in zip(X_train, y_train):
                trainer.append(xseq, yseq)

        # Set the parameters of the model

        trainer.set_params({

        # coefficient for L1 penalty
```

```
    'c1': 0.1,
    # coefficient for L2 penalty

    'c2': 0.01,

    # maximum number of iterations

    'max_iterations': 200,

    # whether to include transitions that
    # are possible, but not observed

    'feature.possible_transitions': True,

    })

    # MODEL
    trainer.train(model)

    # EVALUATE
    tagger = pycrfsuite.Tagger()
    tagger.open(model)
    info = tagger.info()
    y_pred = [tagger.tag(xseq) for xseq in X_test]

    # random sample in the testing set

    i = 1
    for x, y in zip(y_pred[i], [x[1].split("=")[1] for x in X_test[i]]):
         print("{} ({})".format(y, x))

    # Create a mapping of labels to indices

    labels = {"U-NUM_P":0,"B-NAME_P":1,"L-NAME_P": 2,"I-NAME_P":3,"B-
    TYPE_S":4,"L-TYPE_S":5,"U-NUM_S":6,"U-CITY":7,"U-STATE_ABR":8,"U-
    ZIP":9,"B-ZIP":10,"I-ZIP":11,"L-ZIP":12,"U-TYPE_P":13,"U-DIR":14,"B-
    CITY":15,"L-CITY":16,"U-NAME_P":17,"U-TYPE_S":18,"O":19,  "B-PERSON":20,
    "L-PERSON":21, "U-PERSON":22, "I-CITY":23, "I-PERSON":24, "B-
    TYPE_P":25, "L-TYPE_P":26}

    # Convert the sequences of tags into a 1-dimensional array

    predictions = np.array([labels[tag] for row in y_pred for tag in row])
    truths = np.array([labels[tag] for row in y_test for tag in row])

    # Print out the classification report

    print(classification_report(
    truths, predictions,
    target_names=["U-NUM_P", "B-NAME_P", "L-NAME_P", "B-TYPE_S", "L-
    TYPE_S", "U-NUM_S", "U-CITY", "U-STATE_ABR", "U-ZIP", "B-ZIP", "I-ZIP",
    "L-ZIP", "U-TYPE_P","B-TYPE_P", "L-TYPE_P", "U-DIR", "B-CITY", "L-
    CITY","I-CITY", "U-NAME_P", "U-TYPE_S", "O", "I-NAME_P", "B-PERSON",
    "L-PERSON", "U-PERSON", "I-PERSON"]))

    #+-+-+-+-+-+-+-+-+-+-+-+-+-
    # Corpus info visualisation
    #+-+-+-+-+-+-+-+-+-+-+-+-+-

    print("\n---TRAIN---")
    print("Nombre de verbatim train :{}\n".format(len(X_train)))
```

```
print("\n---TEST---")
print("Nombre de verbatim test :{}\n".format(len(X_test)))
#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
# Features info visualisation
#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

print("Les meilleures transitions :")
print_transitions(Counter(info.transitions).most_common(15))
print("\nLes pires transitions :")
print_transitions(Counter(info.transitions).most_common()[-15:])
print("\nLes meilleures features :")
print_state_features(Counter(info.state_features).most_common(20))
print("\nLes pires features :")
print_state_features(Counter(info.state_features).most_common()[-20:])

#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
# Param info visualisation
#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

# Print the current params
print("\n{}\n{}".format("Params:",trainer.get_params()))
if __name__ == '__main__':
    plac.call(main)
```

INDEX