Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Dé-identification cohérente de l'ensemble des documents cliniques d'un patient

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Elise BIGEARD

Directeur de mémoire :

Natalia Grabar

Encadrant :

Frantz Thiessard

Année universitaire 2014/2015

Table des matières

Liste des figures

Liste des tableaux

Remerciements

Je tiens à remercier mes encadrants Natalia Grabar et Frantz Thiessard pour leur aide, leur présence et leur travail pour porter ce travail à terme. Je remercie également Vianney Jouhet pour sa présence complémentaire, Sylvain Tessier pour son travail d'annotation, Cyril Grouin pour son aide avec Medina, toute l'équipe du PAD pour leur accueil et leur présence. Enfin, merci à Michaël et Chloé pour leur soutien et leur aide face à mon code récalcitrant.

INTRODUCTION

Présentation générale

La création puis l'alimentation d'un dossier médical est obligatoire pour tout patient, que la visite du patient soit faite dans le cadre d'une consultation ou d'une hospitalisation. Ce dossier doit contenir la description du motif de venue, les éléments principaux qui ont conduit au diagnostic et au traitement éventuel puis à la surveillance du retour à un état normal. Depuis quelques années, les dossiers passent progressivement d'un format papier à un dossier patient informatisé (DPI). Dans ces derniers, une grande majorité des informations stockées le sont sous forme non structurée ou semi structurée. L'utilisation de données nominatives est une obligation dans la pratique du soin, de la surveillance. Cependant, les données médicales peuvent être utilisées dans d'autres cadres : la recherche clinique, l'enseignement, les tests logiciels effectués par les industriels etc. Dans ces cas, ces données médicales brutes ne peuvent pas être divulguées, sous peine de menacer la vie privée des patients mentionnés dans ces données. Ces dernières doivent impérativement ne plus être identifiantes.

Le travail de dé-identification entre alors en jeu. La dé-identification est un problème que l'on retrouve dans d'autres domaines dès lors qu'il s'agit de documents comportant des données nominatives et que l'on désire ré-utiliser ces données dans un cadre différent de celui initialement prévu. Dans le cas du domaine médical, le poids du secret médical rend la tâche de dé-identification critique et obligatoire. Ainsi, il est impossible de constituer un corpus sans avoir vérifié manuellement que toute information identifiante ait été retirée de chaque document du corpus. Cette contrainte rend les ressources rares pour le développement d'applications TAL. Le développement d'outils de dé-identification automatique pourrait aider à résoudre une partie du problème [1]. Actuellement, le domaine du TAL médical est en manque de données et d'outils comparé à d'autres domaines d'applications du TAL, comme par exemple l'informatique, l'électricité, la biologie ou le traitement des génomes a connu de grandes avancées. [Zweigenbaum, 2008] [Yandell and Majoros, 2002] Nous proposons de nous intéresser aux documents du domaine médical.

Données identifiantes

Pour anonymiser un document médical, il est tout d'abord nécessaire de définir ce que sont des données identifiantes. Les initiatives menées aux Etats-Unis [2] proposent une

1. Nous ne parlerons dans ce travail que de données directement ou quasi-directement identifiantes. Concernant les données de santé, la CNIL est encore plus sévère en considérant certaines données comme indirectement identifiantes par croisements par exemple, et certains type de données ne peuvent pas être présentes de façon conjointes.

2. http://www.ucdmc.ucdavis.edu/compliance/guidance/privacy/deident.html

liste officielle de données identifiantes : noms, prénoms, lieux et adresses plus précis qu'un état, dates plus précises qu'une année, âges supérieurs à 90 ans, numéros de téléphone et de fax, adresses mail, divers numéros d'identification tels numéro de sécurité sociale, de permis de conduire, numéro de série d'appareils implantés ; url, ip, identifiants biométriques, photographies du visage ; toute mention d'une caractéristique telle que tatouage ou cicatrice.

Le masquage (la modification, la déidentification ou l'anonymisation) de ces unités est la condition nécessaire et suffisante à la publication et l'utilisation de données aux Etats-Unis. Notons qu'une telle liste n'existe pas en France, où un accord avec la CNIL doit être effectué pour chaque étude individuelle.

Adopter la liste américaine permet d'anonymiser et de traiter les documents isolés. En revanche, lorsque l'on associe plusieurs documents relatifs à un même patient, le croisement des informations fournies par ces différents documents peut permettre de recréer l'identification de ce patient. Prenons l'exemple d'une situation rare : le cancer du sein chez l'homme. Nous sommes face au dossier anonymisé d'un patient masculin atteint du cancer du sein, indiquant l'année de son diagnostic initial. Estimons qu'environ 49000 cancers du sein soient décelés chaque année en France [Khamis et al., 2005]. Moins d'1% des cancers du sein se déclarent chez l'homme. Notre patient anonyme se situe dans une population d'environ 490 individus : il n'est pas identifié. En revanche, si l'on connaît également son groupe sanguin et son rhésus, et qu'ils s'avèrent être B- ou AB- (1% de la population chacun) notre patient se trouve dans une population totale de moins de 5 individus : il n'est plus anonyme.

D'autres situations dangereuses peuvent apparaître.

Par exemple, connaître la date exacte d'un événement (hospitalisation, examen, etc) peut être identifiant, et plus encore si on a accès à un ensemble de dates. Nous verrons aussi dans l'état de l'art que des outils existants ont pris le parti de décaler toute date d'un nombre fixe de jours, de façon à conserver les intervalles de temps, nécessaires pour l'analyse médicale. Cependant, ces intervalles peuvent également se révéler identifiantes.

Pour illustrer cette situation, la direction du système d'information du CHU de Bordeaux au sein de laquelle ce stage a été effectué, a extrait toutes les dates d'hospitalisation pour la totalité des patients ayant été hospitalisés entre début 2012 et fin 2014. Tous les délais entre deux hospitalisations ont été rattachés aux patients correspondants. Puis les suites d'intervalles identiques ont été regroupées, de façon à pouvoir connaître la "rareté" d'une suite. Puis ont été retenues uniquement les suites correspondant à au maximum X patients (suites identifiantes), divisées par le nombre de suites totales. Ceci nous donne la probabilité qu'une suite quelconque soit identifiante. La table 0.1 donne la probabilité de détecter un patient unique, ou un groupe de 2 ou 3 patients, si l'on connaît respectivement 1 seul, 2, 3, 4 ou 5 intervalles.

TABLE 0.1 – Patients identifiés par intervalles

	1 intervalle	2 intervalles	3 intervalles	4 intervalles	5 intervalles
1 patient	0.02%	0.92%	3.80%	7.31%	10.04%
1 à 2 patients	0.02%	1.31 %	4.69%	8.52%	11.22%
1 à 3 patients	0.03%	1.58%	5.22%	9.21%	11.88%

Anonymiser un dossier patient plutôt que des documents isolés présente donc deux difficultés qui ne sont pas rencontrées lors de l'anonymisation d'un document unique : Les caractères identifiants fictifs qui remplacent les traits d'identification réels, doivent être identiques même si un nouveau document concernant le même patient est traité plus tard ; le problème des intervalles entre deux évènements qui peuvent être identifiants.

Masquer toutes les informations susceptibles d'être identifiantes n'est pas une solution non plus : selon les cas et la nature de la recherche nécessitant ces données, supprimer ou modifier une information particulière peut rendre les documents inutilisables. Il apparaît complexe de concevoir une méthode de dé-identification qui supprime la totalité des informations identifiantes sans risquer de rendre le document inutilisable dans certaines situations. Ceci explique que la dé-identification soit une étape nécessaire mais en général pas suffisante.

Objectif

L'objectif de notre méthode est de fournir des dossiers patient composés de plusieurs documents, où les données identifiantes sont remplacées de façon cohérente tout au long du dossier médical complet du patient. Ainsi, le nom d'une personne en particulier sera toujours changé vers le même nom, les dates seront modifiées de façon à conserver les intervalles de temps entre deux événements, etc.

Nous désirons contrer un attaquant en possession d'un corpus anonymisé par notre méthode, cherchant à y retrouver les documents d'une personne en particulier, connaissant ses informations personnelles triviales telle la date de naissance, et ayant éventuellement accès au code de notre méthode.

Les informations considérées dans notre travail sont : prénom et nom du patient et des soignants (y compris les initiales isolées), adresse postale, numéro de téléphone, adresse e-mail, âge (si supérieur ou égal à 90 ans) et divers numéros d'identification dont notamment le NIP.

Un corpus produit avec cette méthode permettrait de pouvoir observer l'évolution d'un patient et de ses maladies, d'avoir un regard global sur l'historique d'un patient particulier, toujours sans pouvoir l'identifier. Un tel corpus pourrait par exemple servir à alimenter une base de pharmaco-vigilante, ou d'études de cas.

Tout au long de ce mémoire, les exemples qui semblent nominatifs sont bien sûr modifiés ou fictifs afin de préserver l'anonymat des personnes.

CHAPITRE 1

ÉTAT DE L'ART

Sommaire

1.1 Reconnaissance des entités nommées

Avant de pouvoir modifier de façon appropriée les informations identifiantes d'un document, il faut d'abord pouvoir les trouver. La reconnaissance des entités nommées est un problème bien connu du TAL et largement étudié. L'intérêt pour ce sujet commence dans les années 1990 avec les premiers travaux sur la reconnaissance des noms propres [Coates Stephens, 1992] Le terme *entités nommées* apparaît en 1995 durant la campagne MUC-6 [1995], en même temps qu'une première typologie des entités nommées : les ENAMEX portent sur les noms de personnes, des TIMEX désignent des expressions temporelles et des NUM les expressions numériques. La description et le traitement TAL des entités nommées n'a depuis lors cessé de croître, s'appliquant aux organisations, aux entités géographiques, aux noms de produits, jusqu'à quitter le domaine du langage pour être utilisée dans l'analyse du génome [Yandell and Majoros, 2002].

1.2 Dé-identification

Nous présentons ici quelques outils proposés dans l'état de l'art pour effectuer la dé-identification de documents cliniques.

Le logiciel DE-ID (DE-IDentification) [Neamatullah et al., 2008], anonymiseur de documents médicaux pour l'anglais, utilise un système de règles, à base de dictionnaires et d'expressions rationnelles. Il obtient un rappel de 0.967 et une précision de 0.749.

MeDS (MEdical Deidentification System) [Friedlin and McDonald, 2008] est également un outil destiné à l'Anglais basé sur des dictionnaires et des expressions rationnelles, des triggers [1] pour détecter les noms de personne, ainsi que des patrons pour détecter dates, adresses et valeurs numériques. Cet outil utilise de plus une section formatée des documents traités pour en extraire certaines données (nom du patient...) à anonymiser. 98.45% des éléments à anonymiser ont été correctement traités par MeDS lors de son évaluation.

1. déclencheurs

Medina [Grouin et al., 2009] est un outil conçu pour le français, basé lui aussi sur un système de règles, combinant dictionnaires, expressions rationnelles et patrons. De plus, cet outil décale chaque date d'un nombre de jours fixes, de façon à conserver les intervalles de temps, nécessaires pour l'analyse clinique. Il traite chaque document de façon individuelle.

Il n'existe pas à notre connaissance de méthode spécifiquement conçue pour traiter des ensembles de documents, c'est à dire tentant de dé-identifier de façon robuste et constante des documents dissociés, tout en prenant en compte le risque de ré-identification accru. Ainsi nous n'avons pas rencontré de méthode introduisant de "flou" dans le décalage des dates.

1.3 Reconnaissance des dates

Des modules existent pour reconnaître une date en texte libre. Bien que ces modules soient uniquement disponibles pour l'anglais, il n'est pas difficile d'utiliser des pré-traitements pour traduire ce qui est nécessaire, comme les noms de mois en toutes lettres. Nous utilisons le parseur du module dateutil [2] pour python, qui autorise la précédence du jour sur le mois. Ce parseur est capable d'interpréter une date en dehors d'un format précis, issue d'un texte libre. Cependant, il n'est pas conçu pour trouver des dates au sein de texte libre, et ne sait pas différencier une date d'une autre information numérique. Ainsi, "le 22 11 03 et le 11 06 03" sera interprété comme "le 22 11 2003 à 11h 6m 3sc"

L'outil de dé-identification Medina a pris le parti d'utiliser un système d'expressions rationnelles pour la détection des dates. Plusieurs patrons sont employés en cascade afin de couvrir les formats variés de date que l'on peut trouver en texte libre. Par exemple, le jour du mois peut être donné en chiffres, avec ou sans 0 initial, entièrement en lettres, ou sous forme abrégée tel *1er*.

2. https://labix.org/python-dateutil

CHAPITRE 2

RESSOURCES

Sommaire

Nous présentons ici les ressources que nous utilisons dans notre travail.

2.1 Corpus

Le corpus utilisé pendant le développement et l'évaluation de notre méthode provient du corpus de travail du projet ANR RAVEL [1]. Les documents utilisés pour ce travail sont restreints aux courriers provenant des services rhumatologie et cancérologie du CHU de Bordeaux (les sources de données structurées telles que les résultats de biologie ne contenant pas d'élément identifiant n'ont pas été utilisés, ni les formulaires semi-structurés du DPI dont les informations étaient moins riches que les courriers pour les deux services sélectionnés). Ces documents rattachés à un patient donné grâce au service patient qui centralise toutes les informations personnelles et identifiantes des patients. Notre méthode a été développée sur un échantillon de ce corpus comprenant 186 documents de cancérologie non annotés. A ces documents s'ajoutent 104 documents de rhumatologie annotés par un humain, utilisés en fin de développement pour corriger et améliorer la méthode. Le corpus d'évaluation est constitué de 138 documents de rhumatologie produits par le meme annotateur.

Pour les besoins du projet ANR, les documents ont été extraits du DPI au format CDA-R2 : chaque document CDA est un document XML contenant des métadonnées structurées concernant un courrier, ainsi que ledit courrier encapsulé dans le document XML. Le courrier en lui-même est fourni dans un format de traitement de texte type Microsoft Word. Les métadonnées contiennent des informations sur le document lui-même (date d'édition) sur le patient (nom, adresse, sexe, date de naissance...) et sur le personnel soignant impliqué (nom).

Ces courriers sont généralement courts, dépassant rarement 5000 caractères. Il ne s'agit pas de texte brut : les documents comportent très souvent une en-tête et un pied de page comportant des informations très identifiantes pour le patient et pour

1. Projet ANR TECSAN 2011 N ANR-11-TECS-012, Recherche et Visualisation des informations dans le dossier patient électronique [Thiessard et al., 2012]

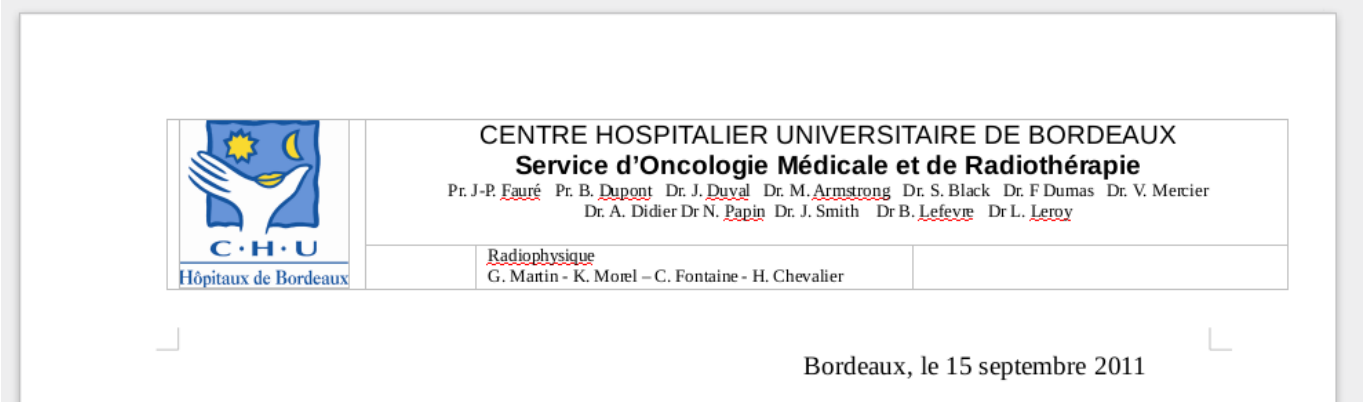


FIGURE 2.1 – En-tête d'un courrier comprenant les informations de nombreux médecins

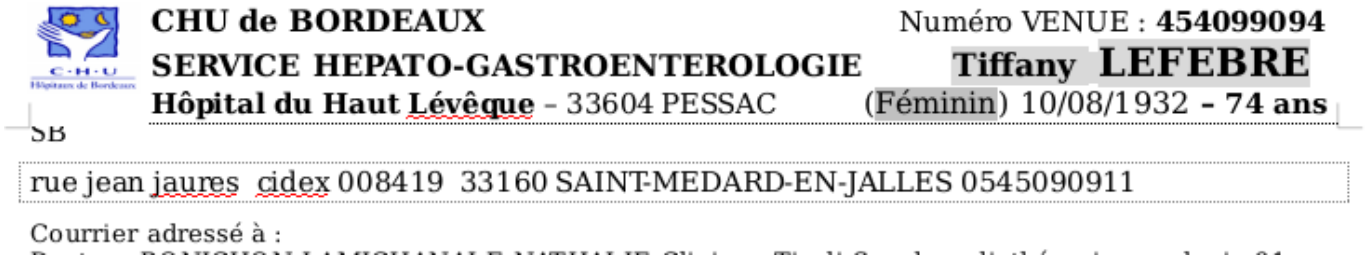
le médecin (nom, date de naissance, adresse, numéro de téléphone...) Ce para-texte peut aussi contenir le nom de nombreux médecins attachés au service d'où provient le courrier (figure **??**).

Outre le para-texte, les courriers peuvent contenir des images (logo de l'hôpital), des tableaux, des champs contenant des informations importées (nom et adresse du patient ou du médecin)... Tous ces éléments peuvent se transformer en artefacts lors du passage au format texte brut : des caractères supplémentaires, qui n'apparaissent pas lors de la visualisation du document, et apportent du bruit lors de son analyse. (capture **??**)

2.2 Ressources d'entités nommées

Outre les méta-données fournies avec les documents, notre méthode repose sur des listes de données possibles pour chaque catégorie d'information à anonymiser : noms et prénoms des personnes, noms de rues, de villes et de pays. Ces listes seront utilisées en deux occasions : pour trouver dans le texte des données à modifier ; et pour remplacer ensuite ces données. Il existe plusieurs possibilités concernant le choix de ces listes. Ci-dessous nous détaillons les avantages et inconvénients de chaque liste pour les noms de personne, étant entendu que des problèmes similaires apparaissent pour les autres types de données.

Pour la première tâche, qui consiste à repérer les données dans le texte, on peut rester à la plus petite échelle : utiliser uniquement les méta-données du document. Mais dans bien des cas cela ne sera pas suffisant. Des proches du patient peuvent être mentionnés mais être absents des méta-données, de même pour les médecins qui ne sont pas forcément tous indiqués. On peut compléter ces métadonnées avec des listes issues de la base de données de l'hôpital. On a alors une meilleure garantie d'exhaustivité sur tous les patients et tous les médecins. Mais les autres personnes mentionnées dans les documents, tels les proches ou un autre médecin qui a vu le patient avant son entrée à l'hôpital, restent inconnus. Enfin, un dictionnaire général des noms propres ou une liste issue de l'INSEE offre les meilleures chances de reconnaître un nom quel que soit le rôle de la personne mentionnée, mais ceci sans garantie de succès sur le patient et les médecins de l'hôpital.


CHU de BORDEAUX Numéro VENUE : 454099094
SERVICE HEPATO-GASTROENTEROLOGIE Tiffany LEFEBRE
Hôpital du Haut Lévêque - 33604 PESSAC (Féminin) 10/08/1932 - 74 ans
SB
rue jean jaures cidex 008419 33160 SAINT-MEDARD-EN-JALLES 0545090911
Courrier adressé à :

```
CHU de BORDEAUX Numero VENUE :  DOCVARIABLE NDA \? 2000/000.0
    \* MERGEFORMAT  454099094
SERVICE HEPATO-GASTROENTEROLOGIE  DOCVARIABLE
    PRENOMPATIENT \? 1003/000.0.1..0.3 \* MERGEFORMAT  Tiffany
       DOCVARIABLE NOMPATIENT \? 1001/000.0 \* MERGEFORMAT
    LEFEBRE
Hopital du Haut Leveque – 33604 PESSAC  (  DOCVARIABLE
    SEXEPATIENT \? 1004/000.0 \* MERGEFORMAT  Feminin )
    DOCVARIABLE DATEDENAISSANCEPATIENT \? 1005/000.0 \*
    MERGEFORMAT  10/08/1932  –  DOCVARIABLE AGEPATIENT \?
    1022/000.0 \* MERGEFORMAT  74 ans
Page  PAGE  3  sur  SECTIONPAGES  \* MERGEFORMAT  3
```

FIGURE 2.2 – En-tête d'un courrier et son équivalent en texte brut (données fictives)

TABLE 2.1 – Noms repérés par chaque ressource

	patient	médecins	autres
métadonnées	oui	certains	non
BDD	oui	oui	non
dictionnaire	la plupart	la plupart	la plupart

Le deuxième usage de ces listes est d'avoir des valeurs avec lesquelles remplacer les éléments à anonymiser. On doit choisir des listes différentes des précédentes, car l'objectif et les contraintes ne sont pas les mêmes. Notamment, on ne peut pas se limiter aux métadonnées du document, les choix possibles seraient bien trop réduits. L'échelle minimale est celle de la base de données de l'hôpital. Choisir cette petite échelle permet de fournir des résultats plus réalistes, ce qui rend moins visibles les informations qui n'auraient pas été identifiées et donc pas modifiées. Une liste courte augmente le risque de collisions, ce qui est une bonne chose car cela réduit encore le risque de ré-identifier une personne. Dans tous les cas, il est nécessaire d'exclure les noms les plus rares. On peut supprimer davantage de noms lorsqu'on dispose d'une base plus large.

De plus, on court toujours le risque de récupérer des données fausses ou problématiques dans les métadonnées ou la base de données hôpital. On peut avoir un soignant dont le nom est *ne plus écrire* [2] ou *interne 4*. On peut avoir un patient dont le nom de

2. cas réellement rencontré, on peut supposer que le soignant a voulu signifier son refus d'apparaître dans ce qui a servit à alimenter la base de données

famille est *Alzheimer* ce qui introduit un risque d'erreur. En choisissant la ressource la plus précise on limite la portée de ces cas problématiques.

Dans nos expériences, nous avons choisi d'utiliser les prénoms provenant des données de l'état-civil de quelques grandes villes, d'années variées [3] qui présentent l'avantage d'être librement disponibles.

Nous avons préféré utiliser les prénoms issus de plusieurs grandes villes afin d'obtenir un ensemble de prénoms plus générique et neutre à l'échelle de la France. Nous y avons ajouté les prénoms des patients et soignants de l'hôpital pour une meilleure exhaustivité. Pour des raisons de disponibilité, nous avons fait un choix différent pour les noms de famille : nous avons utilisé uniquement la base de données de l'hôpital. Pour l'étape de dé-identification, les deux bases ont été filtrées pour n'en retenir que les noms ayant une fréquence supérieure ou égale à 10.

La liste des villes de France associée à leur code postal a aussi été utilisée [4] ainsi que la liste des noms de rue de Bordeaux [5].

Les noms, prénoms et villes ont ensuite été filtrés à l'aide d'un dictionnaire général pour en retirer les tokens pouvant être autre chose qu'un nom propre. [6]

2.3 Ressources linguistiques

En outre, les données linguistiques suivantes vont nous être utiles pour détecter et catégoriser nos éléments identifiants.

Les triggers de noms, qui seront appelés simplement triggers par la suite, sont des tokens qui se présentent devant un nom et permettent de l'identifier comme tel. Notre liste de triggers est la suivante : *monsieur, madame, mademoiselle, docteur, professeur* ainsi que leurs diverses abréviations.

Certains noms de famille peuvent aussi être des noms de maladie. En temps normal *Alzheimer* est ignoré et laissé dans le texte. Mais si ce mot est présent dans la liste des noms de personnes que l'on recherche, il faut être vigilant à ne pas sur-étiqueter. Afin de pouvoir détecter ces situations particulières, nous utilisons une liste noire de noms de maladie pouvant être des noms de personne. Cette liste a été constituée à partir des versions française et anglaise de CIM10, CIM9, Meddra et Mesh de façon semi-automatique : les termes dotés d'une majuscule initiale ont été automatiquement extraits, puis filtrés à la main par un expert. [7]

Enfin nous utilisons également un dictionnaire des noms communs (http ://abu.cnam.fr/DICO/mots-communs.html) pour filtrer les noms de villes qui sont aussi des mots ordinaires (on, y, parce...). Cette ressource a été construite avec un filtre manuel pour prendre quand même les noms communs qui ont plus de chances d'être une ville (bordeaux, paris).

3. données de Paris, données de Nantes, données de Rennes, données de Toulouse et données de Strasbourg
4. villes et codes postaux
5. rues de Bordeaux
6. http://abu.cnam.fr/DICO/mots-communs.html
7. Frantz Thiessard, que je remercie

2.4 Conclusion

Notre méthode est conçue pour être appliquée à des fichiers au format CDA, qui présente l'avantage d'être un format standard [8]. Mais elle peut parfaitement être adaptée pour recevoir des métadonnées d'un fichier de format quelconque.

En ce qui concerne les ressources linguistiques, il est possible de livrer notre méthode complète et fonctionnelle en utilisant des listes générales. Il est donc possible de déployer très facilement cette méthode sur un nouvel hôpital, sans avoir besoin de collecter la moindre donnée outre le contenu des fichiers CDA. Un simple accès aux noms de famille et prénoms des patients et personnels permet d'utiliser des listes plus précises si le besoin se présente.

8. norme ISO 27932:2009

CHAPITRE 3

MÉTHODES

Sommaire

3.1 Introduction

La méthode proposée se déroule en deux étapes : le repérage des données identifiantes dans le texte, et la dé-identification par remplacement de ces données.

L'objectif de la première étape est d'établir les bornes des portions de texte à traiter, et le type d'entités ainsi identifiées : prénom, numéro de téléphone, date... Les types d'information recherchés peuvent être groupés en deux catégories : d'abord les informations de type chaîne de caractères, tels que les noms, prénoms, noms de rue... et ensuite les informations pouvant être représentées par un patron, tels que les numéros de téléphone, les dates, adresses mail... Nous verrons ci-après que ces deux types de données sont traitées de façon très différente.

La deuxième étape est le remplacement des données repérées. Ce remplacement doit être fait de façon cohérente et consistante, afin que les documents restent lisibles sans difficulté, et qu'un minimum d'information soit perdu.

3.2 Repérage des données identifiantes

```
ELECTROENCEPHALOGRAMME
<nom_famille>Martin</nom_famille> <prenomf>Denise</prenomf> ,
    78 ans
Date de naissance : <date>2/12/1930</date>
NCH
Examen realise le <date>23 avril 2005</date>
```

Les catégories traitées sont les suivantes :

- nom de famille
- prénom
- code postal
- ville
- pays (sauf la France, voir ci-dessous)
- autre élément d'adresse (rue...)
- âge (si supérieur ou égal à 90)
- numéro de téléphone
- adresse e-mail
- date
- numéro identifiant quelconque

En ce qui concerne les pays, il est souhaitable de les modifier car ils peuvent devenir identifiants dans des cas comme : "L'opération de Mme X. est reportée de 15 jours en raison de ses vacances en Hongrie." ou encore "Mr X. ayant vécu en Namibie jusqu'à ses huit ans, un test VIH a été effectué." En revanche, une mention de la France dans des documents produits en France à propos de patients français, modifiée vers le nom d'un autre pays, donnerait des documents étranges. Bien sûr, il est nécessaire de modifier ce paramètre en cas d'utilisation de cette méthode avec des documents issus d'un autre pays.

Il est possible de diviser ces étiquettes en deux catégories :

- les ENAMEX repérés par rapport à une liste constituée grâce aux métadonnées et/ou la BDD hôpital (nom, prénom, adresse) ;
- les expressions repérées à l'aide de patrons. Cela concerne les âges, les adresses électroniques (ENAMEX), les numéros de téléphone et certaines parties d'adresse tel le code postal (NUMEX) et enfin les dates (TIMEX).

La catégorie *prénom* est en réalité divisée en *prénom féminin*, *prénom masculin* et *prénom neutre*. Cette distinction permet de remplacer un prénom par un prénom du même genre autant que possible, ce qui évite de créer des erreurs d'accord dans le texte. Un prénom dit neutre est un prénom dont il est impossible de déterminer le genre, soit parce que ce prénom nous est inconnu, soit parce qu'il s'agit d'un prénom neutre (Dominique, Sasha, Charlie...). Les prénoms neutres sont traités de manière spécifique, comme nous l'expliquons plus loin.

3.2.1 Extraction du texte et des métadonnées

Nous commençons par extraire les métadonnées offertes par le format CDA, pour récupérer les données qui sont susceptibles d'apparaître dans le texte : nom et pré-

nom du patient et des soignants, éléments d'adresse, divers identifiants dont le NIP [1]. Par ailleurs, nous récupérons d'autres informations qui sont indirectement utiles : le sexe du patient servira à correctement remplacer son prénom, et la date d'édition du fichier servira de valeur par défaut lors de l'interprétation de dates incomplètes. Toutes ces informations sont extraites d'un document structuré, et cette structure nous permet d'associer à chaque information son type (nom, prénom, téléphone...)

En parallèle, le document au format .doc encapsulé dans le fichier CDA est extrait, puis converti en texte brut à l'aide de l'outil catdoc [2]. Il est ensuite nettoyé pour éliminer les marques de formatage et d'autres artefacts hors-texte. Avant de commencer à rechercher les données identifiantes, le texte est formaté de façon à uniformiser l'encodage et notamment la représentation des diacritiques. La même uniformisation est appliquée aux données extraites des métadonnées du CDA.

3.2.2 Noms d'entités

Nous disposons à présent d'une liste de chaînes de caractères représentant des informations identifiantes, ainsi que le type de chacune de ces informations. Nous effectuons une simple comparaison de ces chaînes avec le texte complet pour repérer la présence de ces informations. Des balises XML sont ajoutées lorsqu'une information est détectée.

Données numériques

Dans le cas des données numériques (NIP, téléphone...), nous recherchons les groupes de chiffres séparés par des groupes d'au maximum un seul caractère, et nous supprimons ces séparateurs avant d'effectuer la comparaison. Cela permet de comparer de façon appropriée les séquences comme *123, 1.2.3, 1-2-3*

Noms de personnes

Dans le cas des noms de personnes, une fois un nom repéré, nous le confrontons à une liste noire de termes médicaux pouvant aussi être des noms : *Alzheimer, Dawn*... Si un nom risque de correspondre à un terme médical, une condition supplémentaire doit alors être remplie : la présence d'un trigger précédent le mot (*Mr*, *Mme*, etc). Si cette condition n'est pas remplie, le mot est laissé dans le texte et n'est pas considéré comme un nom.

Certains noms peuvent être en plusieurs mots : nous considérons l'espace et le tiret comme des séparateurs valides.

Initiales de noms de personnes

Les noms de personnes peuvent aussi apparaître sous la forme d'initiales. Une initiale est définie comme un mot d'une seule lettre, toujours en majuscule, qui remplace un nom de personne. Pour repérer les initiales, nous nous basons sur deux indices contextuels : le voisinage d'un nom précédemment repéré, et la présence d'un trigger.

Nous commençons par examiner le voisinage des noms de famille et prénoms déjà repérés. Si un mot d'un seul caractère apparaît à proximité immédiate d'un nom

1. Numéro Identifiant Patient
2. http://linux.die.net/man/1/catdoc

de famille, nous l'étiquetons comme prénom. Le même processus est appliqué aux prénoms pour étiqueter des noms de famille. Nous pouvons ainsi repérer *j'ai examiné J. Kremer* et *j'ai examiné Jeannette K.* Ce premier passage permet uniquement de repérer un nom ou prénom si sa contrepartie a déjà été détectée.

Dans un second temps, nous allons chercher les noms constitués uniquement d'initiales. Nous allons ici sacrifier le rappel au profit de la précision, car de simples initiales peuvent facilement être confondues avec des unités (L. pour litre) ou diverses abréviations (A.M. pour matin). Nous allons donc nous limiter aux initiales précédées d'un trigger. Si un trigger est suivi d'un mot composé d'une seule lettre, ce mot est étiqueté comme nom de famille ; s'il est suivi de deux mots d'une lettre, ils sont étiquetés comme prénom et nom de famille.

En nous basant sur le genre du trigger (*monsieur* VS *madame*) nous pouvons éventuellement définir le genre.

Genre des prénoms

Nous allons avoir besoin plus loin de connaître le genre de chaque prénom pour pouvoir leur substituer un prénom du même genre, afin de ne pas générer de fautes d'accord dans le texte. Nous connaissons déjà le genre du prénom du patient grâce aux métadonnées, il va nous falloir découvrir celui des autres prénoms du texte.

Si un prénom est doté d'un trigger, une première méthode est de se baser sur le genre de ce trigger s'il en a un (*monsieur* VS *madame* mais pas *professeur*).

Une seconde méthode, qui sera suffisante pour la plupart des cas, est de s'appuyer sur un dictionnaire de prénoms.

Les cas restants correspondront aux prénoms absents du dictionnaire, ou indiqués comme neutres. Ceux-ci resteront neutres.

Il nous reste encore à traiter les initiales. Pour retrouver le prénom se cachant derrière une initiale, et par conséquent son genre, nous construisons un dictionnaire des noms de famille de ce document associés à leur prénom. Nous nous basons à la fois sur les métadonnées, qui associent prénoms et noms de famille, et sur la présence de couples nom/prénom dans le texte en proximité immédiate. Une fois le dictionnaire constitué, pour chaque initiale, nous cherchons un nom de famille à proximité immédiate. Il ne reste plus qu'à vérifier si ce prénom peut correspondre à notre initiale.

Le cas M

Un cas particulier concerne les occurrences de la lettre *M*, qui peut correspondre à une initiale de nom ou à une abréviation du titre *monsieur*. Il existe plusieurs manières de traiter l'ambiguïté :

— Si *M* est précédé d'un trigger (*Mme M*) alors il s'agit d'une initiale de nom
— Si *M* est suivi d'un nom de famille déjà repéré, nous vérifions si le prénom associé à ce nom de famille commence par un *M*. Si c'est le cas, alors il s'agit d'une initiale de prénom
— Si *M* est précédé d'un prénom, alors il peut s'agir d'une initiale de nom de famille. Cependant, cette règle peut provoquer des erreurs dans les listes de noms de médecins tels qu'on en voit dans les en-têtes et pieds-de-page de do-

cuments : un *M.* pris comme initiale de nom de famille peut être l'abréviation de *monsieur* s'appliquant au nom suivant dans la liste.

Si la désambiguïsation n'est pas possible, nous laissons le traitement de *M* à la charge du relecteur humain et le laissons donc dans le texte.

Villes

Les villes sont détectées à l'aide d'une liste de communes françaises, épurée des tokens pouvant être des noms communs. Tout comme pour les noms de personnes, une ville peut être en plusieurs mots. Le tiret, l'espace ainsi que le point (ex : *St. Ouen*) sont des séparateurs valides.

3.2.3 Dates

Le problème de l'identification et de la compréhension des dates en format libre au sein d'un texte ne possède pas de solution complète, bien que le problème soit commun. Nous sommes confrontés ici à un double problème : reconnaître ce qui est ou n'est pas une date ; et interpréter correctement la date. L'interprétation est cruciale car elle permettra, à l'étape suivante, de décaler la date correctement. Nous sommes alors confrontés à plusieurs problèmes :

— Le format libre : premier janvier, 1er janv, début janvier ...
— Les ambiguïtés : 02/04 = 2 avril ou février 2004 ?
— La coordination : du 3 au 5 avril 2010, entre mai et juillet 2005 ...
— Le bruit : valeurs numériques dans le texte qui peuvent être lues comme des dates...

L'ambiguïté, la coordination, et dans certains cas le bruit peuvent être résumés à un même problème, celui des dates incomplètes, dont une partie est implicite. Nous verrons plus loin que nous avons besoin de reconstruire ces dates de façon à être aussi proche que possible de la véritable date. *mi janvier* doit donc être interprété comme *15 janvier*, *début janvier* comme *premier janvier*, etc.

Les dates sont recherchées à l'aide d'un système d'expressions rationnelles. Plusieurs expressions sont exécutées successivement, de la plus gloutonne [3] (coordination de deux dates) à la plus courte (mois ou année isolés). Lorsqu'une date est identifiée, elle est partiellement interprétée afin de normaliser certaines parties (séparateurs des valeurs numériques, valeurs en chiffres ou en toutes lettres, orthographe des valeurs en toutes lettres...) et de compléter les dates partielles (interprétation de début/mi/fin de mois, des coordinations, recherche d'autres dates pour déterminer une année par défaut...)

3.2.4 D'autres patrons

Enfin, certaines informations inconnues à l'avance peuvent être repérées facilement à l'aide d'expressions rationnelles : les adresses mail et les numéros de téléphone. En fin de traitement, un patron exprimant toute suite de plus de 5 chiffres, contenant éventuellement des séparateurs, est associé à une catégorie générique "ID" qui permet de trouver tout ce qui est susceptible d'être un identifiant numérique quelconque.

3. qui consomme le maximum de caractères

TABLE 3.1 – Données consistantes

vrai nom	document 1 patient A	document 2 patient A	document 1 patient B
Dr Médard	Dr Martin	Dr Martin	Dr Duval

Cette dernière étape sacrifie précision au profit du rappel, ce qui est un faible coût face au risque de ré-identification important d'un numéro d'identification.

3.3 Déidentification

L'objectif de cette étape est de remplacer chaque information identifiante par une correspondance qui est :

— consistante pour les noms, éléments d'adresse
— aléatoire pour les numéros de téléphone, NIP et autres identifiants
— consistante mais brouillée pour les dates

Les informations consistantes doivent l'être au sein de différents documents concernant le même patient, mais pas entre les documents de différents patients.

Les informations aléatoires ne doivent montrer aucune consistance : si nous trouvons deux fois la même valeur à deux endroits différents du même document, elles n'ont pas vocation à être modifiées vers la même valeur.

Enfin les informations brouillées doivent respecter une certaine consistance, dont les limites seront explicitées plus bas.

La dé-identification doit être non-réversible : on aura besoin d'une méthode de *hash*. La fonction de hashage n'a pas besoin d'etre particulièrement robuste, étant donné que le *hash* lui-même ne se trouve pas dans les documents produits. Nous avons choisi d'utiliser l'algorithme sha-512[Grembowski et al., 2002], qui présente suffisamment de robustesse pour nos besoins. Nous avons également besoin qu'une même entrée ne produise pas la même sortie dans des documents appartenants à des patients différents. On utilisera pour cela un *salt* : le NIP.

Le NIP présente les avantages d'être toujours indiqué dans les méta-données d'un document et de ne pas être une information facilement accessible sans un accès aux données de l'hôpital.

Usuellement, avant d'être hashée, une chaîne est accolée à un *salt*. Un *salt* est utilisé lorsqu'il est nécessaire de s'assurer qu'une même chaîne hashée en deux occasions différentes ne produise pas le même hash. Dans notre cas, nous voulons obtenir cet effet si et uniquement si une chaîne se trouve dans les dossiers de deux patients différents. Si nous trouvons le même nom à plusieurs endroits dans le dossier du même patient, nous voulons précisément obtenir le même *hash*. En conséquence, nous avons besoin d'un "*salt*" consistant au sein du dossier d'un même patient, mais différent entre deux patients. Nous avons choisi le NIP pour remplir ce rôle.

Ne pas utiliser de véritable salt amène a une faiblesse majeure : si toutes les informations sont toujours hashées exactement de la même façon, il devient très facile de retrouver les valeurs d'origine. Si un attaquant cherche à retrouver les documents d'un patient en particulier au sein d'un grand corpus anonymisé, et s'il connaît son

NIP, il est alors suffisant de créer la rainbow table [4] utilisant ce NIP particulier, et de décrypter l'ensemble du corpus comme si ce NIP était toujours utilisé. Seuls les documents l'intéressant seront alors correctement décryptés, et une simple recherche de chaîne lui permettra de retrouver ces documents.

3.3.1 Noms

Pour les noms, rues, villes, pays, nous recherchons la consistance. Avec la méthode de hash présentée plus haut, cet objectif est facilement atteint. Cependant, dans le cas des noms de personnes, nous recherchons en outre à conserver la cohérence des initiales : Si *Mme Dupont* devient *Mme Martin*, tous les *D.* doivent devenir des *M.* Nous procédons en deux temps. D'abord, nous hashons uniquement l'initiale, qu'il s'agisse uniquement d'une initiale ou bien d'un nom complet. *Dupont* et *D.* deviendront tous deux un M. Ensuite, si nous disposons du nom complet, nous le hashons également, et le faisons correspondre à un nom dans la liste des noms commençant par cette lettre.

Ceci apporte la faiblesse évidente d'indiquer quels sont les noms commençant par la même lettre. Nous pouvons alors calculer la fréquence de chaque initiale et retrouver les véritables initiales, ce qui peut suffire à identifier les noms commençant par une lettre peu courante. Nous limitons cette faiblesse en regroupant les lettres les plus rares (définies par le nombre de noms pour cette lettre dans la liste de remplacements) qui pointent toutes vers une même initiale.

Toujours afin de limiter la correspondance des initiales, nous ajoutons un salt avant de hasher l'initiale : le type d'information, c'est à dire *nom de famille*, *nom féminin* ou *nom masculin*. Ainsi *Amélie* et *Andréa* seront remplacés par des prénoms commençant par la même lettre, mais *Alexandre* aura une initiale différente. Pour les prénoms dont nous ignorons toujours le genre, les initiales A à M sont considérées féminines, les initiales N à Z masculines. Cette séparation est arbitraire. Une nouvelle faiblesse est malheureusement introduite : une initiale peut être considérée à tord comme un prénom ou un nom de famille, un prénom peut alors avoir le mauvais genre.

3.3.2 Dates

Les dates constituent la partie délicate de cette étape, car il est nécessaire de les analyser correctement pour pouvoir les décaler de façon cohérente. Nous ne pouvons pas nous contenter d'une correspondance exacte entre les chaînes de caractères comme dans le cas des noms de personnes.

Notre objectif est de flouter les intervalles : introduire un petit décalage supplémentaire de façon à ce que deux dates, une fois modifiées, n'aient plus exactement le même écart, à quelques jours près. Il faut réaliser ce décalage supplémentaire de façon à ce que les dates restent dans leur ordre d'origine, qu'une date donnée mentionnée à plusieurs reprises soit bien convertie vers la même date, et que deux dates proches ne voient pas une différence trop importante dans la modification de leur écart. Par exemple, si un patient fait une rechute trois mois après un événement, ou trois mois et deux jours, cette différence est acceptable. Par contre, si un patient fait

4. une table de correspondance générée une seule fois entre un grand nombre de chaînes probables et leur hash, qui permet de retrouver la chaîne ayant généré un hash particulier [Oechslin, 2003]

une rechute le lendemain, l'écart de deux jours est suffisamment important pour gêner l'étude clinique. Ces contraintes ne nous permettent pas d'introduire le décalage aléatoire, car chaque date possible doit avoir une et une seule correspondance.

Si tout s'est déroulé correctement à l'étape précédente, les dates sont réécrites de façon à être lisibles par le module *dateutil* [5] de python. Nous manipulons ensuite un objet date, et déléguons ainsi à ce module la tâche de prendre en compte la longueur des mois, les années bissextiles, etc. Ceci nous assure de générer des dates véritablement existantes.

Nous décalerons toujours les dates vers le passé : la plupart des dates que nous pouvons rencontrer se trouveront dans le passé, proche de la date actuelle ; les dates dans l'avenir seront rarement éloignées du jour présent ; en revanche nous pouvons trouver occasionnellement des dates anciennes, comme des dates de naissance.

Nous allons procéder en deux étapes. Tout d'abord, nous soustrayons à toutes les dates un nombre de jours fixe entre 360 et 1080 jours, déterminé par le NIP via la fonction de hash afin qu'il soit constant à travers un dossier patient. Ceci nous garantit un décalage minimal et différent entre les patients. Dans un second temps nous allons ajouter le second décalage qui va nous permettre de flouter les correspondances. Pour ce faire, en partant d'une date maximale fixée arbitrairement au 31 décembre 2020, et en allant vers le passé, et nous allons déterminer la correspondance de chaque jour en intercalant un jour de plus tous les X jours. Ceci permet de ne pas introduire de gros écarts entre des dates voisines, et de bien répartir les jours supplémentaires. Une grande valeur de X introduit un décalage important, et peut décaler de plusieurs dizaines d'années les dates très anciennes, ce qui n'est pas forcément souhaitable. A l'inverse une faible valeur de X permet de ne pas trop repousser les dates anciennes, mais introduit moins de flou.

5. https://labix.org/python-dateutil

CHAPITRE 4

RÉSULTATS

Sommaire

4.1 Repérage des données

Suite à des complications techniques et un manque de temps, les résultats de notre méthode en termes de précision et de rappel sont susceptibles de comporter des erreurs de calcul, dues à la non prise en compte de certaines annotations. Les résultats ci-dessous doivent donc être considérés avec précaution.

Nous rappelons que notre objectif consiste à proposer une méthode de modification des données identifiantes des documents cliniques adaptée au traitement d'ensembles de documents associés à un même patient. Nous avons étendu notre travail à la détection de ces données identifiantes après avoir échoué à intégrer l'outil Medina [Grouin et al., 2009]. Le tableau 4.1 représente les résultats sur l'ensemble du corpus d'évaluation, toutes catégories confondues. La première ligne représente le nombre de vrais positifs stricts, la seconde inclue les éléments correctement identifiés, mais attribués à la mauvaise catégorie (ex : prénom au lieu de nom de famille). Les calculs de précision, de rappel et de F-mesure incluent les erreurs de catégorie dans les vrais positifs. Le tableau 4.2 détaille les résultats par catégorie. Dans ce tableau, les vrais positifs incluent les erreurs de catégorie, qui sont associées à la catégorie correcte.

Les initiales sont des prénoms ou noms de famille, et ne sont pas comptées dans les catégories prénom ou nom de famille. Les initiales bénéficient d'une catégorie propre pour l'évaluation car l'annotateur humain ne peut pas toujours déterminer si une initiale correspond à un prénom ou un nom de famille.

La première étape est le repérage des données. Pour l'évaluer, nous avons utilisé un corpus de 380 documents distincts du corpus de développement. Ce jeu de référence est annoté par un expert [1]. Tout comme le corpus de développement, il s'agit de courriers rédigés par des médecins, pouvant contenir d'autres documents tels des rapports de biologie.

Les catégories les plus présentes dans le corpus sont prénom, nom de famille et date, suivis par ville et initiale. Ce sont en effet des éléments présents dans l'en-tête et/ou

1. Sylvain Tessier externe en pharmacie

TABLE 4.1 – Résultats repérage

vrais positifs	1889
vrais positifs + erreurs de cat	2106
faux positifs	784
faux négatifs	1173
précision	0.80
rappel	0.71
f-mesure	0.75

TABLE 4.2 – Résultats repérage par catégorie

	VP	FP	FN	précision	rappel	f-mesure
prenom	267	14	236	0.95	0.53	0.68
nom de famille	604	107	259	0.85	0.69	0.77
initiale	134	10	238	0.93	0.36	0.52
ville	191	182	135	0.51	0.59	0.55
élément d'adresse	22	1	174	0.96	0.11	0.20
code postal	121	86	18	0.57	0.86	0.69
pays	0	13	1	0	0.08	0.01
age >90	-	-	-	-	-	-
mail	2	2	1	0.5	0.67	0.57
téléphone	25	0	23	0.99	0.52	0.67
id	28	236	7	0.11	0.8	0.18
date	712	52	81	0.93	0.89	0.91

la signature des courriers, qui peuvent se retrouver plusieurs fois par document lorsqu'un courrier en contient un autre, qu'il transmet. Ces en-têtes apportent beaucoup de bruit, et diminuent les performances pour ces catégories. Il ne serait pas difficile de supprimer ces en-têtes pour améliorer les performances.

Prénoms et Noms de famille

Concernant les noms et prénoms, ces deux catégories sont sujettes à de nombreuses erreurs de catégorie, de nombreux tokens pouvant être à la fois nom et prénom. Quelques faux positifs dans les noms de famille s'expliquent par des noms pouvant être également des noms communs. Cependant, les résultats dans ces deux catégories auraient dû être meilleurs. Par exemple, nous avons observé que des noms pourtant présents dans notre base de données n'ont pas été détectés dans les documents, ce qui est certainement dû à un défaut du système. Ce défaut sera éliminé dans une version ultérieure du système, ce qui permettra d'obtenir des résultats plus satisfaisants.

Il faut également considérer que les nom et prénom du patient sont ceux dont la dé-identification est la plus critique. Or, ceux-ci sont toujours présents dans les métadonnées du document : si on les considérait séparément des autres noms du texte, les performances de cette sous-catégorie seraient bien meilleures.

Initiales

Nous avons choisi d'exiger la présence d'un nom déjà identifié ou d'un trigger dans le contexte proche pour identifier une initiale. Ce choix entraîne une grande quantité de faux négatifs, et pour cette catégorie uniquement, cette faiblesse est acceptable : une initiale seule a moins de risque d'être identifiante que les éléments des autres catégories.

Villes

Comme pour les noms de famille, cette catégorie s'appuie sur une liste large comprenant de nombreux tokens, qui peuvent être également des noms communs, des noms de famille ou des prénoms, ce qui entraîne de nombreux faux positifs et erreurs de catégorie. Malheureusement nous avons aussi de nombreux faux négatifs, pouvant être dus à des mots filtrés par le dictionnaire de noms communs, et des villes en plusieurs mots dont certains ont été tronqués (ex : *Gif* au lieu de *Gif-sur-Yvette*). Cependant, comme pour les noms de personnes, nous avons observé que des noms de ville pourtant présents dans notre base n'ont pas été détectés. Ceci sera corrigé dans une version ultérieure du système.

Éléments d'adresse

La seule information que notre méthode associe à cette catégorie sont les noms de rue. Nous pouvons également y trouver des numéros d'appartement, boites postales, cedex... Comme les listes trouvées ne sont pas exhaustives, nous avons aussi des faux négatifs dans cette catégorie. De plus, cette catégorie est hétéroclite et difficile à définir.

Code postal

Nous observons une certaine quantité de faux positifs dus à des nombres à cinq chiffres présents, par exemple dans des résultats de biologie. Les quelques faux négatifs sont en réalité des erreurs de catégorie, où le code postal a été associé à une valeur numérique voisine et l'ensemble a été considéré comme un ID. ex : *bp 18 33049 Bordeaux*.

Pays

Les faux positifs correspondent à des noms de pays présents au sein d'un élément appartenant à une autre catégorie, le plus souvent des noms de rue (ex : *rue du Royaume-Uni*). L'unique faux négatif correspond à une mention de la Guadeloupe, que l'annotateur a marqué comme pays. Étant donné l'objectif de cette catégorie, et le contexte dans lequel *Guadeloupe* peut se trouver, cette annotation fait sens. La Guadeloupe n'était pas présente dans notre liste de pays, qui devrait donc être enrichie d'autres localités n'étant pas stricto sensu des pays.

Age, mail, téléphone et ID

Ces quatre catégories ne s'appuient pas sur des listes mais sur des patrons.

Les mails donnent des résultats étranges, qui sont peut-être imputables à une erreur du script d'évaluation. Pour les téléphones, certains faux négatifs sont dus à des ex-

pressions du type "05.50.04.00.05/06" ou bien des numéros accolés aux mots voisins, sans caractère séparateur.

Les ID contiennent une grande quantité de faux positifs, ce qui est le comportement voulu, cette catégorie faisant office de "voiture-balai" et récupérant tout ce qui est susceptible d'être un identifiant quelconque. Cette catégorie modifie à tord diverses valeurs numériques, notamment des résultats de biologie, mais c'est un prix acceptable face à la forte valeur identifiante d'un numéro d'identification.

Dates

Une date est le plus souvent une expression suffisamment complexe pour ne pas être confondue avec un autre élément, mais suffisamment régulière pour pouvoir être correctement détectée à l'aide de règles. C'est pourquoi cette catégorie a de relativement bonnes performances. Notre système d'expressions rationnelles a cependant ses limites. Les faux négatifs pourraient être récupérés au prix de faux positifs supplémentaires en élargissant notre méthode aux expressions ambiguës comme *4/5* : quatre mai, note ou quantité ?

Parmi ces expressions ignorées, car pouvant amener des faux positifs, nous trouvons aussi les années seules (qui peuvent alors correspondre à un nombre quelconque) ou les numéros de jour isolés. (*Je reverrai Mr X. le 2.*)

Il faut également prendre en compte que, parmi les dates, la valeur la plus identifiante est la date de naissance. Or, celle-ci est souvent plus régulière que les autres dates : on trouve rarement un autre format que DD/MM/YYYY ou DD/MM/YY. Si on évaluait les dates de naissances séparément, les résultats seraient sans doute supérieurs à ceux de la catégorie dates.

4.2 Déidentification

Cette étape se trouve au cœur de notre méthode. Qualitativement, en observant les documents produits lors de l'évaluation, nous pouvons considérer qu'ils sont toujours utilisables pour des tâches d'analyse clinique (pharmaco-vigilance, étude de cas, statistiques...). Seuls certains résultats de biologie, par exemple le nombre de plaquettes, sont perdus lorsqu'ils sont confondus avec des ID (les plaquettes ont classiquement des valeurs comprises entre 150 000 et 400 000 /mm3). Certaines modifications donnent des résultats étranges : par exemple, lorsqu'un prénom est confondu avec une ville. Mais où les documents restent lisibles et utilisables.

En revanche, il est plus complexe d'évaluer s'il est possible de les ré-identifier, ce qui est pourtant critique avant toute utilisation de notre méthode. Une évaluation complémentaire, effectuée par un cryptologue, serait appréciable.

4.3 Conclusion

Des catégories critiques (prénom, nom de famille, adresse) souffrent de nombreux faux négatifs. Cependant, nous espérons pouvoir grandement améliorer les résultats concernant les catégories les plus critiques, *prénom* et *nom de famille*, en corrigeant le fonctionnement du système. Nous pensons qu'en l'état, la partie repérage des données peut au mieux servir de pré-annotation pour un expert clinique.

Notre méthode capture une grande partie des informations identifiantes, mais pas suffisamment pour être utilisée de façon sécurisée et autonome. L'utilisation d'un outil complémentaire pour l'étape de repérage des données, ou des améliorations de notre propre méthode, sont nécessaires pour générer des documents utilisables de façon sûre.

La partie dé-identification ne bénéficie d'aucune preuve concernant sa sécurité, et ne doit pas non plus être utilisée à l'heure actuelle. En revanche, elle montre des résultats prometteurs. Les documents générés sont lisibles et cohérents pour un patient donné, alors que cette cohérence est perdue entre des documents liés à des patients différents, ce qui était notre contrainte principale. Le décalage avec flou des dates est fonctionnel. Moyennant une vérification de sa sécurité, cette étape pourrait être utilisée en l'état.

CHAPITRE 5

DISCUSSION

Sommaire

Quelles seraient les améliorations envisageables pour poursuivre ce travail et le rendre utilisable ?

5.1 Repérage des données

Il serait possible d'améliorer les résultats en s'appuyant sur Medina, tel que prévu à l'origine. Nos règles pourraient notamment bénéficier d'une meilleure prise en compte du contexte syntaxique, notamment pour les prénoms, noms de famille, initiales, villes, rues, codes postaux. Le contexte pourrait aider à déterminer le genre d'un prénom, la catégorie d'un mot apparaissant à la fois dans la liste de prénoms, des noms de famille et/ou des villes, repérer une initiale ou un code postal isolé...

Un système d'apprentissage automatique, par exemple de type CRF, permettrait cette meilleure utilisation du contexte.

Seuls les ID bénéficieraient très peu d'une méthode statistique, cette catégorie étant très hétéroclite. En revanche, il existe des outils dédiés à l'identification des valeurs numériques. L'utilisation d'un tel outil permettrait de filtrer une partie des valeurs amassées par les catégories ID et code postal.

Nous pouvons également ajouter aux catégories qui bénéficieraient d'un système statistique les *établissements de santé*, qui ont été annotés comme éléments potentiellement identifiants par l'annotateur humain. Nous n'avons pas pu trouver de base de données d'établissements de santé suffisamment propre et souple pour être utilisable. Par exemple, le nom "officiel" d'une pharmacie peut être "pharmacie de"+nom et prénom du propriétaire. En général, on ne trouve pas cette appellation complète dans un courrier de soignant à soignant. Nos résultats dans cette catégorie étant médiocres, nous l'avons supprimée. Il serait intéressant de proposer une autre approche pour la prendre en charge efficacement.

Les dates, qui ont bénéficié de davantage de travail que les autres catégories, obtiennent d'excellents résultats. Nous espérons que, dans une version future, accorder autant d'attention aux autres catégories permettra d'améliorer grandement les résultats.

5.2 Déidentification

L'objectif de notre méthode est de pouvoir anonymiser un ensemble de documents de manière cohérente pour un même patient, et différente pour différents patients. Nous n'avons pas trouvé d'autres travaux qui traitent de cet aspect dans l'état de l'art.

Contrairement à d'autres outils de dé-identification, comme Medina [Grouin et al., 2009], qui décalent chaque date d'un nombre de jours fixes de façon à conserver les intervalles de temps, nous avons voulu flouter les intervalles en intercalant des jours supplémentaires. Cependant, nous avions comme contrainte qu'une date soit toujours décalée vers la même date, et qu'on puisse anonymiser des documents au fur et à mesure, sans jamais que l'ordre des dates se croise. Nous avons donc dû conserver une correspondance un-à-un. A cause de cela, il s'est révélé impossible de trouver un équilibre entre trop ou ne pas assez décaler. Si nous ne décalons pas suffisamment, notre méthode revient à un simple décalage d'un nombre de jours fixes. Si nous décalons trop, les dates les plus éloignées de notre point de départ peuvent se retrouver décalées de plusieurs dizaines d'années. Là encore, une analyse cryptologique serait nécessaire pour apporter le réglage rendant cette tâche effective.

5.3 Conclusion

Le repérage des données constitue un bon point de départ, qui pourrait au choix être étoffé ou remplacé par une méthode qui a fait ses preuves. La sécurité de la dé-identification et des décalages n'est pas encore testée, mais les documents constitués sont exploitables et respectent les objectifs.

Conclusion

Notre objectif était de définir une méthode de dé-identification cohérente d'un ensemble de documents médicaux. Cette tâche se divise en deux parties : le repérage des données identifiantes, et la modification de celles-ci.

Le repérage des données ne donne pas de résultats suffisamment fiables pour être utilisé seul. Comme tout outil de dé-identification, la difficulté de la tâche et la nature critique des résultats, rendent la post-annotation humaine indispensable.

La dé-identification remplit l'objectif de créer des documents cohérents, faciles à lire et à ré-utiliser pour la recherche clinique. Il est cependant nécessaire d'effectuer une analyse cryptologique avant de pouvoir rendre public tout document anonymisé par notre méthode en son état actuel.

BIBLIOGRAPHIE

[1995] (1995). *MUC6 '95: Proceedings of the 6th Conference on Message Understanding*, Stroudsburg, PA, USA. Association for Computational Linguistics. – Cité page 11.

[Coates Stephens, 1992] Coates Stephens, S. (1992). The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, 26:441 – 456. – Cité page 11.

[Friedlin and McDonald, 2008] Friedlin, F. and McDonald, C. (2008). A software tool for removing patient identifying information from clinical documents. *Journal of the American Medical Informatics Association: JAMIA*. – Cité page 11.

[Grembowski et al., 2002] Grembowski, T., Lien, R., Gaj, K., Nguyen, N., Bellows, P., Flidr, J., Lehman, T., and Schott, B. (2002). Comparative analysis of the hardware implementations of hash functions sha-1 and sha-512. In Chan, A. and Gligor, V., editors, *Information Security*, volume 2433 of *Lecture Notes in Computer Science*, pages 75–89. Springer Berlin Heidelberg. – Cité page 24.

[Grouin et al., 2009] Grouin, C., Rosier, Dameron, and Zweigenbaum, P. (2009). une procédure d'anonymisation à deux niveaux pour créer un corpus de comptes rendus hospitalier. In Fieschi, M., Staccini, P., and O. Bouhaddou, C. L., editors, *Risques, technologies de l'information pour les pratiques médicales*. Springer. – Cité pages 12, 27 et 34.

[Khamis et al., 2005] Khamis, T. H. H., Tyczynski, J., and Berkel, H. (2005). Comparison of male and female breast cancer incidence trends, tumor characteristics, and survival. *Annals of epidemiology*. – Cité page 8.

[Neamatullah et al., 2008] Neamatullah, I., Douglass, M., Lehman, L.-w., Reisner, A., Villarroel, M., Long, W., Szolovits, P., Moody, G., Mark, R., and Clifford, G. (2008). Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, 8(1):32. – Cité page 11.

[Oechslin, 2003] Oechslin, P. (2003). Making a faster cryptanalytical time-memory trade-off. *Advances in Cryptology: Proceedings of CRYPTO 2003, 23rd Annual International Cryptology Conference*. – Cité page 25.

[Thiessard et al., 2012] Thiessard, F., Mougin, F., Diallo, G., Jouhet, V., Cossin, S., Garcelon, N., Campillo, B., Jouini, W., Grosjean, J., Massari, P., Griffon, N., Dupuch, M., Tayalati, F., Dugas, E., Balvet, A., Grabar, N., Pereira, S., Frandji, B., Darmoni, S., and Cuggia, M. (2012). Ravel: retrieval and visualization in electronic health records. *Stud Health Technol Inform.* – Cité page 13.

[Yandell and Majoros, 2002] Yandell, M. D. and Majoros, W. H. (2002). Genomics and natural language processing. *Nat Rev Genet*. – Cité pages 7 et 11.

[Zweigenbaum, 2008] Zweigenbaum, P. (2008). Natural language processing in the medical and biomedical domains: a parallel perspective. *Proceedings 3rd International Symposium for Semantic Mining in Biomedicine*. – Cité page 7.

Source des données

- Codes postaux et communes
  http://public.opendatasoft.com/explore/dataset/correspondance-code-insee-code-postal/

- Prénoms
  - Paris
    https://www.data.gouv.fr/fr/datasets/liste-des-prenoms-par-annee-prs/
  - Nantes
    https://www.data.gouv.fr/fr/datasets/liste-des-prenoms-des-enfants-nes-a-nantes-entre-2001-et-2012-laod/
  - Rennes
    https://www.data.gouv.fr/fr/datasets/prenoms-des-enfants-nes-a-rennes-de-2007-a-2011-
  - Toulouse
    https://www.data.gouv.fr/fr/datasets/prenoms-declares-a-l-etat-civil-de-toulouse-de-2003-a-2012-tm/
  - Strasbourg
    https://www.data.gouv.fr/fr/datasets/prenoms-inscrits-annuellement-a-l-etat-civil-de-la-ville-de-strasbourg-nd/

- Etablissements de santé
  https://www.data.gouv.fr/fr/datasets/extraction-du-fichier-national-des-etablissements-sanitaires-et-sociaux-finess-par-etablissements/

- Rues de Bordeaux
  http://opendata.bordeaux.fr/recherche/results/taxonomy%253A156

ANNEXE B

EXEMPLE DE DOCUMENT

Après extraction du texte brut à partir du document traitement de texte, et nettoyage des artefacts, on obtient un document tel que suit.

Bordeaux, le 20 décembre 2001
Docteur MARTIN ANTOINE
ORL

16 Rue Général Leclerc
33000 PESSAC

NI/NN SPD
0253.625
Cher Confrère,
Je vous prie de trouver ci-joint le compte-rendu de consultation de Madame Jeannette DUVAL, née le 11/09/1952.

Bien confraternellement.

Héléna LECOCQ

Interne
NI/NN SPD
0253.625
COMPTE-RENDU DE CONSULTATION

Concernant:

Madame Jeannette DUVAL, née le 11/09/1952

Date de la Consultation: 21/05/2011
Service: 1802 PEL PR DEHAIS EXT
Destinataires:

Docteur DURANT CLARA 8 rue des Buissons 33405 LACANAU
Docteur LEFROY MARC ORL 16 Rue General de Gaulle 33405 LACANAU
Motif de consultation

Suivi d'une polyarthrite rhumatoïde.
Histoire de la maladie

Patiente traitée par ENBREL 50 et METHOTREXATE depuis 2004.

Actuellement polyarthrite bien contrôlée.

Pas d'épisode infectieux.
Traitement habituel

ENBREL 50 mg/semaine

METHOTREXATE 5 cp/semaine

SPECIAFOLDINE

BIPROFENID 2/j ALD

DIANTALVIC ALD

UVEDOSE 1 tous les 3 mois
Examen clinique

Taille: 161 cm. Poids: 53 kg
Pas de révéil nocturne.

Pas de dérouillage matinal.

Indice articulaire: 0.

Indice synovial : 0.

Le reste de l'examen est sans anomalie.
Examens complémentaires

NFS normale.

VS : 29mm.

CRP<5mg/l.

Vitamine D : 19 g/ml.

Consultation dermatologique en avril 2007 : pas de lésion suspecte.

Consultation ophtalmologique : syndrome sec oculaire : prescription de larmes artificielles.

Radiographie pulmonaire : normale.

Radiographie des mains et pieds : carpite, légère progression des lésions des 1er et 4ème MCP.
Synthèse de la consultation

Polyarthrite rhumatoïde bien controlée par ENBREL et METHOTREXATE : poursuite du traitement.

Réévaluation dans 6 mois.

Supplémentation en vitamine D
Héléna LECOCQ

Interne

CR consultation Jeannette DUVAL Page

ANNEXE C

EXTRAIT DE CODE

Ci-dessous un extrait du module de détection des dates

```
#fonction principale du parseur de dates
#ne sont pas inclues ici les fonctions appelées par celle-ci


#separateurs
sep=r"[\.-/]+" #entre deux éléments nombre
sep1=r"[\. ’-]+" #entre deux éléments nombre/lettre
sep2=r"[\s\.,;:!?\(\)]" #début ou fin de chaine

debut="(?:^|(?<="+sep2+")"
#attention, il manque la parenthèse fermante du groupe de backreference,
#ceci afin de pouvoir ajouter un trigger qui ne soit pas consommé.
#bien penser à l’ajouter plus bas
fin="(?:$|(?="+sep2+"))"


#pas de groupe de capture ici, on les met à la compilation de la regex
#pour mieux voir ce qu’on capture
day=r"(?:0?[1-9]|1\d|2\d|30|31)"
month=r"(?:0[1-9]|1[012])"
year=r"(?:([12]\d)?\d\d)"
#les diacritiques sont normalisées plus haut dans l’outil, et peuvent être
#supprimées ou non
month_long=ur"(?:janvier|fe\u0301vrier|fevrier|mars|avril|mai|juin|juillet|
aou\u0302t|aout|septembre|octobre|novembre|de\u0301cembre|decembre)"

#construction des jours en lettres
#la normalisation unicode plus haut dans l’outil évite les versions
#particulières de "er" et "nd"
day_uniq=ur"(?:premier|deux|trois|quatre|cinq|six|sept|huit|neuf|dix|onze|
douze|treize|quatorze|quinze|seize|1er|2nd)"
day_unit=ur"(?:et[- ]un|deux|trois|quatre|cinq|six|sept|huit|neuf)"
day_except=ur"(?:dix[- ]sept|dix[- ]huit|dix[- ]neuf|trente[- ]et[- ]un)"
day_long="(?:"+day_uniq+"|(?:vingt[- ]"+day_unit+")|"+day_except+")"
```

```
#alternative long/court
day2="(?:"+day+"|"+day_long+")"
month2="(?:"+month+"|"+month_long+")"



########## à partir d'ici compilation et exécution des regex
########## date2() exécute la regex et ajoute les balises
########## correspondantes au xml



#les patterns en commentaire se lisent de la sorte :
#D = jour en chiffres
#day = jour en lettres
#même chose pour M/month et Y/year
#pour le reste ce sont des symboles de regex.
# | pour le choix, ? pour optionnel

### coordination
#normalement tous les cas de coordination sont gérés brutalement
#par dates_coo() avec une précision douteuse. j'ai rajouté ici
#en dur quelques cas de coordination plus proprement.

#du D|day month au D|day month Y (Y obligatoire sinon c'est pas un pb
#de coordination)
regex=debut+"(?<=du ))(?P<day1>"+day2+") +(?P<month1>"+month_long+")
?P<link>au) (?P<day2>"+day2+") +(?P<month2>"+month_long+")
( +(?P<year>"+year+"))"+fin
xml=date2(regex,xml,annee_defaut,options)

regex=debut+"(?<=de ))(?P<day1>"+day2+") +(?P<month1>"+month_long+")
(?P<link>a) (?P<day2>"+day2+") +(?P<month2>"+month_long+")
( +(?P<year>"+year+"))"+fin
xml=date2(regex,xml,annee_defaut,options)


#du D|day au D|day month Y?
#la plupart des coordinations sont gérées par la fonction dates_coo(),
#mais le cas d'un jour "isolé" doit être traité ici
#même si vous gardez uniquement dates_coo() et virez les cas particuliers
#au-dessus, laissez ce cas là.
regex=debut+"(?<=du ))(?P<day1>"+day2+") (?P<link>au) (?P<day2>"+day2+")
 +(?P<month>"+month_long+")( +(?P<year>"+year+"))?"+fin
xml=date2(regex,xml,annee_defaut,options)
regex=debut+"(?<=le ))(?P<day1>"+day2+") (?P<link>et le)
(?P<day2>"+day2+") +(?P<month>"+month_long+")( +
(?P<year>"+year+"))?"+fin
```

```
xml=date2(regex,xml,annee_defaut,options)
regex=debut+"(?<=le |du ))(?P<day1>"+day2+") (?P<link>jusqu'au)
(?P<day2>"+day2+") +(?P<month>"+month_long+")
( +(?P<year>"+year+"))?"+fin
xml=date2(regex,xml,annee_defaut,options)



#D M Y?
#dans un cas ambigu 04/05 choisit toujours D/M c'est un choix
regex=debut+")(?P<day>"+day+")"+sep+"(?P<month>"+month+")"+sep+
"(?P<year>"+year+")?"+fin
xml=date2(regex,xml,annee_defaut,options)

# D|day? month Y?
regex=debut+")((?P<day>"+day+"|"+day_long+")"+sep1+")?(?P<month>"+
month_long+")( +(?P<year>"+year+"))?"+fin
xml=date2(regex,xml,annee_defaut,options)


# mi-mois début spécial nécessaire car en temps normal - pas autorisé
#en tête de groupe comme séparateur
regex="(?:-| )(?P<month>"+month_long+")( +(?P<year>"+year+"))?"+fin
xml=date2(regex,xml,annee_defaut,options)

# month
regex=debut+")(?P<month>"+month_long+")"+fin
xml=date2(regex,xml,annee_defaut,options)

#en Y (on peut faire Y tout seul pour plus de rappel et moins de précision)
regex=debut+")(?<=\sen )(?P<year>"+year+")"+fin
xml=date2(regex,xml,annee_defaut,options)

#coordination. pour chaque date incomplète, complète avec les infos
#de la date suivante, sans chercher de lien entre les deux
if not options["conserver_texte"]:
xml=dates_coo(xml)

#dernier passage, on met des valeurs moyennes pour chaque endroit
#où on a encore des missing
if not options["conserver_texte"]:
xml=dates_fin(xml,annee_defaut)

return xml
```