Contributions à l'extraction d'information dans un entrepôt de données hospitalier : une aide pour la recherche clinique

Sébastien Cossin

université de BORDEAUX

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE SOCIÉTÉ, POLITIQUE, SANTÉ PUBLIQUE

SPÉCIALITÉ : SANTÉ PUBLIQUE, OPTION: INFORMATIQUE ET SANTÉ

Par **Sébastien COSSIN**

Contributions à l'extraction d'information dans un entrepôt de données hospitalier : une aide pour la recherche clinique

Sous la direction de : **Vianney JOUHET**
Co-directeur : **Gayo DIALLO**

Soutenue le mercredi 28 septembre 2022

Membres du jury :

Rodolphe Thiébaut	PUPH	Université de Bordeaux	Président
Adrien Coulet	Maître de conférences HDR	Université de Lorraine Nancy	Rapporteur
Bastien Rance	Maître de conférences HDR	Université de Paris Diderot	Rapporteur
Sandra Bringay	Professeure des universités	Université Paul Valéry - Montpellier	Examinatrice
Lina Soualmia	Maître de conférences HDR	Université de Rouen Normandie	Examinatrice
Gayo Diallo	Professeur des universités	Université de Bordeaux	Co-directeur
Vianney Jouhet	Praticien hospitalier	Université de Bordeaux	Membre invité

Abstract

The development of digital technologies has led to the digitization of medical information and the transformation of paper records into electronic health records (EHRs). The data generated in a hospital contains valuable information for medical research. Hospitals have set up clinical data warehouses (CDW) to facilitate the secondary use of the data. In a CDW, researchers need to identify eligible patients for a clinical study and return to an EHR to complete the electronic case report form of a study. The main difficulty is the unstructured nature of the free text medical information. Natural language processing methods are needed to structure the data to facilitate its interrogation and retrieval.

The objective of this thesis was to develop tools and methods to help researchers conduct feasibility studies and find information in an EHR. The main contributions of this thesis are the following:

- A French drug terminology

  Many studies are looking at the use, efficacy, and tolerance of medicines in daily life. Medicines can also help to identify some diseases. The lack of a standardized drug terminology has led to the construction of Romedi, a French drug terminology, which offers good performance in detecting and identifying drugs in hospital data.

- A scalable semantic annotator

  Semantic annotation consists of linking sequences of words in a document to concepts of a terminology. It enables the detection and indexing of medical concepts. How to index millions of documents in a CDW with medical terminologies containing several hundred thousand terms? In this work, we propose a new algorithm, IAMsystem, which

is scalable to the size of a data warehouse and whose complexity depends little on the size of a terminology.

- An inventory of French medical abbreviations

  Abbreviations are widely used in medicine. They add complexity to natural language processing tasks and must be taken into account by a semantic annotator. This work presents two algorithms to automatically detect abbreviations from a corpus of medical documents and offers the first inventory of abbreviations extracted from French hospital data.

- Linking hospital records to death certificates

  The vital status of individuals is of central importance for many epidemiological studies and feasibility studies need to know whether eligible patients are alive or dead. Large volumes of data require a strategy to reduce the number of comparisons. We show that a vector space model offers excellent results in reducing the number of comparisons and that it is possible to automatically generate a gold standard from hospital data for linking hospital data and death certificates by machine learning.

- A web application to support the review of EHRs

  An interface, SmartCRF, has been developed to quickly search for information in an EHR. It comprises a lifeline, a search engine, a document viewer, and a recommendation system. Compared to the hospital software, it reduces the time spent on checking the inclusion and exclusion criteria of a feasibility study and facilitates the completion of an electronic case report.

Résumé

Le développement des technologies numériques a conduit à la numérisation des informations médicales et à la dématérialisation des dossiers papiers en dossiers patients informatisés (DPI). Les données générées dans un hôpital contiennent des informations précieuses pour la recherche médicale. Les hôpitaux ont mis en place des entrepôts de données (EDS) pour faciliter l'utilisation secondaire des données. Dans un EDS, les chercheurs ont besoin d'identifier les patients éligibles à une étude clinique et de retourner au DPI pour remplir le cahier d'observation électronique d'une étude. La principale difficulté réside dans le caractère non structuré des informations médicales présentes sous forme de texte libre. Des méthodes de traitement automatique de la langue sont nécessaires pour structurer les données afin de faciliter leur interrogation et leur extraction.

L'objectif de cette thèse était de développer des outils et des méthodes pour aider les chercheurs à mener des études de faisabilité et à trouver des informations dans un DPI. Les principales contributions de cette thèse sont les suivantes:

- Une terminologie sur les médicaments en langue française

  De nombreuses études s'intéressent à l'utilisation, l'efficacité et à la tolérance des médicaments en vie réelle. Les médicaments permettent aussi d'identifier certaines maladies. L'absence d'une terminologie normalisée du médicament a conduit à la construction de Romedi, référentiel ouvert du médicament, qui offre de bonnes performances pour détecter et identifier les médicaments dans les données hospitalières.

- Un annotateur sémantique scalable à un entrepôt de données

  L'annotation sémantique consiste à relier des séquences de mots d'un document aux concepts d'une terminologie. Elle permet la détection et l'indexation de concepts médicaux.

Comment indexer des millions de documents d'un EDS avec des terminologies médicales contenant plusieurs centaines de milliers de termes ? Dans ce travail, nous proposons un nouvel algorithme, IAMsystem, scalable à l'échelle d'un entrepôt de données et dont la complexité dépend peu de la taille d'une terminologie.

- Un inventaire de sens des abréviations médicales

  Les abréviations sont largement utilisées en médecine. Elles ajoutent de la complexité aux tâches de traitement automatique de la langue et doivent être prises en compte par un annotateur sémantique. Ce travail présente deux algorithmes pour détecter automatiquement des abréviations à partir d'un corpus de documents médicaux et propose le premier inventaire d'abréviations issu de données hospitalières en langue française.

- Une stratégie d'appariements de données hospitalières avec les certificats de décès

  Le statut vital des individus est d'une importance capitale pour de nombreuses études épidémiologiques et les études de faisabilité ont besoin de connaître si les patients éligibles sont vivants ou décédés. Les grands volumes de données nécessitent de recourir à un stratagème pour diminuer le nombre de comparaisons. Nous montrons qu'un modèle d'espace vectoriel offre d'excellents résultats pour diminuer le nombre de comparaisons et qu'il est possible de générer automatiquement un gold standard à partir de données hospitalières pour apparier données hospitalières et certificats de décès par apprentissage automatique.

- Une interface pour la revue des DPI

  Une interface, SmartCRF, a été développée pour rechercher rapidement des informations dans un DPI. Elle est constituée d'une ligne de vie, d'un moteur de recherche, d'une visionneuse de documents et d'un système de recommandation. Par rapport au logiciel métier, elle permet de diminuer le temps passé à vérifier les critères d'inclusion et d'exclusion d'une étude de faisabilité et elle facilite le remplissage d'un cahier d'observation électronique.

A ma fille Évy.

Remerciements

A mes directeurs de thèse, Dr Vianney Jouhet et Pr Gayo Diallo Merci d'avoir accepté de m'accompagner sur ce sujet, de m'avoir fait confiance et de votre aide précieuse pour la rédaction de ce manuscrit.

Au Professeur Rodolphe Thiébaut Merci de présider le jury de cette thèse. Soyez assuré de ma reconnaissance pour votre soutien et la confiance que vous m'avez accordée au sein du Service d'Information Médicale du CHU de Bordeaux.

Aux rapporteurs, M. Adrien Coulet et M. Bastien Rance Merci d'avoir accepté d'examiner ce travail. J'espère que nous aurons l'occasion de travailler ensemble sur ces sujets d'informatique médicale qui nous passionnent.

Aux autres membres du jury, le professeur Sandra Bringay et Mme Lina Soualmia Je vous remercie d'avoir accepté de participer à mon jury de thèse, d'avoir pris le temps de lire ce document et de vous être intéressées à mes travaux de recherche.

A l'équipe ERIAS Je remercie vivement tous les membres de l'équipe ERIAS, en particulier Gayo, Fleur, Frantz, Vianney, Valérie, Romain, Bruno, Marie-Odile. C'est une chance de travailler et d'échanger au sein d'une équipe aussi ouverte. Merci de m'avoir formé à l'informatique médicale et de m'avoir confié de nombreuses responsabilités.

Au Dr Moufid Hajjar Merci m'avoir accueilli chaleureusement dans l'unité IAM et pour ton soutien inconditionnel. Ce travail n'aurait pas été possible sans toi.

A l'unité IAM du CHU de Bordeaux Merci à toutes les personnes avec qui j'ai eu la chance de travailler ces nombreuses années, Luc, Aymeric, Camille, François, Romain, Florence.
Merci à tous les internes, externes et stagiaires qui m'ont beaucoup aidé dans la réalisation de cette thèse, Romain, Grégory, Nadine, Sidali, Marithée, Arnaud, Marine, Kévin, Philippine, Guillaume, Bertrand, Thomas, Hélène, Tatiana, Alix, Yvon, Nicolas, Iban, Margaux, Louis, Benjamin.

Au CHU de Bordeaux Merci à toutes les personnes qui ont participé directement ou indirectement à ce travail en apportant leur soutien et leur aide aux projets que nous avons menés et en particulier aux équipes de la DSI, Mme Valérie Altuzarra, M. Hervé Delengaigne, M. Thierry Barthe, M. Olivier Jecker.

Au Professeur Geneviève Chêne Merci de m'avoir donné l'opportunité de rejoindre cette équipe passionnante et de votre soutien.

Au Professeur Roger Salamon Merci pour votre soutien à l'informatique médicale pendant ces nombreuses années.

A ma famille et mes amis A mes parents et à toute ma famille pour m'avoir fourni l'environnement nécessaire, pour m'avoir soutenu dans les moments difficiles. A mes amis qui, même éloignés, sont toujours présents, et particulièrement mes amis d'enfance. A Nora pour son aide et son soutien inestimable au quotidien. A Mr Chipper pour sa sagesse et sa compagnie.

A la communauté du logiciel libre Merci pour la quantité et la qualité des outils que vous développez et partagez. Ce travail s'appuie sur le vôtre.

Table des matières

Glossaire

ATC	Anatomical Therapeutic Chemical, système de classification des médicaments
API	Application Programming Interface
ANSM	Agence Nationale de Sécurité du Médicament et des produits de santé
BERT	Bidirectional Encoder Representations from Transformers
CDW	Clinical Data Warehouse
CIM-10	Classification Internationale des Maladies, 10ème révision
CRF	Case Report Form
CSV	Comma Separated Values
DPI	Dossier Patient Informatisé
eCRF	electronic Case Report Form
EDS	Entrepôt de Données de Santé
EHR	Electronic Health Record
ETL	Extraction, Transformation and Load
HIS	Hospital Information System
NER	Named Entity Recognition
TAL	Traitement Automatique de la Langue
TSV	Tab Separated Values
UMLS	Unified Medical Language System

Chapitre 1

Introduction générale

Ce chapitre est une introduction générale à la thèse. La section Contexte décrit les données hospitalières, les enjeux d'utilisation secondaire des données et l'entrepôt de données de santé (EDS) du CHU de Bordeaux qui est le principal objet d'étude de cette thèse. La section Problématiques décrit les problématiques rencontrées en recherche clinique pour réutiliser les données.

1.1 Contexte

Système d'information hospitalier Dans un établissement de santé, un système d'information Hospitalier (SIH) facilite la gestion de l'ensemble des informations médicales et administratives afin d'une part d'améliorer la qualité des soins et, d'autre part, de permettre une meilleure maîtrise des coûts [1]. La priorité d'un SIH est de permettre aux acteurs du système de soins de prendre en charge efficacement les patients, notamment en permettant un accès facilité et complet à l'ensemble des données de santé d'un patient à travers le dossier patient informatisé (DPI). Un DPI peut être défini comme l'ensemble des informations sous format électronique qui est lié à l'état de santé ou aux soins de santé passés, présents et futurs d'une personne [2]. Les données médicales d'un DPI sont dispersées au sein des nombreux logiciels d'un SIH comme les logiciels de gestion des dossiers médicaux (ex: DxCare®), les logiciels de gestion des prescriptions médicamenteuses, les logiciels de gestion des résultats de biologie ou d'imagerie. Le logiciel de gestion des identités patient assure la création et la gestion d'une

identité patient unique au sein d'un établissement. Chaque logiciel est indépendant des autres et possède sa propre base de données, les données d'un SIH sont donc organisées en silo.

Utilisation secondaire des données Les données d'un DPI sont collectées pour assurer la prise en charge médicale d'un patient. L'utilisation de données pour des raisons autres que celles prévues à leur acquisition porte le nom d'utilisation secondaire de données [3]. L'utilisation secondaire de données médicales est définie comme *l'utilisation d'informations personnelles sur la santé à des fins autres que les soins directs, y compris, mais sans s'y limiter, à des fins d'analyse, de recherche, de mesure de la qualité/sécurité, de santé publique, de paiement, de certification ou d'accréditation des prestataires, de marketing et d'autres activités, y compris des activités strictement commerciales* [4]. En 2007, l'American Medical Informatics Association a rédigé un livre blanc pour souligner l'importance de l'utilisation secondaire des données de santé et pour inciter les décideurs politiques aux États-Unis à lever les verrous techniques et juridiques à leur exploitation [5].

L'utilisation secondaire des données médicales a une place importante en recherche clinique [6]. De nombreux travaux de recherche s'appuient uniquement sur l'utilisation secondaire de données présentes dans les dossiers de soins [4], ce qui évite de re-solliciter le patient pour la collection des données. L'une des difficultés techniques à l'utilisation secondaire est de mobiliser les données en silos des différents logiciels d'un SIH [7]. Au CHU de Bordeaux, c'est une approche intégrative qui a été choisie pour désiloter les données et les intégrer dans un EDS.

Entrepôt de données de santé Un EDS est une solution intégrative pour faciliter l'utilisation secondaire des données. Un EDS est une base de données regroupant les données médicales de nombreuses applications métiers composant le SIH. Les données sont extraites des bases de données des applications sources, transformées (conversion de format, remodélisation) puis chargées dans une base de données cible dont la structure est pensée pour un usage secondaire. Ce processus est appelé ETL (pour Extract, Transform et Load). Il existe autant de processus d'ETL qu'il existe de sources de données à extraire dans le SIH. Un EDS permet de désiloter les données en les réunissant dans une base de données commune, permettant ainsi d'envisager le développement d'outils pour répondre à de nombreuses finalités d'utilisation secondaire [8].

Depuis 2017, l'EDS du CHU de Bordeaux agrège dans une base de données Oracle®, optimisée pour l'interrogation de données de grande dimension, les données biomédicales produites dans le cadre du soin. Le format de données utilisé est i2b2 développé par Harvard en 2004 [9]. I2b2 est la solution open-source la plus utilisée à travers le monde avec plus de 110 établissements de santé l'ayant déployée en 2014 [10]. En France, i2b2 a notamment été déployé à l'AP-HP [11] et l'Hôpital Européen Georges Pompidou [8].

L'EDS du CHU de Bordeaux agrège, via des processus d'ETL, les données de 6 domaines correspondant à 12 sources différentes (figure 1.2). Les données de l'ensemble des patients venus au CHU depuis 2010, en hospitalisation ou en consultation, sont intégrées ce qui représente plus de 2 millions de patients, environ 16 millions de venues et plus de 2 milliards d'observations. Une observation correspond à une ligne dans la table centrale d'i2b2 (observation_fact) qui correspond à un élément d'une source de données rattaché au séjour d'un patient : un résultat biologique, une administration médicamenteuse, un compte rendu radiologique etc.... Les principales données d'un DPI intégrées dans l'entrepôt de donnée de santé du CHU de Bordeaux sont les suivantes :

- Les documents, qu'ils soient produits dans DxCare® ou issus d'une autre source (OPERA® pour la partie bloc opératoire, XPLORE® pour la partie imagerie médicale...). L'intégration des documents est effectuée à partir d'un serveur de documents mis en place et maintenu par une équipe de la direction du système d'information. Ces documents sont souvent rédigés dans le logiciel Microsoft Word®. L'ETL extrait le contenu textuel d'environ 60 millions de documents.

- Les traitements administrés aux patients issus de 3 grandes sources :

  - DxCare®, pour les traitements médicamenteux conventionnels.
  - Traceline®, pour les prescriptions et administrations de produits sanguins labiles et autres produits médicaux d'origine humaine (culots globulaires, plasma frais congelé...).
  - Chimio®, pour les traitements chimio-toxiques.

- Les notes cliniques rédigées pendant une hospitalisation sont renseignées dans des formulaires DxCare®. Le logiciel DxCare® permet la création de formulaires spécifiques à une activité clinique pour faciliter la saisie d'information. Un formulaire DxCare® est composé de couples questions-réponses où la réponse peut correspondre à une donnée structurée (la liste des réponses est alors prédéfinie) ou non structurée sous forme de texte libre court ou long. La figure 1.1 montre le formulaire utilisé aux urgences. La majorité des réponses aux formulaires sont courtes (ex : température, poids, antécédents, traitements) et parfois longues (ex : synthèse clinique).

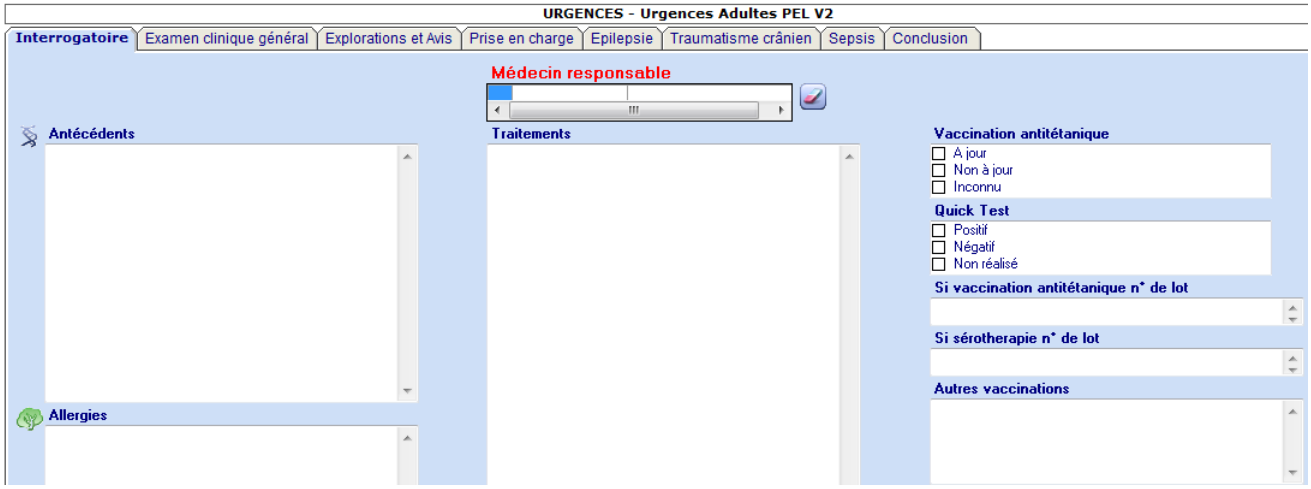


Figure 1.1: Formulaire *Urgences Adultes PEL V2* de DxCare® utilisé au CHU de Bordeaux. Le formulaire est composé de plusieurs pages (interrogatoire...) et de questions-réponses sur chacune. Certaines informations sont saisies en texte libre comme les antécédents médicaux ou les traitements, d'autres sont des cases à cocher ou des listes déroulantes.

- Les données de biologie, provenant de Synergie®, Glims® et du serveur de résultats DxCare®.
- Les données d'anatomo-pathologie, provenant de Diamic® et de la tumorothèque.
- Les données du Programme de Médicalisation des Systèmes d'Information (PMSI), contenant des informations sur les actes et les diagnostics d'une hospitalisation [12].

Les images radiologiques ne sont pas intégrées dans l'EDS mais les comptes rendus rédigés par les radiologues sont présents. Un DPI contient donc trois grands types de données : les données structurées comme les données de laboratoire ou les codes PMSI, les données semi-structurées des formulaires et les données non structurées des documents textuels.

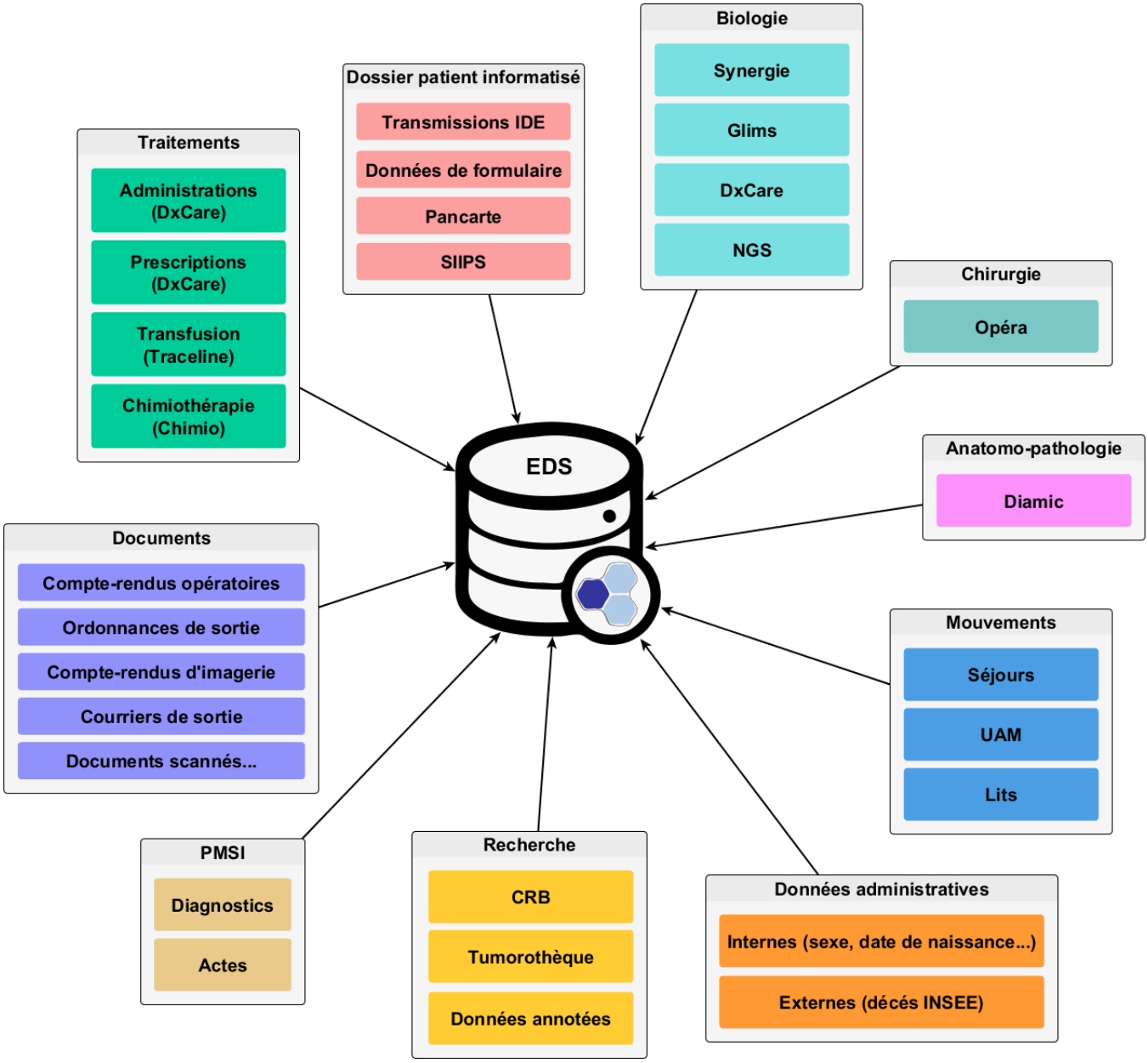


Figure 1.2: Sources de données intégrées dans l'EDS du CHU de Bordeaux.

1.2 Enjeux en recherche clinique

1.2.1 Identification de cohortes

L'une des principales finalités d'un EDS en recherche clinique est l'identification de cohortes [13]. Une cohorte est un ensemble de patients possédant des critères clinico-biologiques similaires et recherchés pour la réalisation d'une étude. L'identification de cohortes dans un EDS permet de mener des études rétrospectives et d'accélérer le recrutement de patients dans les essais cliniques [4, 14]. Pour trouver des patients éligibles à une étude, des outils de requêtage sont disponibles pour interroger les données d'un EDS. Ils permettent de sélectionner des critères clinico-biologiques structurés et de rechercher des mots clefs dans les documents textuels. La figure 1.3 montre l'exemple d'une requête avec l'outil i2b2 Workbench[10].

Ces outils de requêtage offrent rarement d'excellentes performances en termes de sensibilité et de spécificité à cause de l'hétérogénéité des données d'un DPI. La principale difficulté pour identifier une cohorte dans un EDS est la complexité du langage naturel qui rend difficile l'interrogation des données [3]. La majorité des informations cliniques sont présentes dans le texte libre [4] et donc difficilement requêtables. Le texte libre permet aux professionnels de santé d'exprimer des histoires cliniques riches, des réflexions et des raisonnements complexes [13, 15]. Les abréviations, les termes ambigus, les anaphores, le contexte, les relations temporelles entre évènements sont autant de difficultés à résoudre par les algorithmes de traitement automatique de la langue.

1.2.2 Remplissage d'eCRF

Une autre finalité est d'aider au remplissage du cahier d'observation papier ou électronique (eCRF), principal support à la collecte d'information d'une étude. Un eCRF contient plusieurs champs structurés pour collecter les données d'un patient au sein d'une étude [4]. Certains champs d'un eCRF peuvent être complétés automatiquement si les données de l'EDS sont structurées pour permettre un gain de temps et éviter des erreurs de saisie. Utiliser un entrepôt de données pour pré-remplir des cahiers d'observation électronique avec des données structurées du système d'information, comme les résultats biologiques, apporte des bénéfices

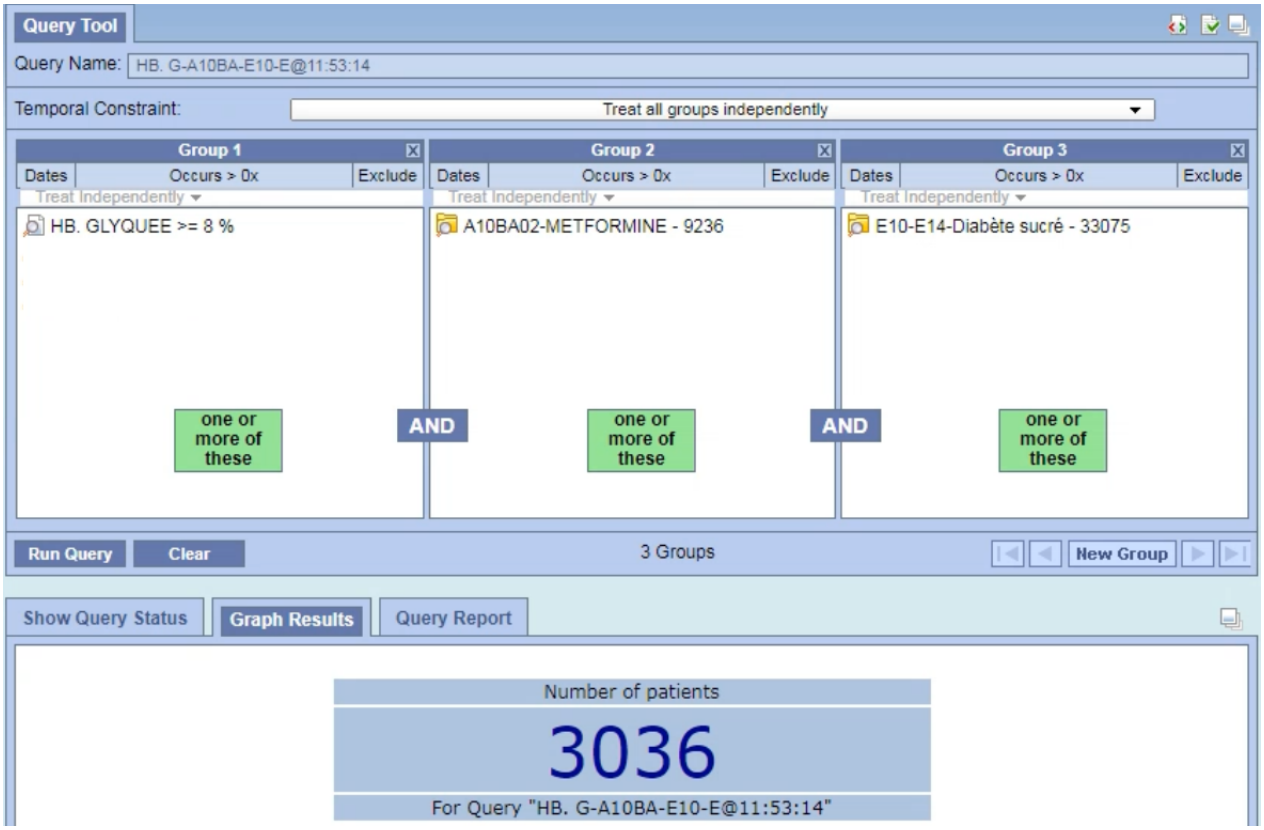


Figure 1.3: Réalisation d'une requête dans l'interface i2b2 Workbench pour rechercher des patients dans l'EDS du CHU de Bordeaux. Dans cet exemple, les patients ayant une hémoglobine glyquée supérieure ou égale à 8%, une administration de metformine et un code CIM-10 de diabète sucré sont recherchés.

évidents : le remplissage manuel, fastidieux et chronophage de données structurées est remplacé par l'alimentation rapide et sans erreur d'une machine. Comme la majorité des informations est présente dans les documents textuels, il est pertinent de se demander dans quelle mesure une machine peut aider les chercheurs à structurer automatiquement les informations contenues dans les documents médicaux.

1.3 Problématiques

L'extraction d'information consiste à structurer automatiquement les informations cliniques d'un document textuel. L'extraction d'information est une tâche essentielle pour faciliter l'utilisation secondaire des données [16]. La structuration automatique d'informations contenues dans les documents textuels facilite l'interrogation des données, l'identification de cohortes et le remplissage automatique d'eCRF. Les approches de l'extraction d'information sont divisées en deux grandes catégories : les approches basées sur des règles et l'apprentissage automatique [16]. Les solutions proposées aux utilisateurs nécessitent de prendre en compte leurs besoins et leurs contraintes. Premièrement, les ingénieurs informatiques sont au service des utilisateurs ; ils doivent sélectionner les méthodes adaptées aux spécificités des données hospitalières, à la difficulté de la tâche et aux contraintes des utilisateurs. Dans le contexte d'un EDS, les données annotées sont un luxe : chronophages et coûteuses à produire, elles nécessitent une expertise du domaine, un guide d'annotation et des outils spécialisés dans l'annotation de documents. Par conséquent, l'utilisation d'algorithmes d'apprentissage supervisé est plus souvent l'exception que la règle. Deuxièmement, dans le cadre d'un EDS, les algorithmes de structuration de l'information doivent favoriser la précision au rappel. Il n'est pas acceptable qu'une machine structure des informations de façon erronée. Une erreur de structuration conduirait à alimenter un cahier d'observation avec une information fausse ce qui pourrait conduire à des résultats épidémiologiques ininterprétables ou erronés. Si une information est structurée automatiquement, l'expert humain ne devrait pas avoir besoin de vérifier l'information extraite, la précision devrait être proche de 100%. En pratique, lorsque la tâche de structuration de l'information est trop complexe, elle doit être réalisée par un expert humain.

Notre stratégie a consisté à identifier les informations importantes à structurer automatique-

ment et fournir une solution dans les situations complexes. Le statut vital et la date de décès sont des informations très utiles pour les études de faisabilité et pour les études rétrospectives. La publication en open data de certificats de décès établis par l'Insee depuis 1970 a rendu possible un travail sur la structuration automatique de l'information sur le statut vital. Le chapitre 2 présente la stratégie développée pour apparier les données hospitalières et les certificats de décès.

Une revue de la littérature a montré que les outils développés en extraction d'information visaient principalement à structurer les maladies et les médicaments dans les documents médicaux [16]. Structurer ces informations était aussi une priorité pour de nombreuses études cliniques menées au CHU de Bordeaux. L'absence d'une terminologie normalisée du médicament a conduit à la construction de Romedi, référentiel ouvert du médicament, qui offre de bonnes performances pour détecter et identifier les médicaments dans les données textuelles. Romedi permet de coder automatiquement les médicaments notés dans des formulaires et dans les ordonnances de sortie. Le graphe de connaissance de Romedi permet d'intégrer des données hétérogènes pour interroger plus facilement les données relatives aux médicaments. Le chapitre 3 est consacré au développement de Romedi.

Les antécédents médicaux sont fréquemment mentionnés dans les formulaires (figure 1.1). Les terminologies médicales en langue française contiennent de nombreux concepts de maladies pour coder l'information sur les antécédents. Cependant de nombreux concepts ne sont pas détectés à cause de la présence d'abréviations. La fréquence élevée d'abréviations est un frein important à la détection de concepts médicaux dans les données hospitalières. Un travail spécifique a été réalisé pour les détecter et créer un inventaire d'abréviations médicales en langue française. Le chapitre 5 est dédié à la création de cet inventaire.

La structuration des informations relatives aux médicaments et aux antécédents nécessitait un algorithme de détection de concepts médicaux dans les documents textuels. Cet algorithme devait être suffisamment rapide et performant pour annoter les millions de documents textuels d'un EDS avec des terminologies médicales et sans annotation préalable. Le chapitre 4 présente IAMsystem, un algorithme d'annotation sémantique adapté aux contraintes d'un EDS et capable de prendre en compte le dictionnaire d'abréviations présenté au chapitre 5.

Le chapitre 6 présente une interface graphique, SmartCRF, facilitant l'extraction

d'information par interactions homme-machine. De nombreuses informations sont difficiles à extraire ou nécessitent des algorithmes spécifiques. L'utilisation en routine d'algorithmes de traitement automatique de la langue requière des efforts importants pour annoter des données, développer des algorithmes, les évaluer et les déployer. Leurs performances ne sont pas toujours suffisantes pour remplacer un expert humain. En routine, les chercheurs ont besoin de disposer d'une interface interactive pour faciliter la revue des dossiers et l'extraction d'information. L'objectif de SmartCRF était de remplacer la revue manuelle des dossiers dans DxCare®, très chronophage, par une revue assistée par ordinateur.

La figure 1.4 montre les liens entre ces chapitres.

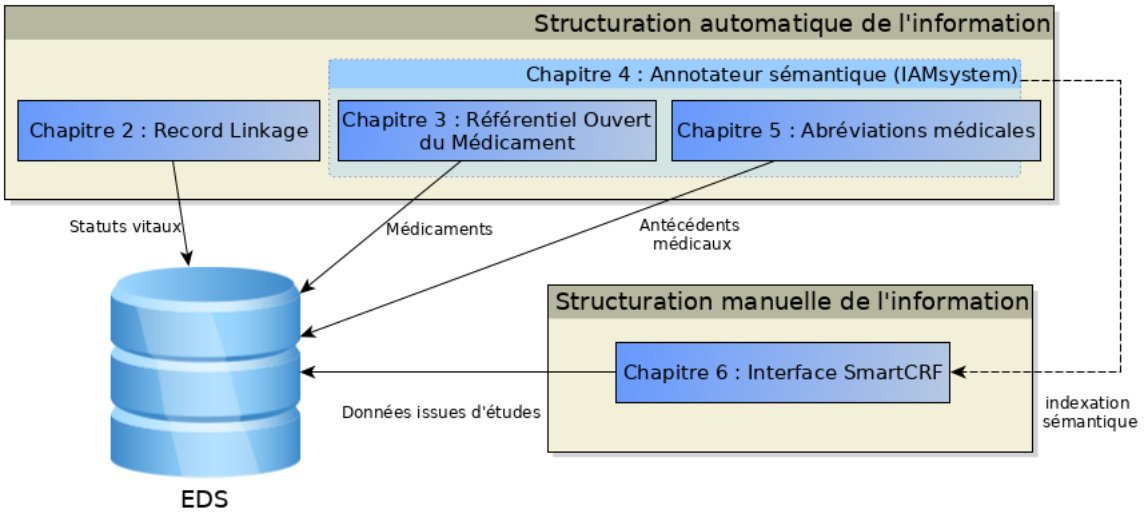


Figure 1.4: Liens entre les chapitres. Les chapitres 2 à 5 s'intéressent à la structuration automatique d'informations sur les statuts vitaux, les médicaments et les antécédents médicaux. L'annotateur sémantique IAMsystem présenté au chapitre 4 structure les informations de documents textuels avec des terminologies comme Romedi présentée au chapitre 3 et tire parti du dictionnaire d'abréviations créé au chapitre 5. L'interface SmartCRF présentée au chapitre 6 utilise IAMsystem pour détecter les concepts médicaux d'un document.

Chapitre 2

Appariements des données hospitalières aux certificats de décès

Ce chapitre est indépendant des suivants car les données traitées sont administratives et non médicales. La structuration du statut vital par appariement avec une source de données externe est très importante pour les études de faisabilité et les études rétrospectives menées sur l'EDS. Ce chapitre montre que les spécificités des données hospitalières offrent la possibilité de développer de nouvelles stratégies d'appariement.

2.1 Introduction

Le statut vital des individus est d'une importance capitale pour de nombreuses études épidémiologiques [17]. Dans les hôpitaux, les études rétrospectives basées sur les données cliniques nécessitent souvent de contacter le patient ou sa famille pour connaître le statut vital [18]. Les études de faisabilité consistant à dénombrer le nombre de patients éligibles à une étude ont aussi besoin de connaître le statut vital. Cependant, les informations relatives au décès ne sont enregistrées dans le système d'information hospitalier (SIH) que si le décès est survenu à l'hôpital. Lorsque le décès survient en dehors de l'hôpital ces informations sont souvent manquantes [19]. Un certain nombre de pays ont mis en place des registres d'état civil à partir desquels les chercheurs peuvent obtenir des informations sur le statut vital [19, 20, 21]. Par exemple, aux États-Unis, deux administrations fournissent des informations sur la mortalité :

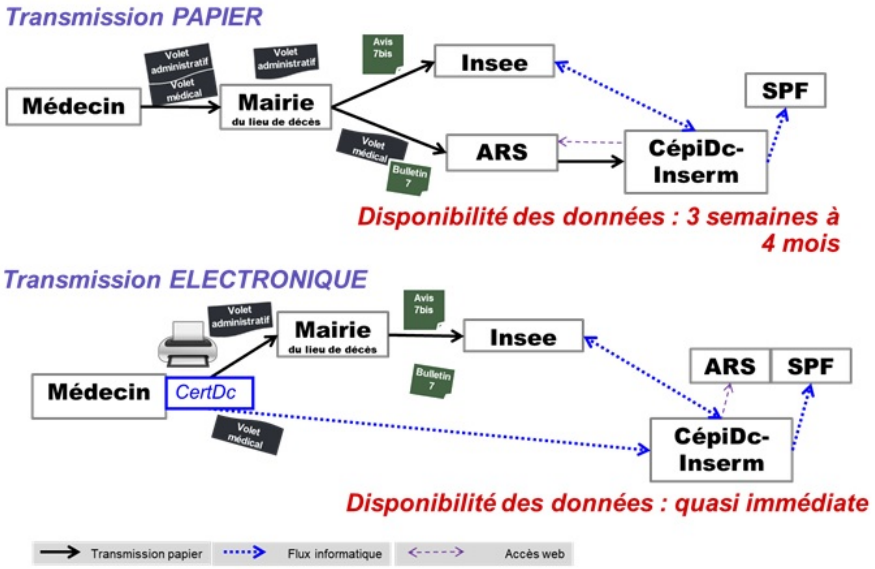


Figure 2.1: Le circuit administratif du certificat de décès en France. L'Insee enregistre les informations d'état civil. Des informations pseudonymisées et la cause de décès sont stockées par le CépiDc.

le Death Master File (DMF) de la Social Security Administration et le National Death Index (NDI) des Centers for Disease Control qui contient la cause du décès [19].

En France, les statistiques de l'état civil et les actes de décès sont enregistrés par l'Institut national de la statistique et des études économiques (Insee). Les actes de naissance et de décès sont transmis à l'Insee par les communes françaises. Le circuit administratif[1] français du certificat de décès est présenté figure 2.1. L'acte de décès étant obligatoire pour l'inhumation, l'information sur les décès peut être considérée comme exhaustive en France.

Le 5 décembre 2019, le département Etalab de la direction interministérielle du numérique a mis à disposition du public en open data les certificats de décès établis par l'Insee depuis 1970. Ce jeu de données pourrait permettre de compléter les informations relatives au statut vital des patients dans un SIH.

Dans ce contexte, notre objectif était de compléter les informations sur le statut vital en identifiant les décès extra-hospitaliers à l'aide de cette base de données française publique sur les décès. Une nouvelle stratégie d'appariement de données *(record linkage)* a été employée pour relier les dossiers hospitaliers aux certificats de décès. Le record linkage vise à identifier avec précision si deux ou plusieurs enregistrements se rapportent au même individu dans la

[1] `https://www.cepidc.inserm.fr/le-circuit-administratif-du-certificat-de-deces`

même base de données ou dans plusieurs bases de données. Dans ce chapitre, nous montrons la spécificité des données hospitalières pour le record linkage et présentons l'approche utilisée et les résultats obtenus. La prochaine section présente les grandes stratégies de record linkage et les travaux similaires. La section "Méthodes" présente la pipeline utilisée au CHU de Bordeaux. La section "Résultats" présente les résultats de cette stratégie et ses performances.

2.2 Etat de l'art

2.2.1 Record linkage

Il existe deux principales stratégies de record linkage : déterministe et probabiliste. La première est basée sur des règles et la seconde sur des pondérations ou des scores de variables. La méthode déterministe consiste à effectuer des liens sur la base d'une concordance exacte en utilisant plusieurs variables de correspondance. Les enregistrements sont comparés à l'aide d'un ensemble d'une ou plusieurs variables de correspondance (aussi appelés identifiants) qui sont communes aux deux enregistrements à comparer [22]. Le record linkage probabiliste consiste à calculer un score ou une probabilité entre deux enregistrements. Les paires avec un score élevé ont des probabilités plus élevées d'être des liens corrects. Dans les approches probabilistes, deux seuils sont généralement choisis (figure 2.2). Les paires dont le poids est supérieur au seuil supérieur sont classées comme des liens ; les paires dont le poids est inférieur au seuil inférieur sont classées comme des non-liens ; et celles qui se trouvent au milieu doivent être validées manuellement [23, 24].

Pour réduire le nombre de comparaisons entre les sources de données, les stratégies de blocage déterminent quels enregistrements sont des correspondances potentielles [23, 25]. Par exemple, le blocage sur une région géographique particulière impliquerait que les paires d'enregistrements ne soient considérées comme des correspondances potentielles que si elles s'accordent sur cette localisation. Les méthodes déterministes sont peu coûteuses en termes de calcul par rapport aux méthodes probabilistes. Elles sont aussi plus faciles à mettre en œuvre mais peuvent ne pas atteindre des performances suffisantes [22, 23]. Les approches par apprentissage automatique (machine learning) ont également été utilisées pour le record linkage [26].

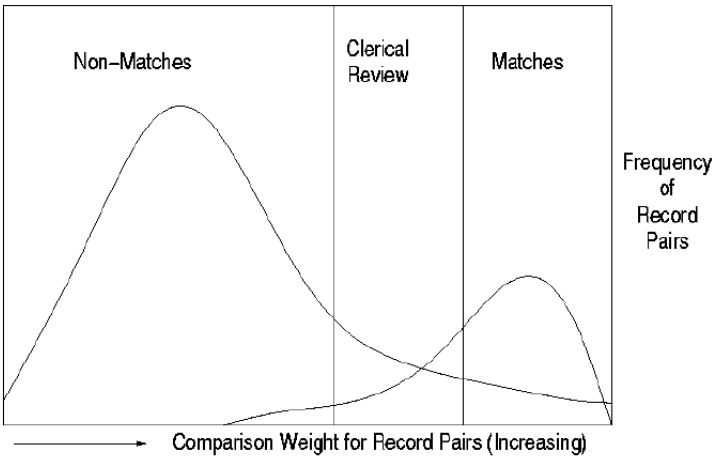


Figure 2.2: Dans une stratégie probabiliste de record linkage, deux seuils sont généralement définis pour les classer les paires potentielles. En dessous du seuil inférieur, une paire est classée comme un non-lien ; au dessus du seuil supérieur une paire est classée comme un lien ; entre les deux seuils la paire est classée comme indéterminée et doit être validée manuellement. D'après Gu et al. [25]

Les modèles d'apprentissage supervisés nécessitent des données d'entraînement pour lesquelles le statut de correspondance est connu [27]. En général, ce processus est chronophage et n'est pas réalisable pour les grandes sources de données administratives [28].

Les mesures d'évaluation courantes d'une méthode de record linkage sont le rappel (aussi appelé sensibilité) et la précision (aussi appelée valeur prédictive positive). Le rappel est le nombre de vraies correspondances correctement identifiées par un algorithme par rapport au nombre total de vraies correspondances dans le gold standard. La précision est le nombre de vraies correspondances correctement identifiées par rapport au nombre total de correspondances produites. D'autres critères de performance sont le temps de calcul et le nombre d'enregistrements nécessitant une revue manuelle [25]. Certaines études indiquent également la proportion d'enregistrements couplés, c'est-à-dire le taux de correspondance entre les deux bases de données.

2.2.2 Travaux similaires

Levin et al. [29] ont estimé la qualité des données d'un registre de décès avec les données de mortalité hospitalière comme gold standard. Dans ce but, le numéro de sécurité sociale a été utilisé et des étapes déterministes de record linkage ont été effectuées. Parmi les 2102 patients décédés à l'hôpital avant 2011, 1868 (88,9%) avaient un enregistrement de décès dans

le registre.

Newman et al. ont utilisé le logiciel commercial de record linkage Automatch™ pour identifier les patients dans la base de données d'un hôpital avec les fichiers de mortalité des registres de décès nationaux [30, 31]. Dans ce logiciel, l'utilisateur définit des variables de blocage qui doivent correspondre exactement dans les deux fichiers et des pondérations sont calculées dans chaque bloc pour un ensemble de variables de correspondance restantes. Les pondérations sont dérivées de deux probabilités conditionnelles introduites par Fellegi et Sunter : la probabilité m (probabilité qu'un identifiant soit identique lorsque qu'il s'agit du même individu) et la probabilité u (probabilité qu'un identifiant appartienne à des individus différents) [32]. Pour chaque paire d'enregistrements, la pondération globale est calculée en calculant le rapport $\log_2 \frac{m}{u}$ pour chaque identifiant puis en faisant la somme de tous les identifiants. Newman et al. ont atteint une sensibilité très élevée pour les décès de patients hospitalisés (99,3 %) lorsque les numéros de sécurité sociale étaient disponibles.

En 2017, le projet matchID[2], financé par le gouvernement français, visait à relier la base de données des permis de conduire français aux données de mortalité de l'Insee. Ce logiciel utilise Elasticsearch pour indexer les données de mortalité et rechercher les certificats de décès. Il comprend une interface web pour la validation manuelle et un module de machine learning.

2.3 Méthodes

2.3.1 Jeux de données

Le SIH du CHU de Bordeaux contient des informations détaillées sur toutes les admissions à l'hôpital de Bordeaux depuis 2005. La base de données hospitalière contient des informations administratives sur 2,2 millions de patients. Le décès est enregistré lorsqu'il survient à l'hôpital. Fin 2020, seulement 58 020 décès survenus en milieu hospitalier avaient été enregistrés. Les données de l'Insee contiennent des données sur la mortalité de plus de 25 millions de personnes décédées depuis 1970.

Les données publiées en open data sont des fichiers texte pour chaque année depuis 1970

[2] `https://matchid.io/about`

Données hospitalières	Certificats de décès	Comparaison
Nom de famille (de naissance et de famille) (0%)	Nom de famille (0%)	Distance de chaînes de caractères
Prénom (0%)	Prénoms (0%)	Distance de chaînes de caractères
Date de naissance (0%)	Date de naissance (0,66%)	Différence pour l'année, le mois et le jour
Sexe (0%)	Sexe (0%)	Egal ou non
Code postal de naissance (39%)	Code postal de naissance (0,54%)	Egal ou non
Ville de naissance (39%)	Ville de naissance (0,54%)	Egal ou non
Pays de naissance (6.9%)	Pays de naissance (0,02%)	Egal ou non
Dernière adresse postale connue (1,7%)	Département de décès (0,11%)	Egal ou non pour le département et la région
Date de la dernière visite (0%)	Date du décès (0%)	Différence en nombre de jours

Table 2.1: Attributs communs des données hospitalières et des certificats de décès de l'Insee. Les pourcentages de valeurs manquantes figurent entre parenthèses. Plusieurs méthodes ont été utilisées pour comparer les attributs d'un dossier hospitalier et d'un certificat de décès.

et des fichiers mensuels pour l'année en cours. Chaque fichier a le même format et contient les informations suivantes sur le décès de chaque citoyen français : nom de famille, prénoms, sexe, date et lieu de naissance, date et lieu de décès, numéro du certificat de décès. Ces données contiennent environ 8,8 millions actes de décès produits depuis 2005.

Le tableau 2.1 montre les attributs communs entre les deux sources de données. L'hôpital enregistre deux noms de famille : le nom de naissance et le nom de famille utilisé alors que l'Insee ne fournit qu'un seul nom de famille. Il existe également des différences concernant le prénom. Les Français ont un ou plusieurs prénoms. L'un d'entre eux est utilisé dans la vie quotidienne alors que les autres sont uniquement destinés aux documents officiels. L'hôpital n'enregistre que celui qui est utilisé dans la vie quotidienne alors que les actes de décès les contiennent tous. Les certificats de décès contiennent également le code postal et le département du décès et l'hôpital enregistre la dernière adresse du patient qui correspond parfois au lieu du décès. Aucun identifiant fort tel que le numéro de sécurité sociale n'était disponible pour relier les enregistrements.

Les mort-nés (date de naissance égale à date de décès) à l'hôpital ont été retirés car un certificat de naissance et un certificat de décès ne sont pas systématiquement établis dans ces situations.

2.3.2 Stratégie de record linkage

La figure 2.3 présente la pipeline développée dans cette étude.

Notre approche consistait à combiner une stratégie de blocage basée sur le modèle vectoriel de recherche d'information avec une stratégie de machine learning. Pour entraîner les algorithmes de machine learning, nous avons utilisé une approche déterministe aux données de 2005 à 2018 afin de générer automatiquement un gold standard. La pipeline a été évaluée sur l'année 2019, chacune de ses étapes est détaillée ci-dessous.

Étape 1 : Stratégie de blocage

Elasticsearch est un moteur de recherche distribué construit au-dessus d'Apache Lucene[3]. Le jeu de données open data de l'Insee a été indexé dans Elasticsearch version 7.6.1 après plusieurs étapes de normalisation : les accents et les diacritiques ont été supprimés, les dates en chaînes de caractères ont été transformées au format date, les départements de décès ont été extraits des valeurs du code postal et les années, mois et jours ont été extraits des dates de naissance pour créer de nouveaux champs. Pour trouver les certificats de décès candidats pour un enregistrement hospitalier donné, une requête était envoyée à Elasticsearch. Chaque requête comprenait le nom de famille, le prénom, le sexe, ainsi que le jour, le mois et l'année de naissance. Les résultats étant classés par leur score, les certificats de décès les plus pertinents apparaissaient très souvent en premier. Elasticsearch utilise la fonction de similarité de Lucene basée sur la métrique TF-IDF (*term frequency-inverse document frequency*). Un score par variable est calculé et le score global du certificat de décès correspond à la somme du score de chaque variable. Plus il y a de correspondance d'identifiants (prénom, nom de famille...), plus le score global est élevé et plus la valeur d'une variable est rare, plus le score de cette variable est élevé. Par exemple, le poids était de +7,15 pour un nom commun tel que DUPONT et de +14,24 pour les noms très rares. Chaque requête renvoie un nombre prédéterminé de résultats, c'est-à-dire des certificats de décès. A cette étape, un petit nombre de résultats pourrait diminuer le rappel et un nombre élevé pourrait diminuer la précision et ralentir la pipeline. Le temps calcul d'une stratégie de record linkage est dominé par le nombre de comparaisons effectuées [25]. Le

[3] `http://www.apache.lucene.org`

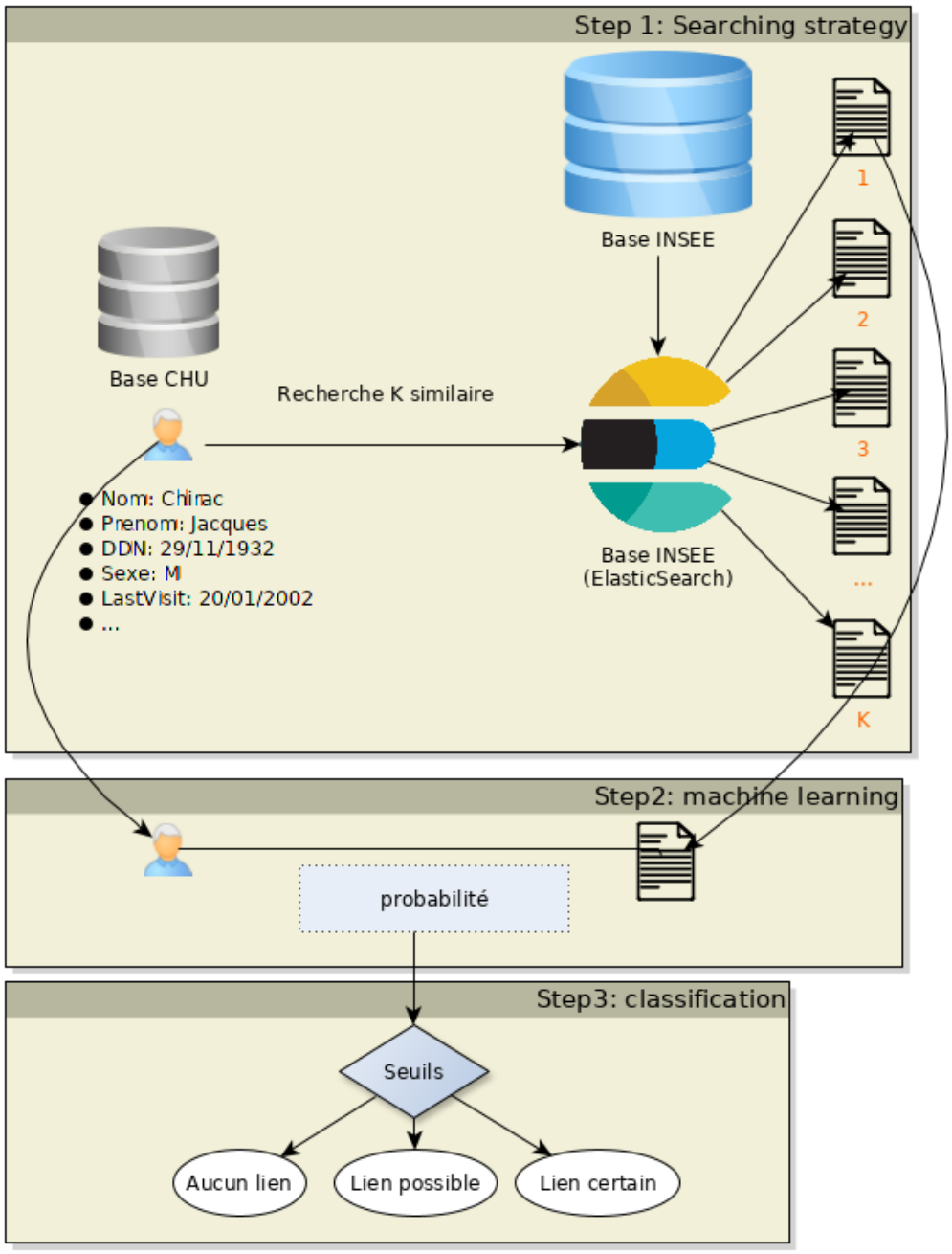


Figure 2.3: Vue d'ensemble de la stratégie de record linkage. Dans la première étape, une requête est envoyée à Elasticsearch pour chaque enregistrement hospitalier afin de récupérer un nombre limité k de certificats de décès candidats. Dans la deuxième étape, des modèles d'apprentissage machine prédisent la probabilité de correspondance entre un enregistrement hospitalier et un certificat candidat. Dans la troisième étape, la paire est classée comme non liée, indéterminée ou liée en fonction des seuils supérieur et inférieur

nombre optimal de résultats k a été défini à l'étape 2, en utilisant le gold standard.

Étape 2 : Machine Learning

Gold standard L'objectif de l'étape machine learning est de comparer un enregistrement hospitalier avec un certificat de décès candidat et de produire une probabilité de correspondance. Pour entraîner un algorithme de machine learning, il faut créer un gold standard contenant à la fois des correspondances vraies et fausses. Une vraie correspondance est un enregistrement hospitalier correctement lié à un certificat de décès et une fausse correspondance est une paire d'enregistrements incorrectement liés. La difficulté de la construction d'un gold standard est d'identifier des vraies correspondances, tâche souvent manuelle et chronophage, et de trouver des fausses correspondances dans la zone grise [33], où les paires d'enregistrements n'ont pas une concordance exacte, pour que l'algorithme apprenne l'importance de chaque variable.

Nous avons utilisé une approche déterministe pour trouver les vraies correspondances. Pour les identifier correctement, nous avons tiré profit des décès intra-hospitaliers. Ces décès ont deux variables de correspondance supplémentaires : la date et le département du décès. Sept champs communs étaient disponibles pour relier les décès intra-hospitaliers : nom, prénom, date de naissance, code postal, sexe, date de décès et département de décès. La première étape de l'approche déterministe a consisté à apparier avec un accord parfait sur ces sept champs. En gardant la date et le département de décès comme des variables de blocage, la stratégie déterministe a été légèrement assouplie pour autoriser une différence dans l'un des champs restants. L'examen manuel des enregistrements lorsqu'une seule variable ne correspondait pas nous a permis de nous assurer que les différences étaient dues à des erreurs dans l'une des deux sources de données et qu'aucun enregistrement n'était faussement lié.

Ces paires de correspondances vraies ont ensuite été utilisées pour identifier les correspondances incorrectes. Pour chaque correspondance vraie, une requête a été envoyée à Elasticsearch et la correspondance incorrecte ayant le score le plus élevé a été retenue. Le tableau 2.2 présente un exemple de sélection d'un faux alignement.

En sélectionnant un seul faux alignement pour chaque bon alignement, nous avons généré un ensemble de données équilibré, c'est-à-dire qu'il y avait autant d'exemples positifs que négatifs. Cet équilibre est important pour éviter la sur-représentation d'une classe qui peut conduire à un

	Vrai	Faux n°1	Faux n°x
Nom	Chirac	Aupetit	Pompidou
Prénoms	Jacques Rene	Jacques	Georges
Date de naissance	1932-11-29	1932-11-29	05/07/1911
Sexe	Homme	Homme	Homme
Lieu de naissance	Paris 5	Paris 12	Montboudif
Pays de naissance	France	France	France

Table 2.2: Exemple fictif pour la sélection de faux appariements. Une requête est envoyée à Elasticsearch avec les données hospitalières. Parmi ces trois certificats de décès, le bon certificat de décès ayant déjà été identifié avec l'appariement déterministe, le meilleur faux appariement est le plus proche selon la mesure de similarité d'Elasticsearch basée sur le TF-IDF. Dans cet exemple, le faux n°1 est assez proche du vrai certificat tandis qu'un autre certificat pris aléatoirement présentera de nombreuses différences avec le bon appariement.

biais d'apprentissage vers la classe majoritaire [34].

Le résultat de la recherche Elasticsearch d'un décès intra hospitalier permettait aussi d'enregistrer le rang du bon certificat de décès. Dans la très grande majorité des cas, le bon certificat de décès avait le rang 1 dans les résultats d'Elasticsearch. La distribution de cette variable a permis de choisir le nombre optimal de résultats k de l'étape 1.

Features Le gold standard a été converti en une matrice de caractéristiques pour entraîner les algorithmes de machine learning. Une caractéristique est une fonction qui prend une paire d'identifiants et produit un nombre réel. Les caractéristiques binaires ont été générées en vérifiant simplement l'égalité des valeurs (1 si les valeurs sont les mêmes, 0 sinon) entre un enregistrement hospitalier et un certificat de décès. Plusieurs méthodes de distance entre chaînes de caractères basées sur les éditions (Damerau-Levenshtein, Hamming, Levenshtein, alignement optimal des chaînes de caractères), les qgrammes (q-gram, cosinus, distance Jaccard), la phonétique (soundex) et les métriques heuristiques (Jaro, Jaro-Winkler) ont été calculées pour le prénom et le nom de famille avec le paquet R stringdist [35]. Le score global Elasticsearch a également été ajouté à la matrice des caractéristiques. Au total, 40 caractéristiques ont été calculées. Un modèle forêts aléatoires et un réseau de neurones ont été entraînés pour prédire si une paire d'enregistrements était une vraie ou une fausse correspondance. Le gold standard a été divisé en un jeu d'entraînement, un jeu de validation et un jeu de test selon un ratio de 60:20:20. Le jeu d'entraînement a été utilisé pour entraîner les modèles, le jeu de validation

pour comparer des modèles avec différents hyperparamètres, et le jeu de test a été utilisé une seule fois pour évaluer les performances des modèles.

Étape 3 : Seuil

Un seuil supérieur et un seuil inférieur ont été fixés à la fin de l'évaluation de la pipeline pour maximiser la précision et le rappel, respectivement. Les paires d'enregistrements dont les probabilités étaient supérieures au seuil supérieur ont été classées automatiquement comme des liens, les paires dont les probabilités étaient inférieures au seuil inférieur comme des non-liens, et celles qui se situaient au milieu comme indéterminées.

2.3.3 Evaluation

Notre stratégie globale de record linkage a été évaluée en constituant un fichier dans lequel le statut de décès des individus inclus était connu au préalable. Les décès hospitaliers de 2019 ont été inclus pour évaluer la sensibilité. La date de décès a été utilisée pour valider automatiquement si deux enregistrements étaient de vrais positifs. En cas de différence entre les deux dates de décès, un examen manuel était effectué pour vérifier s'il s'agissait d'une erreur de classification ou d'une erreur de source de données. La spécificité a été évaluée en sélectionnant de manière aléatoire 15 000 patients ayant fréquenté l'hôpital en 2020, c'est-à-dire des patients connus pour ne pas être décédés en 2019.

2.3.4 Description formelle de la pipeline

Soit H et I l'ensemble des enregistrements hospitaliers et des certificats de décès de l'Insee, respectivement. Il existe une relation $R_{deces} \subset H \times I$ liant les informations d'un même individu. L'objectif est d'identifier les éléments de cette relation.

La relation R_{deces} est partitionnable : $R_{deces} = R_{deces\_intra} \cup R_{deces\_extra}$
et $R_{deces\_intra} \cap R_{deces\_extra} = \varnothing$ avec R_{deces_intra}, R_{deces_extra} les relations des décès intra-hosptialiers et extra-hospitaliers, respectivement. Les éléments de la relation R_{deces_intra} sont plus faciles à identifier que ceux de la relation R_{deces_extra}. Un patient décédé à l'hôpital a deux éléments d'informations supplémentaires : la date de décès et le département de décès. Les

éléments de la relation R_{deces_intra} peuvent être trouvés par approche déterministe en utilisant des attributs fortement identifiants, notamment la date de décès qui est aussi importante que la date de naissance. Les éléments de R_{deces_intra} constituent les exemples positifs du gold standard. Pour chaque élément $(h, i) \in R_{deces\_intra}$, on recherche avec Elasticsearch un autre certificat de décès i' pour créer un exemple négatif (h,i')[4]. L'ensemble des couples (h,i') forment les exemples négatifs du gold standard.

Une fonction de classification $f : H \times I \rightarrow \mathcal{R}$ prédit une probabilité $p \in [0, 1]$ qu'une paire (h,i) appartienne ou non à R_{deces_intra}. Dans notre approche, cette fonction de classification combine deux algorithmes de machine learning. Pour apprendre, un algorithme de machine learning a besoin d'une matrice X de features (caractéristiques) et d'un vecteur Y de valeurs à prédire. Dans ce but, chaque paire (h,i) est vectorisée en un vecteur de features $x_1, ..., x_n$ par la fonction $vectorize : H \times I \rightarrow R^n$. Chaque élément x_i de ce vecteur est produit par une fonction de similarité g_i qui compare un attribut de h (enregistrement hospitalier) à un attribut de i (certificat de décès). Par exemple, l'élément x_3 est calculé par la fonction Damerau-Levenshtein qui compare la distance entre le nom de famille de l'enregistrement hospitalier et celui du certificat de décès. Chaque ligne de la matrice X correspond à la vectorisation d'un couple (h,i) ou (h,i') du gold standard et chaque colonne à une fonction de similarité. La dimension de la matrice X est égale à ($2 \times |R_{deces\_intra}|$,40) car pour chaque décès hospitalier un exemple négatif est recherché et 40 fonctions de similarité ont été utilisées. La longueur du vecteur Y est égale au nombre de ligne de la matrice X. Un élément de vecteur Y vaut 1 pour le couple (h,i) et 0 pour le couple (h,i').

La fonction de classification f est ensuite utilisée pour identifier les éléments de R_{deces_extra}. La cardinalité du produit cartésien $H \times I$ est très grande dans un SIH, de l'ordre de 10^{13} éléments au CHU de Bordeaux. Il est inenvisageable de comparer chaque enregistrement hospitalier avec chaque certificat de décès. La stratégie de blocage consiste à limiter le nombre de comparaisons à un sous-ensemble $R_{blocage} \subset H \times I$. Une stratégie de blocage diminue le nombre de comparaisons au prix d'une diminution de la sensibilité. La sensibilité d'une stratégie de blocage

[4]Cette étape est réalisée après avoir déterminé la requête optimale.

est:

$$Se = \frac{|R_{deces} \cap R_{blocage}|}{|R_{deces}|}$$

La stratégie de blocage de notre approche est basée sur le modèle d'espace vectoriel de Lucene. Ce modèle calcule une similarité basée sur la métrique TF-IDF pour une paire (q,d) où *q* est une requête et *d* un document. Dans notre contexte, *d* correspond à un certificat de décès et *q* est une requête créée à partir d'un enregistrement hospitalier. De nombreuses requêtes peuvent être créées à partir d'un enregistrement hospitalier. Par exemple, il est possible de rechercher la date de naissance complète ou le jour, le mois et l'année de naissance séparément. Plusieurs requêtes différentes ont été testées. Pour chaque requête, le rang de chaque certificat de décès i, avec $(h, i) \in R_{deces\_intra}$, dans les résultats a été enregistré. Une distribution du rang des certificats de décès est ainsi obtenue pour chaque requête. La sensibilité dépend à la fois de la requête *q* et du nombre de résultats *k* choisis :

$$Se(k, q) = \frac{|R_{deces\_intra} \cap R_{blocage\_q_k}|}{|R_{deces}|}$$

Pour un nombre de résultats k sélectionné, on retient la requête qui maximise la sensibilité : $Se(k) = max(\bigcup_{i=1}^{n} Se(k, q_i))$. Le nombre k a été fixé empiriquement, il détermine la sensbilité de la stratégie de blocage mais aussi le nombre total de comparaisons réalisées ; par exemple une valeur à 5 plutôt qu'à 10 permettrait de diminuer par 2 le temps total de la pipeline. La sensibilité d'une requête en fonction de k, Se(k), est donnée figure 2.4.

Le nombre de résultats k a été fixé à 10. Pour ce nombre, la sensibilité de la méthode de blocage a été estimée à 99,783%, c'est-à-dire qu'un certificat de décès de R_{deces_intra} était présent dans les 10 premiers résultats d'Elasticsearch dans 99,783% des cas. Cette stratégie de blocage diminue le nombre de comparaisons de 10^{13} à 22 millions (2,2 million d'enregistrements x 10 résultats) pour une perte minime de sensibilité.

La pipeline complète a été évaluée en créant un gold standard $R_{evaluation}$ contenant des décès intrahospitaliers de 2019 $R_{deces\_intra\_2019}$, non présents dans les données d'entraînement, et des patients vivants en 2020 (donc en 2019) $R_{vivants\_2019}$ pour mesurer la sensibilité et la spécificité, respectivement. Cette évaluation a permis de fixer un seuil supérieur de 0.95 et un seuil inférieur de 0.4 ; le premier maximise la spécificité et le second la sensibilité sur ce jeu

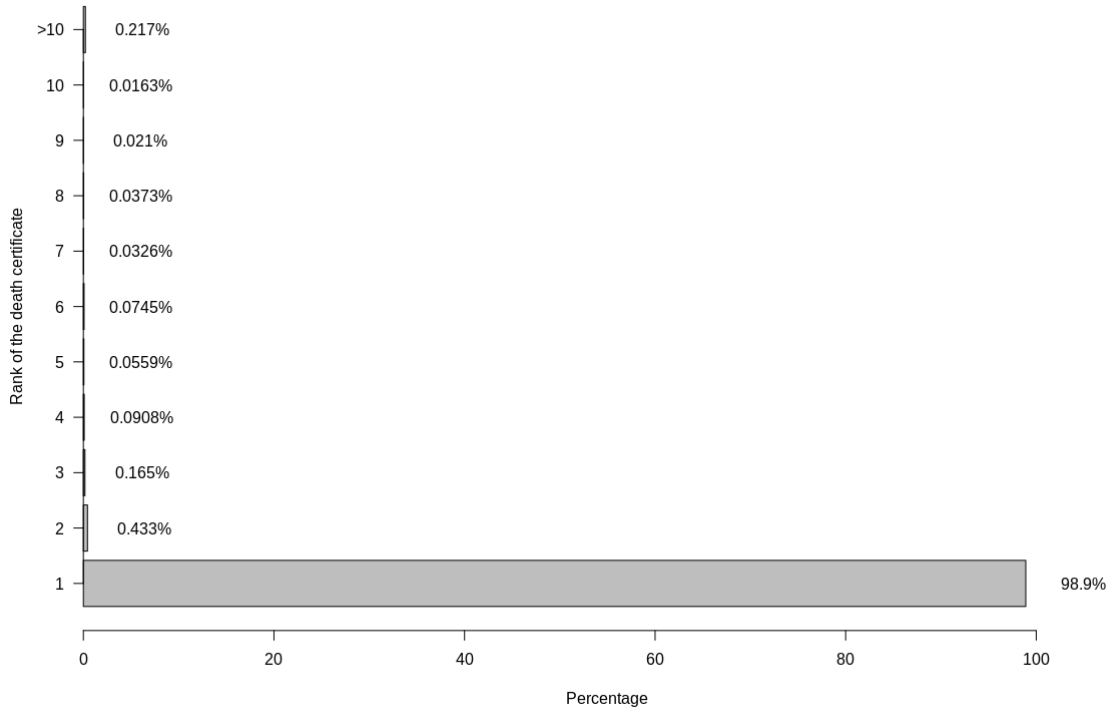


Figure 2.4: Mesure de la sensibilité de la stratégie de blocage en fonction du nombre de résultats k. Le rang du certificat de décès indique le gain de sensibilité. Pour calculer Se(k), il faut additionner les valeurs de 1 à k.

d'évaluation.

Pour chaque élément (h,i) des 22 millions d'éléments de $R_{blocage}$, la fonction de classification f prédit une probabilité d'appariement, c'est-à-dire la probabilité que cet élément appartienne à R_{deces}. L'ensemble $R_{blocage}$ a été finalement divisé en trois sous-ensembles :

$R_{blocage\_links} = \{(h,i) \mid f(h,i) \geq 0.95, (h,i) \in R_{blocage}\}$

$R_{blocage\_undetermined} = \{(h,i) \mid 0.4 \leq f(h,i) < 0.95, (h,i) \in R_{blocage}\}$

$R_{blocage\_no\_links} = \{(h,i) \mid f(h,i) < 0.4, (h,i) \in R_{blocage}\}$

2.4 Résultats

2.4.1 Apprentissage supervisé

Entre 2005 et 2018, et après avoir retiré les mort-nés, 44 689 décès hospitaliers ont été enregistrés dans le système d'information du CHU de Bordeaux. Avec l'approche déterministe, 44 127 (98,7%) décès ont été reliés avec succès à un certificat de décès. Parmi eux, 39 273 (89%)

avaient une concordance exacte sur les sept champs et 4 854 (11%) avaient une différence dans un champ :

- 3,8% le nom de famille était différent
- 3,1% le département de naissance était différent
- 2,7% le prénom était différent
- 1,3% la date de naissance était différente
- 0.1% le sexe était différent

La divergence la plus fréquente était une différence dans le nom de famille (3,8%) et l'erreur la moins fréquente était le sexe (0,1% d'erreurs). Il est intéressant de noter qu'après un examen manuel, la plupart des erreurs concernant le sexe se trouvaient dans les données de l'Insee.

La grande majorité des décès intra-hospitaliers (89%) avait une correspondance parfaite avec leur certificat de décès sur les sept identifiants recherchés. Un sous-échantillonnage du nombre de correspondances parfaites a été réalisé pour avoir un ratio de 50% de correspondances parfaites et imparfaites sur le jeu de données. Cet échantillonnage a été réalisé pour sur-représenter les différences dans certains identifiants et permettre aux algorithmes de machine learning d'apprendre l'importance de chaque identifiant.

Le jeu de données pour l'apprentissage supervisé contenait un total de 16 460 paires d'enregistrements. Sur l'ensemble de test (n = 3 294 paires d'enregistrements), le nombre d'erreurs était de 34 (17 faux positifs, 17 faux négatifs) et de 35 (21 faux positifs, 14 faux négatifs) pour le modèle de forêts aléatoires et le modèle d'apprentissage profond, respectivement. Ces deux algorithmes avaient donc des performances comparables pour prédire si un enregistrement hospitalier et un certificat de décès était lié ou non.

2.4.2 Nombre de résultats k

Sur les 44 127 vraies correspondances décrites ci-dessus, 98,9% ont été classées en premier par Elasticsearch et seulement 0,2% ont été classées après le 10e résultat (figure 2.4). Une diminution de la sensibilité de 0,2% à cette étape de la pipeline a été jugée acceptable et la valeur de k a été fixée à 10.

2.4.3 Évaluation de la stratégie de record linkage

En 2019, 3 565 décès en milieu hospitalier ont été enregistrés et 15 000 patients ayant fréquenté l'hôpital en 2020 ont été sélectionnés au hasard pour mesurer la précision. Pour chaque enregistrement, les 10 premiers certificats de décès ont été récupérés avec Elasticsearch et classés avec les modèles de machine learning.

Seuls 20 décès en milieu hospitalier (0,5%) ne figuraient pas dans les 10 premiers résultats d'Elasticsearch. Après un examen manuel et l'utilisation des dates de décès, 19 ont été trouvés. Un enregistrement hospitalier n'a pas pu être lié manuellement car le nom et le prénom réels avaient été remplacés pour des raisons d'anonymat. Les principales raisons de l'absence de certificats de décès dans les résultats de recherche étaient les inversions de nom et de prénom et les prénoms transformés pour les rendre culturellement français, par exemple Maria en Marie. Les 3 565 décès survenus à l'hôpital en 2019 ont tous été liés avec succès à un certificat de décès, ce qui montre l'exhaustivité des données open data publiées. Les dates de décès étaient différentes pour 71 paires d'enregistrements (1,9%) et 11 étaient différentes de plus de 30 jours. Ces 71 paires ont été vérifiées manuellement et certaines inversions entre le jour et le mois ont été détectées. Sur la base des prédictions du modèle, le seuil supérieur a été fixé à 0,95 pour maximiser la précision et le seuil inférieur à 0,4 pour maximiser le rappel. Si les probabilités du réseau de neurone et des forêts aléatoires étaient supérieures à 0,95, la paire était classée comme lien, non classée si les deux probabilités étaient supérieures à 0,4 mais inférieures à 0,95, et comme non-liens si l'une des deux probabilités était inférieure à 0,4. Le rappel et la précision étaient respectivement de 97,5% et 99,97% pour le seuil supérieur et de 99,4% et 98,9% pour le seuil inférieur. Avec le choix de ces seuils, un seul faux positif survenant toujours avec le seuil supérieur, en raison de frères jumeaux qui partageaient le même premier prénom. Seuls deux décès à l'hôpital se situaient sous le seuil inférieur et étaient presque impossibles à prédire correctement sans connaître la date du décès en raison de différences dans le nom de famille et de données manquantes pour le lieu de naissance.

2.4.4 Estimation des décès extra-hospitaliers

En appliquant la pipeline aux 2,2 millions d'enregistrements hospitaliers, 207 507 enregistrements ont été liés à un certificat de décès avec une probabilité supérieure au seuil supérieur et 29 152 avaient des probabilités comprises entre les deux seuils, nécessitant une validation manuelle. En comparaison, une requête de correspondance exacte dans une base de données relationnelle sur le nom, le prénom, la date de naissance et le sexe n'a permis de trouver que 200 824 paires d'enregistrements : 195 465 étaient au-dessus du seuil supérieur (94%), 3 885 étaient entre les deux seuils et 1 474 étaient en dessous du seuil inférieur. Ces derniers correspondaient à des homonymes que l'algorithme réussit à différencier en utilisant d'autres identifiants.

En termes de performances, il a fallu environ 4 minutes et 30 secondes pour rechercher et classer 1 000 dossiers hospitaliers et environ 3 jours pour un million de dossiers sur une machine virtuelle avec Intel Core i7-8650U @1.90GH x 8 CPU sans parallélisation. Le code source, y compris la matrice et les modèles pré-entraînés sont disponibles.

2.5 Discussion

Dans ce chapitre, nous avons démontré la faisabilité d'un appariement de données hospitalières avec les données de certificats de décès de l'Insee par une nouvelle stratégie combinant moteur de recherche et apprentissage supervisé.

Une originalité de la pipeline proposée est de créer un gold standard automatiquement, sans annotation manuelle. L'annotation manuelle est coûteuse en temps et peut être source d'erreurs [25]. Le gold standard a été créé par une approche déterministe qui a utilisé notamment la date de décès des personnes décédées à l'hôpital. Cette étape de la pipeline est spécifique aux données hospitalières. Utiliser la date de décès dans une approche déterministe avait également été employé par Newman et al. pour vérifier l'exactitude d'autres variables [30]. Combinée à d'autres variables, la date de décès est hautement discriminante et la probabilité de lier faussement deux enregistrements est faible.

Une autre originalité est d'utiliser un espace vectoriel pour la méthode de blocage. Cette méthode avait déjà été utilisée mais n'était pas mentionnée dans des articles scientifiques. Une

étude brésilienne de 2020 a également appliqué une méthode de blocage basée sur Lucene pour réduire le nombre de comparaisons dans les étapes suivantes [36]. Les auteurs revendiquent l'originalité dans l'utilisation de Lucene. Basé sur Lucene, Elasticsearch est scalable et rapide ce qui permet d'indexer de grands volumes de données, comme les données de l'Insee. La méthode de blocage basée sur son score de pertinence a donné des résultats satisfaisants : un certificat de décès, s'il existait, apparaissait dans 99,8% des cas dans les 10 premiers résultats. L'approche proposée présente des similitudes avec la procédure probabiliste de Fellegi-Sunter. La probabilité qu'un identifiant soit identique pour des individus différents (u-probabilité) est faible pour les valeurs peu fréquentes, ce qui se traduit par un rapport $\log_2 \frac{m}{u}$ plus élevé. Dans le modèle vectoriel proposé par Lucene, les valeurs rares ont un IDF (inverse document frequency) élevé, ce qui conduit à augmenter la valeur du TF-IDF et le score global du certificat de décès. Le score de pertinence d'Elasticsearch est la somme globale du poids de chaque identifiant, comme dans l'approche probabiliste [31, 37]. Le modèle Fellegi-Sunter intègre une hypothèse d'indépendance de chaque variable, alors que les méthodes de machine learning comme les forêts aléatoires et les réseaux neuronaux ne le font pas. Les modèles de machine learning peuvent également tirer parti de caractéristiques de plus haut niveau, telles que les distances entre chaînes de caractères et les différences de temps entre la date de la dernière visite et la date du décès [38].

Par rapport à une approche déterministe basée sur la concordance exacte du nom, du prénom, de la date de naissance et du sexe, l'approche proposée a amélioré le rappel d'au moins 6,2 %. La spécificité pourrait également avoir été améliorée car quelques paires d'enregistrements (n = 1 474 ; 0,7 %) détectées par l'approche déterministe ont été classées comme non liées par les algorithmes de machine learning.

Les décès en milieu hospitalier sont fréquemment utilisés pour évaluer la fiabilité des données sur les décès [29]. Le taux d'appariement évalué en 2019 était de 100%, ce qui montre l'exhaustivité des données open data publiées. Les données mises en ligne apparaissent être une source fiable, bien que la qualité des données soit imparfaite. Plusieurs erreurs ont été détectées dans les sexes et les dates de décès, par exemple.

Le jeu d'entraînement et les modèles entraînés ont été publiés car il est impossible de ré-identifier une personne à partir de la matrice de caractéristiques. Ces résultats suggèrent que ce

travail pourrait être répliqué facilement par d'autres hôpitaux français.

Les informations sur la mortalité contenues dans le système d'information du CHU de Bordeaux se sont révélées incomplètes. Ceci n'est pas surprenant car seuls les décès des patients hospitalisés étaient enregistrés. Nous avons trouvé un rapport de 1:4 entre les décès intra-hospitaliers et extra-hospitaliers. Les décès extra-hospitaliers sont des informations importantes pour les études de recherche clinique menées dans les hôpitaux. Le statut vital est également essentiel pour les systèmes d'information hospitalier qui doivent faire face à la quantité toujours croissante de données des dossiers médicaux électroniques, ce qui menace de ralentir les applications de soins. Une façon de diminuer la taille d'une base de données est d'archiver les données qui ne sont plus utiles, par exemple celles des patients décédés.

2.5.1 Limites de l'approche

Cette étude présente plusieurs limites. Premièrement, le gold standard a été créé automatiquement par une approche déterministe et la différence maximale entre deux enregistrements était limitée à un seul identifiant. Aucun exemple de paires d'enregistrements présentant deux différences ou plus n'était présent dans le gold standard, ce qui pourrait biaiser la prédiction d'un modèle de machine learning.

De plus, le gold standard pouvait contenir des erreurs qui n'ont pas été identifiées lors de l'examen manuel rapide. Un gold standard de meilleure qualité aurait pu être obtenu en utilisant un identifiant commun aux deux sources de données, tel que le numéro de sécurité sociale. Ce numéro n'est pas accessible au public et nécessite des autorisations difficiles à obtenir [39].

Deuxièmement, les prénoms dans diverses langues n'ont pas été pris en compte, ce qui a diminué le rappel. Comme Elasticsearch offre la fonctionnalité d'ajouter des synonymes, il serait envisageable d'élargir les requêtes de recherche en fournissant une ressource de prénoms équivalents. Par exemple, Wiktionary, un projet Web visant à créer un dictionnaire de termes à contenu libre, fournit les traductions des prénoms masculins et féminins dans plusieurs langues.

Troisièmement, il s'agissait d'une étude monocentrique. L'excellente concordance des décès en milieu hospitalier avec les certificats de décès ne peut pas être généralisée à d'autres régions françaises. Le nombre optimal de certificats retourné par Elasticsearch pourrait être

plus élevé en Ile-de-France où le volume de certificats de décès est plus élevé. Nous faisons l'hypothèse que notre approche fonctionne mieux dans les régions peu peuplées car moins une commune est peuplée, plus elle est discriminante pour rapprocher deux enregistrements.

2.5.2 Perspectives

L'ajout d'informations cliniques comme caractéristiques pourrait améliorer les modèles de machine learning. Par exemple, une condition clinique grave telle qu'un cancer du pancréas métastatique rendrait le décès probable moins d'un an après le diagnostic. La date prévue du décès pourrait être une caractéristique à ajouter, mais elle serait complexe à calculer à partir des données des dossiers médicaux électroniques.

2.6 Conclusion

La pipeline d'appariement proposée basée sur le moteur de recherche Elasticsearch et une stratégie de machine learning fournit des résultats satisfaisants et pourrait être encore améliorée. Le grand volume des données hospitalières a nécessité un méthode de blocage efficace. Les décès intra-hospitaliers ont permis de créer un gold standard. Les informations sur la mortalité dans notre SIH étaient incomplètes car seuls les décès en milieu hospitalier étaient enregistrés. Ces résultats peuvent être reproduits dans les autres hôpitaux français pour mettre à jour le statut vital.

Le chapitre introductif a expliqué l'importance de configurer les algorithmes avec une excellente précision. La machine permet un gain de temps aux utilisateurs, si et seulement si, les informations sont structurées avec exactitude. La présence d'informations erronées obligerait l'expert humain à vérifier chaque information structurée par l'algorithme, ce qui n'apporterait aucun bénéfice. Ce principe est utilisé dans le domaine du record linkage où un seuil supérieur et un seuil inférieur sont généralement définis. En pratique, le statut vital d'un individu et sa date de décès peuvent être réutilisés automatiquement si la probabilité d'appariements est au-dessus du seuil supérieur, c'est-à-dire si la précision de l'algorithme est proche de 100%.

Chapitre 3

Intégration et extraction d'information sur les médicaments

3.1 Introduction

De nombreuses études s'intéressent à l'utilisation, l'efficacité et à la tolérance des médicaments après leur mise sur le marché afin de comprendre le contexte de prescription, vérifier le respect des recommandations et mesurer la fréquence des effets indésirables en situations réelles d'utilisation. La surveillance post-commercialisation s'intéresse à la sécurité et l'efficacité des produits pharmaceutiques en vie réelle via la collecte, l'analyse et l'interprétation de données [40]. Les études observationnelles ré-utilisent souvent des données saisies pour d'autres finalités par les professionnels de santé ou mentionnés par les patients sur les réseaux sociaux. Ces données ont l'avantage d'être produites au fil de l'eau mais présentent des difficultés de réutilisation à des fins de recherche. La détection de médicaments permet aussi d'identifier certaines maladies [41]. Identifier des médicaments est donc un objectif important pour un entrepôt de données qui est régulièrement utilisé pour mener des études de faisabilité.

Dans les données hospitalières, certaines informations sur les médicaments sont présentes sous forme structurée. Les données structurées sont issues d'un référentiel interne à l'hôpital utilisé pour la prescription et la délivrance des médicaments. Ce référentiel contient les libellés des médicaments disponibles et un code interne pour les identifier. Chaque hôpital possède son propre référentiel et le référentiel interne d'un hôpital est différent de celui utilisé dans les

pharmacies de ville. Une étude combinant données de ville et données hospitalières nécessitera de prendre en compte les différentes façons de coder l'information. La situation est encore plus complexe dans une étude internationale qui doit faire face à une grande hétérogénéité dans la codification des médicaments entre les pays. Lorsque des codes différents sont utilisés pour coder l'information, la signification d'un code peut être ambiguë et des codes différents peuvent représenter la même entité. L'interopérabilité sémantique est la capacité des systèmes informatiques à échanger des données structurées sans ambiguïté par l'utilisation d'un référentiel commun. L'utilisation d'un référentiel commun sur les médicaments faciliterait leur identification et leur recherche dans différentes sources de données [42]. Les difficultés pour intégrer les informations de différents référentiels sont les suivantes :

1. Les référentiels codent des concepts différents

   Le référentiel des médicaments de ville identifie des boîtes de médicaments commercialisées par un laboratoire pharmaceutique comme "clamoxyl 1 g, comprimé dispersible. plaquette(s) PVC PVDC aluminium de 6 comprimé(s). Laboratoire GLAXOSMITHKLINE".

   A l'hôpital, le code d'un médicament représente un produit administrable au patient comme "AMOXICILLINE - CLAMOXYL, 1 G CPR DISPERSIBLE". Ce libellé est utilisé pour la prescription et l'administration d'un médicament par les professionnels de santé qui n'ont pas besoin de préciser le laboratoire. Le référentiel de prescription de l'hôpital ne code pas l'information sur le fabricant et le référentiel des médicaments de ville n'a pas de code faisant référence au produit administrable. Aussi, certains médicaments délivrés à l'hôpital (antibiotiques, anticancéreux...) ne sont pas disponibles en pharmacie de ville.

2. Les référentiels ne structurent pas toutes les informations sur un médicament

   Souvent, un libellé concatène des informations différentes comme dans "AMOXICILLINE - CLAMOXYL, 1 G CPR DISPERSIBLE" qui contient le nom d'une molécule (amoxicilline), un nom commercial (Clamoxyl), un dosage (1 G) et une forme pharmaceutique (CPR dispersible). Un effort de normalisation est nécessaire pour séparer

la molécule, son dosage, la forme galénique et la voie d'administration d'un libellé.

3. Les référentiels ne codent pas la même information de la même façon

   En France, le référentiel des médicaments de ville attribue un code unique à chaque molécule. Le référentiel américain utilise une codification différente. Comme le nom des molécules varie en fonction de la langue, une machine est incapable d'identifier les molécules identiques dans ces deux référentiels.

4. Il peut exister une hétérogénéité des informations au sein d'un même référentiel

   En l'absence de code pour identifier une information, un référentiel peut utiliser des libellés différents pour exprimer une même information. Les unités de masse utilisées peuvent varier au sein d'un même référentiel : "amoxicilline 1000 mg" et "amoxicilline 1g". Des abréviations peuvent être utilisées pour exprimer une forme pharmaceutique, "CPR" pour comprimé par exemple.

5. Il existe des relations d'équivalence pharmacologique entre médicaments

   Certains médicaments existent sous forme de sels différents. Par exemple, la morphine est disponible sous forme de sulfate de morphine et de chlorhydrate de morphine. Le chlorhydrate de morphine n'est pas référencé dans la pharmacopée américaine tandis que c'est la forme la plus utilisée en France [43]. Par voie injectable, ces deux sels sont considérés comme équivalents [43]. Les relations d'équivalence pharmacologique entre des médicaments de référentiels différents sont importantes à prendre en compte dans certaines études, par exemple pour calculer la dose totale de morphine administrée.

En plus du problème d'interopérabilité sémantique de différents référentiels, de nombreuses informations sur les médicaments sont présentes dans les documents textuels. Les médicaments sont mentionnés à l'admission du patient, en cours d'hospitalisation et sur l'ordonnance de sortie. L'identification des médicaments dans les données textuelles nécessite d'utiliser un référentiel pour coder l'information sans ambiguïté et faciliter la recherche d'information dans l'entrepôt de données (EDS). Le référentiel nécessite d'être suffisamment exhaustif pour coder précisément une information. Dans les données en texte libre, les professionnels de santé peuvent mentionner uniquement la substance active (traité par amoxicilline), préciser le dosage

(traité par 1g d'amoxicilline) ou la voie d'administration (1g d'amoxicilline per os). Les médicaments sont aussi souvent mentionnés par leur nom commercial (CLAMOXYL®) plutôt que leur dénomination internationale (amoxicilline).

Notre objectif était de construire un référentiel du médicament qui intègre des données hétérogènes et qui permet d'identifier les médicaments dans les données textuelles. Ce chapitre présente Romedi [44], référentiel ouvert du médicament, créé pour répondre à cet objectif.

La prochaine section présente les différentes données sur les médicaments qui seront utilisées par la suite. La section "état de l'art" présente les modèles de représentation des connaissances existants sur les médicaments ainsi que les méthodes de traitement automatique de la langue (TAL) utilisées pour leur détection dans les documents. Les technologies du web sémantique utilisées pour la construction du graphe de connaissance sont aussi présentées dans cette section. La section "méthodes" présente le modèle de Romedi, son implémentation et l'étude menée pour la détection des médicaments en texte libre. La section "résultats" présente les données intégrées et ses performances pour identifier les médicaments dans les documents textuels.

3.2 Données sur les médicaments

3.2.1 Données françaises

Répertoire des médicaments et BDPM

Il existe deux sources de données open data qui décrivent les médicaments commercialisés en France : le répertoire des médicaments de l'ANSM[1] et la la base de données publique des médicaments (BDPM)[2]. Cette dernière est mise en œuvre par l'ANSM, en lien avec la Haute Autorité de Santé (HAS) et l'Union nationale des caisses d'assurance maladie (Uncam). Elle contient « des données et documents de référence sur les médicaments commercialisés ou ayant été commercialisés durant les trois dernières années en France ». La base de données publique des médicaments est un sous-ensemble du répertoire des médicaments de l'ANSM qui est plus

[1] `https://ansm.sante.fr/documents/reference/repertoire-des-medicaments`
[2] `https://base-donnees-publique.medicaments.gouv.fr/`

Figure 3.1: MOTILIUM 10 mg, comprimé pelliculé" est le nom de la spécialité pharmaceutique de ce médicament. Sa substance active est la dompéridone. Sa présentation sous forme de 40 comprimés dans une plaquette d'aluminium PVC est identifiée par le code à 13 chiffres : 3400932341122. Son titulaire est le laboratoire Janssen.

exhaustif, et contient « toutes les spécialités ayant une autorisation en cours de validité, ainsi que pour les spécialités suspendues, retirées ou abrogées depuis la mise en ligne du répertoire ».

En France, chaque présentation d'une spécialité pharmaceutique est identifiée par un code dit "code CIP"[24]. Ce code figure sur la boîte d'un médicament (figure 3.1), il identifie chaque médicament vendu en pharmacie. Ces codes CIP sont enregistrés dans la base de données nationale de l'assurance maladie à des fins de remboursement des soins. Une spécialité pharmaceutique décrit un médicament par son nom commercial, son titulaire (laboratoire pharmaceutique), son dosage et sa forme pharmaceutique (comprimé, gélule, sirop...). Chaque spécialité pharmaceutique est identifiée par un code, appelé "code CIS", et est associée à un résumé des caractéristiques du produit (RCP). Par exemple, "MOTILIUM 10 mg, comprimé pelliculé" (figure 3.1) est une spécialité pharmaceutique qui a pour code CIS 63679194, sa vente est autorisée dans une plaquette d'aluminium PVC de 40 comprimés (code CIP : 3400932341122) ou 30 comprimés (code CIP : 3400933688295). En mars 2022, le répertoire des médicaments de l'ANSM décrit la composition 140 498 présentations pharmaceutiques (codes CIP13) et 32 260 spécialités pharmaceutiques (codes CIS).

Une spécialité pharmaceutique peut contenir un ou plusieurs principes actifs. Un principe actif peut être présent sous différentes formes. Par exemple, l'ésoméprazole existe sous forme

d'ésoméprazole sodique, d'ésoméprazole magnésique et d'ésoméprazole magnésique trihydraté.

Un médicament générique peut être composé de substances différentes et avoir une forme pharmaceutique différente que le médicament princeps. Les études de bioéquivalence permettent de montrer que la pharmacocinétique du ou des principe(s) actif(s) est similaire à celle du médicament princeps. Une définition des médicaments génériques est donnée par l'article L.5121-1[3] du Code de la Santé Publique: "*une spécialité générique d'une spécialité de référence est celle qui a la même composition qualitative et quantitative en principes actifs, la même forme pharmaceutique et dont la bioéquivalence avec la spécialité de référence est démontrée par des études de biodisponibilité appropriées. Les différents sels, esters, éthers, isomères, mélanges d'isomères, complexes ou dérivés d'un principe actif sont regardés comme ayant la même composition qualitative en principe actif, sauf s'ils présentent des propriétés sensiblement différentes au regard de la sécurité ou de l'efficacité*". Toutes les formes orales à libération immédiate sont considérées comme ayant la même forme pharmaceutique [45]. Il est également possible, sous certaines conditions, que les princeps sous forme orale à libération modifiée puissent avoir un générique ayant une forme pharmaceutique différente. Selon la législation française, un comprimé pelliculé peut être équivalent à une gélule ou un comprimé effervescent [45]. L'ANSM définit la substance active (SA) comme la partie d'un médicament possédant les propriétés thérapeutiques, par opposition aux excipients, et la fraction thérapeutique (FT) comme la partie de la substance active qui porte l'activité pharmacologique.

Le répertoire du médicament de l'ANSM est librement téléchargeable en ligne et sa réutilisation est autorisée sous une licence open data. Les données de l'ANSM comprennent trois fichiers au format CSV décrivant les présentations, les spécialités pharmaceutiques et la composition des spécialités. Les données sont assez peu normalisées. Le libellé du médicament est la concaténation du nom commercial, du nom du laboratoire, du dosage et de sa forme galénique. Par exemple dans le libellé 'MOTILIUM 10 mg, comprimé pelliculé', MOTILIUM est le nom commercial, 10 mg est le dosage et comprimé pelliculé est la forme pharmaceutique. Dans le fichier de composition, la quantité de substance est exprimée sous forme de chaîne

[3] `https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000029719721/2012-12-22`

cis	SA	dosage	FT	dosage
67567004	ABACAVIR (SULFATE D')		ABACAVIR	600 mg

Table 3.1: La quantité de substance active (SA) ou de fraction thérapeutique (FT) est parfois manquante du fichier composition du répertoire des médicaments de l'ANSM. Dans cet exemple, la quantité de sulfate d'abacavir est manquante.

de caractères "10 mg" avec nombreuses variations lexicales (MG, milligramme, milligrramme etc...). La quantité de substance d'un médicament peut être indiquée par sa substance active, sa fraction thérapeutique ou les deux. Le tableau 3.1 montre un exemple où la quantité de substance active est manquante.

OpenMedic

La base de données OpenMedic est une base de données disponible en open data produite par l'assurance maladie. Elle contient le détail des ventes de médicaments en officine en France. Les médicaments vendus sont identifiés par un code CIP13.

Médicabase

Dans sa licence d'utilisation téléchargeable sur son site web, l'association Medicabase™ se décrit comme *une association regroupant des entités éditrices de base de données médicamenteuses. Ses membres se sont réunis afin de travailler à la production d'un référentiel destiné à permettre l'interopérabilité des médicaments virtuels dans le cadre de la prescription et de la dispensation en dénomination commune (DC) à travers des logiciels professionnels d'aide à la prescription ou à la dispensation.*

L'association fournit en téléchargement un référentiel sous format CSV contenant deux colonnes et 4436 lignes (mai 2022). Chaque ligne identifie un médicament virtuel par un code unique (MV + 8 chiffres) et fournit un libellé standardisé selon des règles pré-établies. Une ligne du fichier est présentée dans le tableau 3.2.

Id_MVN	Name_MVn
MV00000031	Morphine sulfate 30 mg gélule

Table 3.2: Extrait du fichier CSV de Médicabase

Le référentiel est mis à jour tous les mois. Il ne contient pas de liens vers les spécialités

pharmaceutiques commercialisées en France. Par exemple *ACTISKENAN 30 mg, gélule* est une spécialité pharmaceutique contenant 30 mg de sulfate de morphine. Il correspond donc au médicament virtuel MV00000031 de Medicabase (tableau 3.2) mais ce lien n'est pas fourni par l'association.

Réferentiel interne hospitalier

Un extrait du référentiel interne des médicaments du CHU de Bordeaux est présenté figure 3.3. Ce référentiel décrit plus de 11412 médicaments possédant un code interne (NIMED) et un code UCD13 associé dans 94% des cas.

NIMED	CONCEPT_NAME	UCD13
15233	PARACÉTAMOL, 1000 MG CPR	3400892390918
21181	CHLORURE DE SODIUM (NACL), 0,9% POCHE 100 ML	

Table 3.3: Extrait du référentiel interne de prescription du CHU de Bordeaux

Le *guide pratique pour la facturation des médicaments rétrocédés par les établissements de santé* de l'assurance maladie explique que *"Le code UCD (unité commune de dispensation) est une codification établie par l'association Club Inter-Pharmaceutique. Un code UCD correspond à la plus petite unité de dispensation (exemple : 1 comprimé, 1 flacon) contrairement au code CIP qui correspond à la présentation du médicament pour les médicaments remboursables délivrés en officine de ville."* L'association Club Inter-Pharmaceutique fournit, sur le serveur multi-terminologique de l'agence numérique en santé, un fichier CSV contenant les liens entre codes CIP13 et codes UCD13.

3.2.2 Données internationales

De nombreuses sources de données sur les molécules et les médicaments existent sur le web. Ces sources de données sont interconnectées, c'est-à-dire qu'une molécule décrite par une source possède souvent un ou plusieurs liens vers d'autres sources de données. Une brève description de chacune de ces sources est fournie ci-dessous.

PubChem PubChem [46] est une banque de données américaine gérée par le National Center for Biotechnology Information (NCBI) décrivant plusieurs millions de molécules. Chaque

molécule est identifiée par un code unique, le CID. Pour chaque molécule, PubChem fournit la structure moléculaire, la formule chimique, le poids moléculaire, les propriétés chimiques et de nombreuses autres informations. Ses données peuvent être interrogées par une API.

ChEBI ChEBI(Chemical Entities of Biological Interest)[47] est une base de données gérée par l'institut européen de bio-informatique, elle réalise un inventaire de plus de deux millions de petites molécules. ChEBI ne décrit pas les macromolécules comme les protéines ou les acides nucléiques.

DrugBank DrugBank [48] est une base de données canadienne décrivant les molécules de médicaments commercialisés et expérimentaux aux États-Unis et au Canada. Elle contient à la fois des données pharmacologiques sur les molécules et des informations sur les médicaments commercialisés. Les données de DrugBank sont téléchargeables et réutilisables à des fins non commerciales.

RxNorm RxNorm [49], vise à décrire et standardiser les informations sur les médicaments commercialisés aux États-Unis. L'utilisation de différentes terminologies sur les médicaments a conduit la National Library of Medicine à construire RxNorm pour permettre l'interopérabilité sémantique lors d'échanges d'information sur les médicaments aux États-Unis. RxNorm intègre les informations des fournisseurs de bases de données sur les médicaments aux États-Unis et les codes nationaux NDC (national drug code) de la FDA utilisés pour le remboursement [50]. Les données de RxNorm sont revues et normalisées par un groupe d'experts. Les informations relatives au médicament comme les indications ou les interactions ne sont pas présents dans le modèle [50]. Contrairement à PubChem, ChEBI ou DrugBank, RxNorm ne contient pas d'information sur les propriétés pharmacologiques des substances. Les données de RxNorm sont téléchargeables sous forme de fichiers texte et des scripts permettent de les charger dans une base de données relationnelle. Les données sont aussi visualisables dans l'interface RxNav (figure 3.2) et interrogeables via des API. RxNorm a un modèle de représentation du médicament qui est décrit dans la suite de ce chapitre.

Figure 3.2: Visualisation dans l'interface RxNav des différents concepts liés au sulfate de morphine dans RxNorm

OMOP Le modèle commun de données OMOP-CDM (Observational medical outcomes partnership - Common Data Model) est un modèle relationnel de bases de données. Le schéma OMOP possède des tables dédiées au stockage de terminologies standards fournies par le consortium OHDSI (Observational Health Data Sciences and Informatics) en charge du développement et de la maintenance d'OMOP. Ces terminologies standards sont téléchargeables en ligne via le logiciel Athena[4]. L'intégration de données au format OMOP nécessite d'aligner les terminologies locales vers les terminologies standards d'OMOP. Ces alignements garantissent l'interopérabilité sémantique permettant le partage de requêtes SQL entre entrepôts de données OMOP et la réalisation d'études internationales. Pour standardiser les informations sur les médicaments, OMOP utile le modèle de RxNorm [50]. Lorsqu'un médicament n'est pas décrit par RxNorm, le plus souvent car il n'est pas commercialisé aux États-Unis, OHDSI crée et ajoute de nouveaux concepts. Ces nouveaux concepts sont appelés RxNormExtension.

Classification ATC La classification ATC (Anatomique, Thérapeutique et Chimique) est une terminologie devéloppée et maintenue par l'organisation mondiale de la santé. Elle est recommandée et utilisée internationalement dans les études pharmacologiques. C'est une classification hiérarchique mono-axiale composée de 14 axes identifiés par une lettre majuscule correspondant à des groupes anatomiques. La hiérarchie comprend cinq niveaux:

[4] `https://athena.ohdsi.org/`

- 1er niveau : groupes anatomiques

  A : Alimentary tract and metabolism

- 2ème niveau : sous-groupe thérapeutique

  A10 : Antidiabétiques

- 3ème niveau : sous-groupe pharmacologique

  A10B : Antidiabétiques oraux

- 4ème niveau : groupes chimiques

  A10BA : Biguanides

- 5ème niveau : nom de la molécule en DCI

  A10BA0 : Metformine

La classification est utilisée pour classer des médicaments. Certaines molécules, comme l'acide clavulanique, sont présentes seulement en combinaison avec d'autres et n'ont pas d'identifiant individuel. Comme une molécule peut être utilisée dans plusieurs indications, elle peut avoir plusieurs codes ATC. Par exemple, la bromocriptine est utilisée dans le traitement de la maladie de Parkinson (N04BC01) et des hyperprolactinémies physiologiques ou pathologiques (G02CB01). Bien qu'il s'agisse de la même molécule, il n'existe pas de relation entre ces deux codes dans la hiérarchie. Cette classification ne permet pas d'identifier toutes les molécules et regroupe des médicaments qui ont des formes galéniques et des voies d'administration différentes.

Wikidata Wikidata [51] est une base de connaissances généraliste éditée de façon collaborative. Chaque entité de Wikidata (personne, ville, molécule...) est identifiée par un code unique. Chaque entité possède un libellé principal dans chaque langue et éventuellement plusieurs libellés alternatifs. Chaque entité est décrite par un ensemble de triplets : entité, propriété, valeur. La liste des propriétés est fixée par certains utilisateurs nommés administrateurs de Wikidata. Certaines propriétés ont été créées pour décrire une substance : sa masse moléculaire en dalton, sa formule chimique, son(ses) code(s) ATC et son identifiant dans les autres sources de données (RxNorm, PubChem, ChEBI, DrugBank...). Les valeurs autorisées pour chacune des propriétés

sont vérifiées par des expressions régulières. Un code DrugBank doit par exemple commencer par "DB" suivi d'une série de chiffres. Wikidata propose une interface d'édition qui permet à toute personne d'ajouter de l'information et de modifier son contenu. Wikidata propose aussi un SPARQL endpoint pour interroger ses données de façon programmatique.

3.3 Etat de l'art

3.3.1 Modélisation du médicament

Standards internationaux

En 2012, l'organisation internationale de normalisation (ISO) a publié l'ISO 11615 *Informatique de santé, Identification des médicaments, Éléments de données et structures pour l'identification unique des médicaments* [52]. Cette norme constitue le noyau d'une série de standards connus sous le nom d'IDMP (Identification of Medicinal Products)[5] qui décrivent les concepts tels que les substances, les formes pharmaceutiques, les voies d'administration, les unités de présentation et les unités de mesure [52]. L'objectif de l'ISO IDMP est de fournir un système harmonisé pouvant être utilisé dans le monde entier pour identifier les médicaments. L'utilisation de l'ISO IDMP est une exigence réglementaire selon le règlement d'exécution (UE) n°520/2012 de la Commission (articles 25 et 26) qui rend obligatoire l'utilisation de l'ISO IDMP pour l'échange d'informations sur les médicaments dans l'Union européenne. Cette norme a été développée pour faciliter les échanges d'information sur les médicaments dans le cadre d'activités de pharmacovigilance, d'essais cliniques, de prescription et de dispensation médicamenteuse. L'ISO IDMP est composé de cinq standards ISO :

- ISO 11238 décrit l'identification des substances
- ISO 11239 aborde l'identification des formes pharmaceutiques, des unités de présentation, des voies d'administration et des emballages.
- ISO 11240 concerne les unités de mesures

[5] `https://www.iso.org/standard/70150.html`

- ISO 11616 traite des produits pharmaceutiques (pharmaceutical product). Un produit pharmaceutique décrit le médicament : substances, dosages, formes pharmaceutiques, voie d'administration etc...
- ISO 11615 aborde les produits médicinaux (medicinal product). Un produit médicinal décrit les informations annexes à un produit pharmaceutique : nom du produit, caractéristiques du produit (indications ...), laboratoire, numéro d'autorisation etc...

Ces normes ISO IDMP définissent les informations requises pour identifier un médicament mais ne fournissent pas d'implémentation.

Pour l'identification des substances, l'Agence européenne des médicaments (EMA) utilise un vocabulaire contrôlé des substances nommé SMS (Substances Management Services). Le travail de normalisation des substances par l'EMA est encore en cours [53].

L'EMA travaille avec la direction européenne de la qualité du médicament et des soins de santé (EDQM), une direction du Conseil de l'Europe. L'EDQM propose des normes de qualité pour les médicaments et leur utilisation. L'EDQM a développé une implémentation de l'ISO 11239 pour les formes pharmaceutiques, les voies et/ou méthodes d'administration, les unités de présentation, les conteneurs, les fermetures et les dispositifs d'administration pour les médicaments à usage humain et vétérinaire. Cette implémentation porte le nom de base de données de termes standards (standard terms datababase). Cette base de données contient plus de 900 termes en anglais et des traductions dans 35 langues. Elle est accessible en ligne[6] et requêtable par une API.

La forme pharmaceutique est utilisée pour décrire le produit fabriqué, comme il est présenté dans l'emballage (poudre pour injection) et le produit pharmaceutique qui est transformé pour obtenir la forme qui sera administrée (solution pour injection). La forme pharmaceutique est une notion complexe car elle est composée de plusieurs concepts et peut décrire des combinaisons de formes ainsi que le contenant (seringue, ampoule ...).
Une forme possède six caractéristiques : l'état de la matière, sa forme basique, son mode d'administration, son site d'administration, ses caractéristiques de libération et sa transformation. Par exemple, un collyre en solution (Eye drops, solution) possède les caractéristiques

[6] `https://standardterms.edqm.eu/`

suivantes : son état est liquide, sa forme basique est une solution, son mode d'administration est une instillation, le site d'administration est l'œil, sa libération est conventionnelle et aucune transformation n'est nécessaire avant de l'administrer. La définition de chacune de ces caractéristiques est traduite du document de l'EDQM et reproduite ci-dessous :

Etat de la matière Condition physique décrivant la forme moléculaire d'un produit (solide, semi-solide, liquide et gazeux).

Forme basique Version généralisée de la forme pharmaceutique utilisée pour regrouper des formes apparentées (comprimé, gélule . . .)

Mode d'administration Le mode d'administration est utilisé pour indiquer la manière dont le médicament doit être administré.

Site prévu d'administration (intended site) Site anatomique où un produit pharmaceutique est destiné à être administré.

Caractéristiques de libération Description du moment où un ingrédient actif est rendu disponible dans le corps après l'administration du produit pharmaceutique, par rapport à une libération directe conventionnelle.

Transformation Procédure qui est effectuée afin de convertir un article manufacturé (qui nécessite une telle procédure) en un produit pharmaceutique, c'est-à-dire de sa forme pharmaceutique fabriquée à sa forme pharmaceutique administrable.

Une forme pharmaceutique peut être simple (poudre), une "forme combinée" (poudre et solvant pour solution injectable) ou un "terme combiné" (solution pour injection dans une seringue pré-remplie). L'EDQM précise que les formes pharmaceutiques qui ne diffèrent qu'au niveau du contenant/dispositif d'administration ne sont pas toujours considérées comme des formes pharmaceutiques différentes.

L'EDQM définit le concept d'**unité de présentation** pour décrire un dosage ou une quantité en termes d'une entité dénombrable, plutôt qu'une unité de mesure. Par exemple, *30 mg de*

sulfate de morphine par comprimé a une unité de présentation qui est le comprimé, *30 mg de sulfate de morphine par ampoule* a une unité de présentation qui est l'ampoule. L'EDQM fournit la définition suivante de l'unité de présentation : "*terme qualitatif décrivant l'entité discrète dénombrable dans laquelle se présente un produit pharmaceutique ou dans lequel un produit pharmaceutique ou un article manufacturé est présenté, dans les cas où la force ou la quantité est exprimée par rapport à une instance de cette entité dénombrable".*

L'EDQM fait la différence entre la voie d'administration d'un médicament et le site d'administration d'un médicament, caractéristique d'une forme pharmaceutique. La définition de la voie d'administration est la suivante : *voie par laquelle le produit pharmaceutique est introduit dans le corps ou entre en contact avec celui-ci. La voie d'administration indique la partie du corps sur laquelle, à travers laquelle ou dans laquelle le médicament doit être introduit.*

RxNorm

La figure 3.3 présente la structure de RxNorm. RxNorm utilise les classes suivantes pour décrire les composants d'un médicament[7]:

- Precise ingredient : il correspond à la substance active (ex: azithromycine dihydratée)
- Ingredient : il correspond à la fraction thérapeutique (ex: azithromycine)
- Dose Form : la forme pharmaceutique (comprimé oral, solution injectable)
- Clinical Drug Component : il combine une fraction thérapeutique et son dosage (ex: azithromycine 250 mg).
- Clinical Drug : il combine une fraction thérapeutique, sa quantité et sa forme pharmaceutique (ex: abacavir 250 mg comprimé oral).
- Generic Pack : il décrit la combinaison d'un ou plusieurs clinical drug (ex: 6(abacavir 250 mg comprimé oral))

[7] `https://www.nlm.nih.gov/research/umls/rxnorm/overview.html`

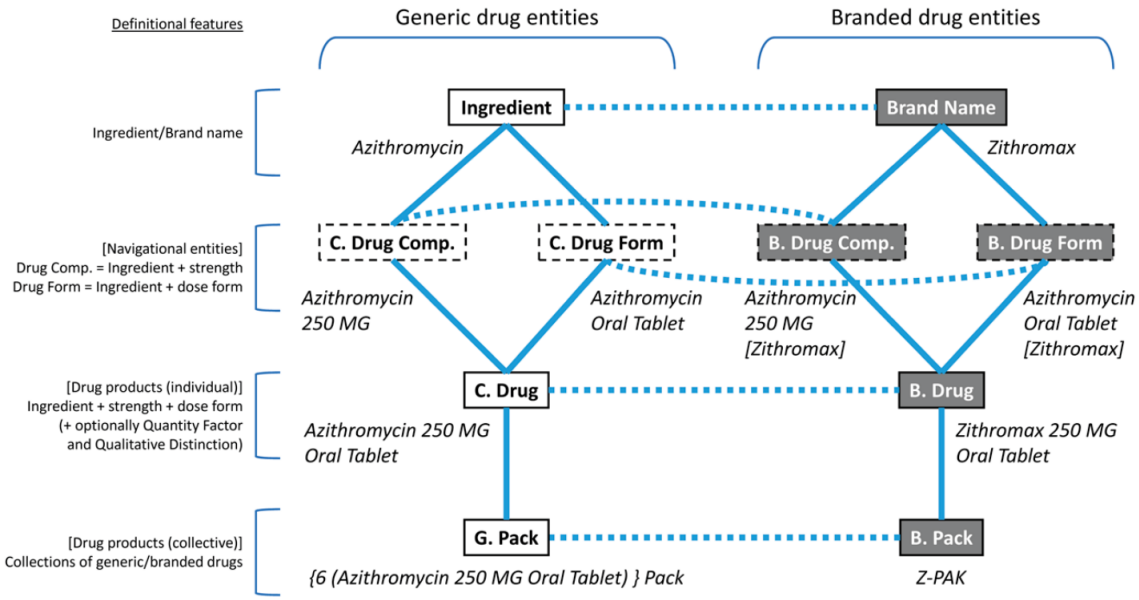


Figure 3.3: Structure du modèle de RxNorm d'après Bodenreider et al.[50]

- Nom commercial : nom donné par le laboratoire pharmaceutique figurant sur l'emballage (ex: Zithromax)

RxNorm distingue les entités génériques, non associées à un produit commercial, et les entités liées à un produit commercial (figure 3.3).

Un produit pharmaceutique est entièrement défini par l'ensemble de ses ingrédients, leur dosage, la forme galénique et, optionnellement, un facteur de quantité et des distinctions de formes (sans sucre, libération prolongée ...) [50]. Dans les cas où certains produits ont une même concentration, le facteur de quantité (quantity factor) permet de préciser la quantité de produit. Par exemple, pour une concentration à 1mg/ml, un facteur de quantité 2ml signifie que le médicament contient 2mg de substance dans 2ml de solution.

La classe Ingrédient du modèle (figure 3.3) regroupe les ingrédients à proprement parler, notion équivalente à celle de fraction thérapeutique définie par l'ANSM, les ingrédients précis, notion équivalente à la substance active de l'ANSM et les ingrédients multiples. Lorsqu'un médicament contient plusieurs ingrédients, un concept d'ingrédient est créé pour représenter cette combinaison.

RxNorm possède de nombreuses relations entre les concepts pour faciliter la navigation dans le graphe d'entités. Certains concepts sont nommés "concepts navigationnels" (figure 3.3) car ils sont utilisés pour naviguer dans la terminologie. La figure 3.2 présente l'interface RxNav

permettant de visualiser et naviguer dans le contenu de RxNorm.

SNOMED CT

La SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) est la plus grande et exhaustive terminologie médicale. Elle est maintenue et distribuée par SNOMED International, un organisme international à but non lucratif localisé en Angleterre. Elle vise à assurer l'interopérabilité sémantique des informations échangées dans le domaine médical. La SNOMED CT est utilisée dans plus de 40 pays [54]. L'utilisation de la SNOMED CT dans les logiciels commercialisés nécessite une licence, son utilisation est gratuite à des fins de recherche scientifique en informatique médicale. La SNOMED CT est une terminologie multiaxiale contenant douze axes (anatomie, morphologie, médicaments ...). Chaque concept est lié à un ou plusieurs concepts parents par une relation "is-a". La terminologie contient de nombreuses autres relations binaires sur l'ensemble des concepts. La SNOMED CT utilise une logique de description, le langage $\mathcal{EL}$++ [55], adaptée aux ontologies possédant un très grand nombre de classes et/ou de propriétés [8]. Ce langage permet de vérifier la cohérence (absence de contradiction), la subsomption des classes et la vérification des instances en temps polynomial.

La SNOMED CT s'intéresse à la standardisation des informations sur les médicaments pour permettre l'interopérabilité sémantique des informations au niveau international. Le concept SNOMED CT de *30 mg de sulfate de morphine dans un comprimé* est représenté figure 3.4.

Le type sémantique de ce concept est 'Clinical Drug'. Il s'agit d'un concept défini par une classe d'équivalence en logique de description. L'utilisation de classes définies par des conditions nécessaires et suffisantes permet de grouper par raisonnement les médicaments dans des classes d'équivalence. Les instances de médicaments contenant uniquement de la morphine par voie orale, sous forme de comprimé à libération conventionnelle, contenant 30 mg de sulfate de morphine dans un comprimé seront groupés dans cette classe. L'utilisation de la SNOMED CT devrait permettre d'identifier des médicaments similaires commercialisés dans des pays différents.

Dans la SNOMED CT, un dosage est exprimé en tant que concentration (30mg/ml) ou en unité de présentation (30mg par comprimé). La SNOMED CT normalise les unités de présenta-

[8] `https://www.w3.org/TR/owl2-profiles/`

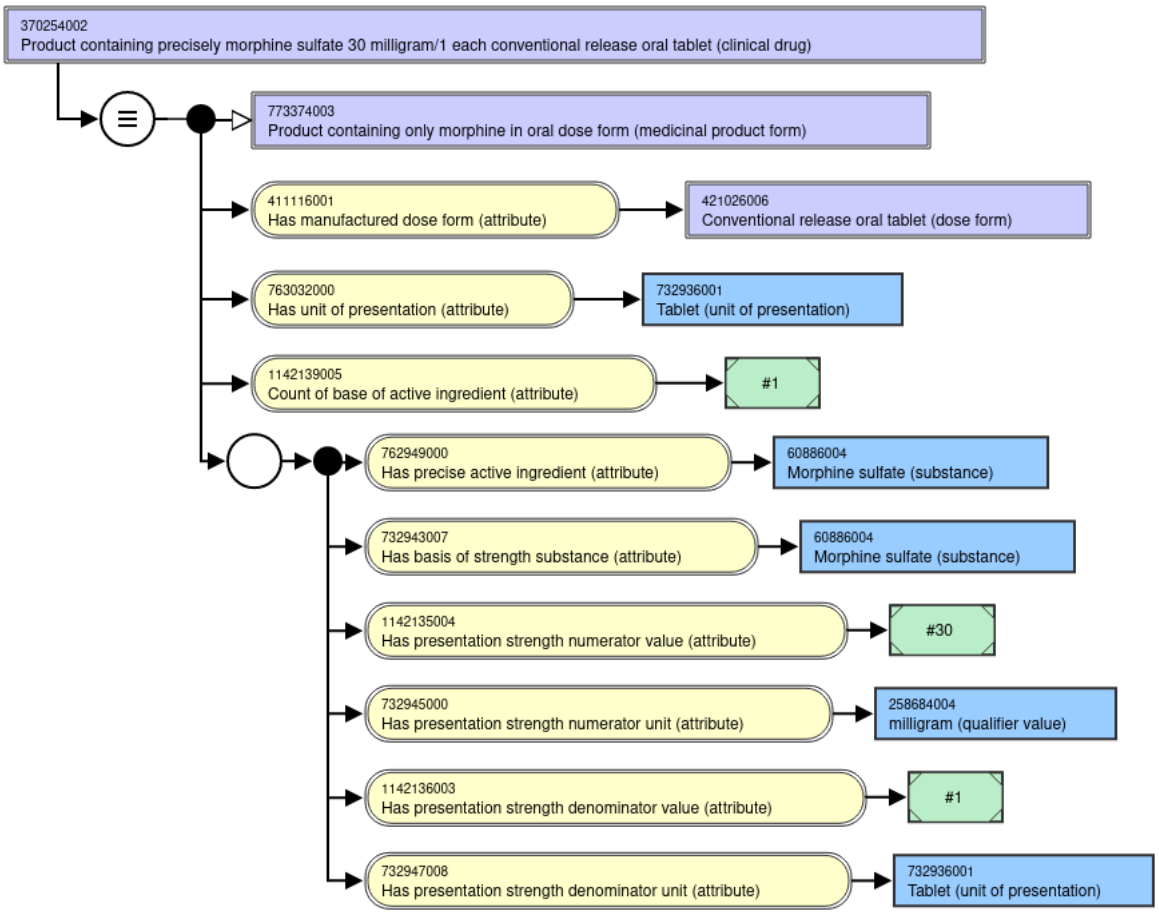


Figure 3.4: Représentation du concept de *30 mg de sulfate de morphine dans un comprimé* par la SNOMED CT

tion selon le standard ISO IDMP. Dans l'exemple de la figure 3.4, le dosage est exprimé en unité de présentation. La substance de base du dosage (Basis of strength substance, BoSS) est la substance à laquelle le dosage du médicament fait référence. Il peut s'agir de la fraction thérapeutique ou de la substance active comme dans l'exemple de la figure 3.4. L'expressivité de $\mathcal{EL}$++ ne permet pas l'utilisation d'une restriction universelle pour restreindre un médicament aux ingrédients décrits par l'ontologie. Puisque la sémantique du langage utilise l'hypothèse d'un monde ouvert, un médicament pourrait contenir d'autres ingrédients non décrits par l'ontologie. La solution choisie par la SNOMED CT pour résoudre ce problème a été de créer un attribut *(count of base of active ingredient)* décrivant le nombre d'ingrédients. La SNOMED CT lie une fraction thérapeutique à une substance active par la relation "is a modification of". Le sulfate de morphine est par exemple une modification de la morphine.

Le type sémantique de "Clinical Drug" dans la SNOMED CT est similaire au concept de RxNorm : il décrit un médicament par ses ingrédients, leur dosage et leur forme pharmaceu-

tique. Les autres types sémantiques sont "medicinal product" décrivant uniquement les ingrédients et "medicinal product form" décrivant les ingrédients et la forme pharmaceutique.

La SNOMED CT a réalisé une révision majeure de sa modélisation du médicament en 2018 [52]. Dans la précédente version de la SNOMED CT, les ingrédients étaient des sous-classes (is-a) des ingrédients précis ce qui était incorrect et conduisait à des raisonnements faux [52]. Les classes représentant les médicaments étaient des classes primitives et non des classes définies. Le dosage des médicaments n'était pas représenté par une classe mais par une chaîne de caractères. D'après Bodenreider et al.[52], ce nouveau modèle de la SNOMED CT étend et complète le modèle IDMP de manière compatible et harmonieuse.

Différences entre RxNorm et SNOMED CT

RxNorm et la SNOMED CT sont deux terminologies utilisées pour modéliser les informations sur les médicaments. RxNorm ne décrit que les médicaments tandis que la SNOMED CT est une terminologie médicale généraliste. RxNorm a été conçue dans les années 90 pour permettre l'interopérabilité des médicaments aux États-Unis. La SNOMED CT vise l'interopérabilité sémantique des informations sur les médicaments à l'échelle internationale. SNOMED CT est plus conforme avec les standards internationaux IDMP mais RxNorm est en cours d'évolution pour améliorer la conformité avec les normes internationales [50]. Le modèle des médicaments dans la SNOMED CT est plus récent avec une révision majeure en 2018. Ces différences historiques peuvent expliquer le choix du consortium international OHDSI qui a développé OMOP d'avoir utilisé et étendu le modèle de RxNorm plutôt que d'utiliser la SNOMED CT, par ailleurs utilisée pour standardiser d'autres informations comme les actes médicaux. Le modèle de RxNorm a été utilisé pour intégrer les référentiels de médicaments de plusieurs pays [56].

Nikiema et Bodenreider ont analysé les différences des modèles RxNorm et SNOMED CT. Une différence majeure en termes de formalisme est que RxNorm n'utilise pas un langage de logique de description. Les modèles RxNorm et SNOMED CT sont proches même si des différences de représentation existent [54]. RxNorm ne précise pas explicitement si le dosage est exprimé en unité de présentation ou en concentration et RxNorm n'a pas de notion d'unité de présentation. La principale forme pharmaceutique dans RxNorm est "Oral Tablet" (comprimé oral) mais l'unité de présentation "Tablet" (comprimé) n'est pas un concept. La notion

d'ingrédient dans RxNorm est ambiguë car elle peut signifier soit une substance unique soit la combinaison de plusieurs substances [54]. Le concept d'ingrédient dans RxNorm peut correspondre à celui de "medicinal product" (plusieurs ingrédients) ou à celui de "substance" (un seul ingrédient) dans la SNOMED CT.

Le concept "clinical drug" de RxNorm est similaire à celui de SNOMED CT. Le facteur de quantité (quantity factor) de RxNorm n'a pas d'équivalence dans la SNOMED CT mais cette information est présente implicitement. Les caractéristiques de libération d'une forme pharmaceutique sont absentes dans le modèle de la SNOMED CT.

Modèle sémantique d'HeTOP

Un modèle sémantique d'identification du médicament en France a été développé par le Département d'Information et d'Informatique Médicale (D2IM) de Rouen et l'équipe de recherche LIMICS [57]. Ce modèle permet d'intégrer les informations relatives aux informations des médicaments français : base de données BDPM, Medicabase, classification ATC, codes CIP et UCD notamment. Le modèle est composé d'un ensemble de classes reliées par des liens sémantiques permettant de lier les informations des spécialités pharmaceutiques françaises à leur forme, voie d'administration, médicament virtuel, codes ATC, CIP et UCD. Les auteurs soulignent les nombreuses difficultés de normalisation de la BPDM pour instancier le modèle. Les instances de ce modèle ont été intégrées au serveur multiterminologique HeTOP et son contenu est accessible en ligne[9]. Le modèle est très différent de celui de RxNorm et de la SNOMED CT. Par exemple, il ne normalise pas les informations sur le dosage des substances actives.

3.3.2 Identification des médicaments dans le texte libre

L'identification des médicaments dans les données textuelles a fait l'objet de nombreux travaux de recherche en TAL [58, 59]. Cette identification est réalisée en deux étapes [60]. La première étape, la reconnaissance d'entités nommées, vise à identifier les mots d'un texte correspondant à des entités en lien avec le médicament (ingrédient, dosage, forme). Plusieurs méthodes de TAL peuvent être utilisées : méthodes à base de dictionnaires, de règles ou par apprentissage

[9] `https://www.hetop.eu/hetop/drugs/`

automatique. L'apprentissage automatique est nécessaire lorsque la base de connaissance n'est pas exhaustive. Chaque année, des nouvelles molécules sont créées en laboratoire et mentionnées dans des articles de recherche. Il est nécessaire de les répertorier et de les indexer dans une base de connaissance pour être identifiables. L'apprentissage automatique est utile dans cette situation car elle détecte une substance en utilisant son contexte. Par exemple, dans la phrase "le patient prend de l'abacavir 300 mg", le verbe prendre, le nombre 300 et le terme "mg" permettent de prédire que le terme "abacavir" est un ingrédient. Ces algorithmes d'apprentissage supervisé nécessitent des annotations manuelles par un expert du domaine. Ils ont obtenu de bonnes performances pour détecter les entités liées à un médicament dans des documents médicaux d'un EDS [61].

La deuxième étape consiste à relier le mot détecté d'un document, étiqueté par une entité, à une base de connaissance. Elle peut être réalisée automatiquement par comparaison de chaînes de caractères ou manuellement par un expert du domaine. La base de connaissance apporte de l'information sémantique, le sens au mot détecté. Par exemple, lorsque le mot "abacavir" lié au code DB01048 de DrugBank, la base de connaissance permet de connaître son type (petite molécule), sa formule chimique (C14H18N6O), sa biodisponibilité (83% en administration orale), sa demi-vie (1,54 heures) et beaucoup d'autres informations. L'annotation sémantique permet d'enrichir les entités détectées dans un texte par des informations contenues dans une base de connaissance. L'annotation sémantique est particulièrement utile pour la recherche d'information car elle permet de regrouper et de rechercher les médicaments détectés par leurs propriétés pharmacologiques ou par leurs classes thérapeutiques (anti-hypertenseur, inhibiteur de l'enzyme à proton...).

3.3.3 Web sémantique

Les traitements réalisables par une machine sont limités par la façon dont les données sont stockées et représentées. Les technologies du web sémantique facilitent la représentation, la publication, les liens entre les données et la recherche d'information [62]. Au lieu de naviguer à travers des pages webs, le web sémantique propose de naviguer à travers les données [63]. D'un web de données en silo où l'information est inaccessible aux machines, le web sémantique,

appelé parfois web 3.0, est constitué de liens sémantiques entre données de plusieurs sources de données distribuées sur le web.

L'intégration des données du web est confrontée aux problèmes d'interopérabilité technique et sémantique. Le problème d'interopérabilité technique résulte de la présence de données dans divers formats (PDF, Texte...) et encodées de différentes façons (ISO8559-1,UTF8-UTF16...). L'expression de différentes manières d'un même concept pose le problème d'interopérabilité sémantique : la signification des informations contenues dans les données ne doit pas être ambiguë. Les technologies du web sémantique offrent des solutions pour résoudre ces problèmes d'interopérabilité au travers de standards et d'outils. Elles incluent un modèle de données (RDF), des syntaxes pour sérialiser les données (XML, Turtle), un langage de requête (SPARQL) et des schémas (RDFS, OWL) pour décrire les métadonnées et raisonner sur les données [64].

Le RDF (Resource Description Framework) est un modèle graphe de données. Son unité de base est un triplet constitué de :

- un sujet, il représente la ressource à décrire
- un prédicat (ou relation ou propriété), il décrit une propriété du sujet
- un objet, il représente une valeur (texte, numérique, date...) ou une autre ressource

Par exemple, « le paracétamol est une molécule » est une affirmation en langage naturel qui peut être décrite en RDF : le paracétamol est le sujet, le prédicat est la relation "est un" et l'objet est le concept de molécule. Les ressources et les propriétés sont identifiées de façon unique par un identifiant uniforme de ressource ou URI (Unique Ressource Identifier). Les URL (Unique Ressource Locator) identifient un document sur le web (page web, fichier audio ...) et constituent un sous-ensemble des URI qui identifient aussi des objets du monde réel (une personne, une ville...) ou un concept abstrait comme une molécule. Les URI servent à identifier ou référencer toute chose réelle ou non. Bien que les URI contiennent le préfixe http:// comme les URL, il n'est pas garanti qu'une page web soit retournée si l'URI est entré dans la barre d'adresse d'un navigateur. Il est cependant considéré comme une bonne pratique de retourner une description d'un URI quand un navigateur cherche à y accéder. Ce procédé s'appelle le

déréférencement de l'URI. Des exemples de déréférencement peuvent être observés en entrant les URI précédents dans un navigateur.

Les propriétés (ou relations) sont aussi identifiées par des URI. Contrairement aux liens hypertextes du web classique, les liens du web de données ont un sens. Les propriétés peuvent être des liens internes pour décrire les données ou des liens externes pour se lier à d'autres sources. Une source de données sur le web sera d'autant plus utile et utilisable qu'elle sera connectée à d'autres sources.

Les triplets RDF forment un graphe qu'il est possible de stocker et d'interroger. Un triplestore est un système de gestion de bases de données (SGBD) spécialement conçu pour les données RDF. Il permet de stocker de façon efficiente les données RDF pour les interroger et les récupérer avec le langage de requête SPARQL. Il existe de nombreux triplestores dont certains sous licence libre. L'interface d'un triplestore qui permet d'envoyer des requêtes SPARQL est appelé un SPARQL endpoint. De nombreux sites Open Data sur le web fournisse un SPARQL endpoint pour interroger leur base de données. L'une des forces du langage SPARQL est d'offrir la possibilité d'interroger plusieurs SPARQL endpoint simultanément lors d'une même requête. Le web sémantique fournit une solution d'intégration distribuée [62]. Elle permet de récupérer des informations distribuées sur le web sans avoir besoin de télécharger l'intégralité des bases de données.

3.4 Méthodes

La première version du modèle de Romedi visait à normaliser les données de la BDPM afin de détecter et d'identifier les médicaments dans les documents en texte libre [44]. La deuxième version, présentée ci-dessous, visait à représenter les informations d'un médicament, à intégrer des bases de données hétérogènes (figure 3.5) dans un modèle commun et à identifier des relations pharmacologiques entre des médicaments différents. Le modèle de Romedi est proche de celui de RxNorm et de la SNOMED CT bien qu'il utilise un formalisme très différent. Ces modèles ne permettent pas d'intégrer simultanément les informations de dosage de la substance active et de la fraction thérapeutique ou de calculer une quantité de fraction thérapeutique à partir d'une quantité de substance active (tableau 3.1). Ces modèles permettent de standardiser les

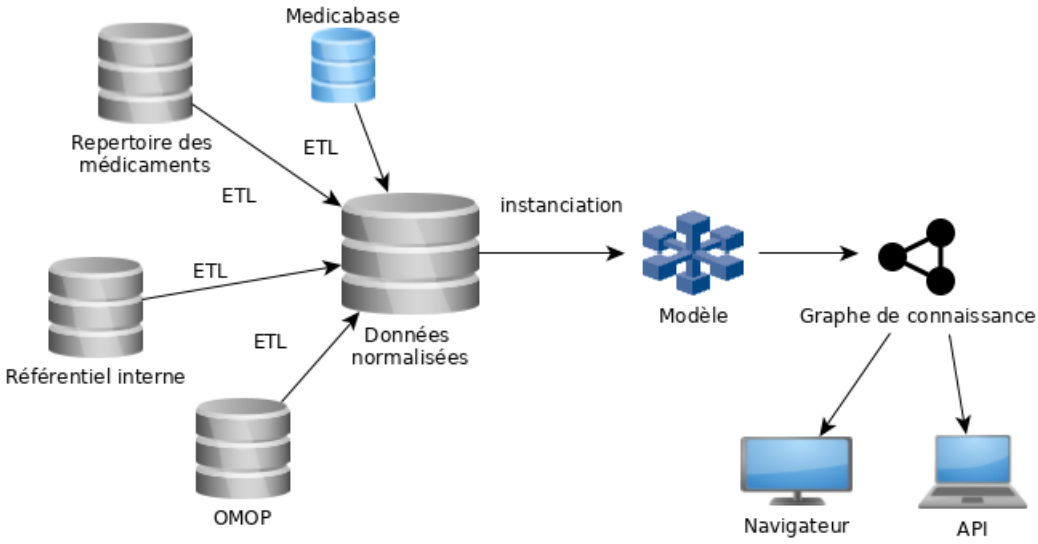


Figure 3.5: Architecture de Romedi. Les données sur les médicaments sont d'abord normalisées puis chargées et instanciées dans le modèle de Romedi. Les instances et les relations d'équivalence sont stockées dans un triple store. Les résultats sont visualisables dans une interface web et requêtables via un SPARQL endpoint

informations d'un médicament mais ne permettent pas de découvrir des relations d'équivalence pharmacologique comme celle qui existe entre 1 mg de sulfate de morphine et 1 mg de chlorhydrate de morphine. Notre motivation initiale était d'intégrer des référentiels hétérogènes et de trouver des alignements terminologiques dans le cadre du projet européen EHDEN[10].

Dans le modèle de Romedi, une base de données doit d'abord être normalisée (figure 3.5). La base de données est ensuite instanciée dans le modèle puis transformée en triplets RDF. L'intégration de différentes bases de données conduit à générer un graphe de connaissance unifié permettant de trouver des alignements terminologiques.

Cette section est organisée de la façon suivante :

1) le modèle de représentation est d'abord exposé ci-dessous.

2) Le choix des terminologies retenues pour identifier les éléments primitifs du modèle (substances, formes pharmaceutiques...) est ensuite expliqué

3) La normalisation des bases de données, l'instanciation du modèle et la création du graphe de connaissance sont présentées.

Le mot "médicament" peut avoir plusieurs significations en fonction de son contexte. Dans cette section, le mot "médicament" désigne une ou plusieurs substances dans une unité de dis-

[10] `https://www.ehden.eu/`

pensation (comprimé, flacon ...). Le terme "médicament commercialisé" désigne une boîte de médicaments fabriquée par un laboratoire pharmaceutique.

3.4.1 Modèle de Romedi

Le modèle de Romedi s'appuie sur la théorie des ensembles pour décrire un médicament et n'utilise pas de logique de description pour la représentation des connaissances. Le modèle a recours aux notions d'ensemble (appelée classe ou concept en logique de description), d'élément (aussi appelée instance ou individu), de relation (aussi appelée propriété ou rôle) et de fonction (aussi appelée application). Le modèle de Romedi comprend des ensembles primitifs (tableau 3.4), des ensembles construits (tableau 3.5), des fonctions et des relations d'équivalence.

Ensemble	Notation	Exemple
Substance	S	abacavir
Quantité de matière	AMOUNT	600
Unité de masse	UNIT	mg
Forme galénique	FORM	comprimé
Voie d'administration	ROUTE	orale

Table 3.4: Ensembles primitifs du modèle de Romedi

Pour faciliter la comparaison future avec RxNorm et la SNOMED CT, Romedi réutilise les libellés de certaines classes de ces modèles : ingredient (IN), precise ingredient (PIN), clinial drug component (CDC), clinical drug (CD) et pack. Pour faciliter la lecture, ces libellés en anglais ne sont pas traduits.

Substances et relation isPartOf

L'ensemble *S* des substances est un ensemble fini de molécules possédant une activité pharmacologique. La relation *isPartOf* est une relation binaire entre deux substances: 's_1 isPartOf s_2' signifie que la substance s_1 est une partie de la substance s_2. Par exemple la paire (abacavir, sulfate d'abacavir) appartient à la relation *isPartOf* d'après PubChem (figure 3.6).

Suivant la recommandation du W3C[11], le modèle de Romedi décrit la relation *isPartOf* comme réflexive, transitive et antisymétrique. L'ensemble (S,≤) est donc un ensemble partiellement ordonné par la relation *isPartOf* [65]. Une substance $s_1 \in$ (S,≤) est un élément minimal

[11] https://www.w3.org/wiki/PartWhole

Ensemble	Définition	Exemple
Dosage (STRENGTH)	AMOUNT × UNIT	(600,mg)
Dosage de la référence (STRENGTH_REF)	AMOUNT × (UNIT ∪ {form}*)	(1,ml)
Concentration (C)	STRENGTH × STRENGTH_REF	((600,mg),(1,ml))
Clinical Drug Component (CDC)	S × C	(abacavir,((600,mg),(1,ml)))
Multiple CDC (CDCS)	$\mathcal{P}(CDC)$	{(abacavir,((600,mg),(1,ml))), (lamivudine,((300,mg),(1,ml)))}
Multiple ingredients (MIN)	$\mathcal{P}(S)$	{abacavir,lamivudine}
Clinical Drug (CD)	CDCS × FORM	({(abacavir,((600,mg),(1,ml))), (lamivudine,((300,mg),(1,ml)))}, solution)
Quantified Clinical Drug (QuantCD)	$\mathcal{Q}$ × CD	(2.5,({(abacavir,((600,mg),(1,ml))), (lamivudine,((300,mg),(1,ml)))}, solution))

Table 3.5: Ensembles construits du modèle de Romedi. Les éléments des ensembles construits sont créés par différentes combinaisons d'éléments primitifs.
×: produit cartésien.
$\mathcal{P}(X)$: ensemble des parties de l'ensemble X.
*: l'élément 'form' est une unité spéciale qui désigne une unité de dispensation.

s'il n'existe aucune substance s_2 telle que $s_2 < s_1$, c'est-à-dire s'il n'existe aucune substance s_2 qui soit une partie de s_1. Les éléments minimaux sont appelés "fractions thérapeutiques" (IN) et les autres éléments "substances actives"(PIN). Cette relation *isPartOf* permet de définir les sous-ensembles *IN* et *PIN*. En notant R la relation *isPartOf* :

$PIN = \{s_2 \in S \mid \exists s_1 \in S$ tel que $s_1 R s_2$ et $s_1 \neq s_2\}$,

$IN = \{s_2 \in S \mid \forall s_1 \in S\ (s_1, s_2) \notin R$ et $s_1 \neq s_2\}$.

Pour démontrer qu'une substance s_1 n'est pas une substance active, c'est-à-dire qu'elle n'appartient pas à l'ensemble *PIN*, il faut vérifier qu'il n'existe aucune autre substance $s_2 (s_1 \neq s_2)$ qui ne soit pas une partie de s_1. Démontrer qu'une substance s_1 n'appartient pas à l'ensemble *PIN* est donc équivalent à démontrer que la substance s_1 appartient à l'ensemble *IN*, c'est-à-dire que s_1 est une fraction thérapeutique. Les ensembles *IN* et *PIN* sont mutuellement disjoints et forment deux partitions de l'ensemble des substances *S* :

$S = PIN \cup IN$ et $PIN \cap IN = \varnothing$

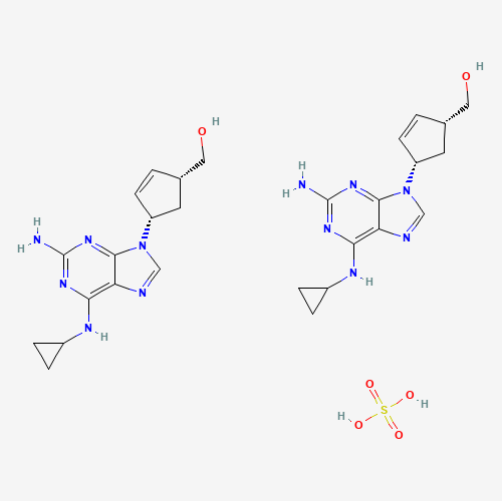


Figure 3.6: Sulfate d'abacavir composé de deux molécules d'abacavir et d'un acide sulfurique d'après PubChem [46]. Les éléments de la relation *isPartOf* sont majoritairement issus des informations fournies par PubChem.

Une autre conséquence des propriétés de la relation *isPartOf* est que toute substance possède au moins une fraction thérapeutique. En effet, pour tout élément s_1 de l'ensemble *S*, le sous-ensemble de la relation *isPartOf* contenant s_1 est aussi un ensemble partiellement ordonné [65]. Ce sous-ensemble contient donc au moins un élément minimal qui est une fraction thérapeutique. Toute substance possède donc au moins une fraction thérapeutique.

Dans RxNorm, la propriété 'has precise ingredient' lie un ingrédient (IN) à un ingrédient précis (PIN) et sa relation inverse est 'has ingredient'. Ces relations sont définies par Romedi comme des sous-ensembles de la relation *isPartOf* :

$has\_pin = \{(s_1, s_2) \in isPartOf \mid s_1 \in IN, s_2 \in S\}$,

$has\_in = has\_pin^{-1} = \{(s_1, s_2) \mid (s_2, s_1) \in has\_pin\}$

Comme toute substance possède au moins une fraction thérapeutique le domaine de la relation *has_in* est l'ensemble des substances *S*. Pour toute substance $s_1 \in S$ il existe au moins une substance $s_2 \in IN$ tel que (s_1,s_2) $\in has\_in$.

Une conséquence importante pour le modèle de Romedi est que la relation *has_in* peut être utilisée pour identifier l'ensemble des fractions thérapeutiques d'un médicament.

Un exemple de relations *isPartOf* contenant seulement 3 substances est donné figure 3.7. Pour simplifier la notation, on appelle s_1 le bromure d'ipratropium monohydraté, s_2 le bromure d'ipratropium et s_3 l'ipratropium. Dans cet exemple,

isPartOf = $\{(s_1, s_1), (s_2, s_1), (s_2, s_2), (s_3, s_2), (s_3, s_3), (s_3, s_1)\}$.

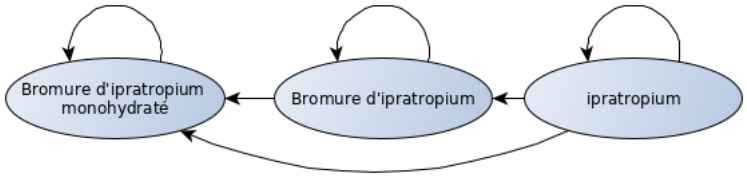


Figure 3.7: Graphe orienté représentant les relations *isPartOf* entre trois substances

s_3 est un élément minimal donc $s_3 \in$ IN, s_1 et $s_2 \in$ PIN.

has_pin = $\{(s_3, s_2), (s_3, s_3), (s_3, s_1)\}$.

has_in = $\{(s_2, s_3), (s_3, s_3), (s_1, s_3)\}$.

Dans cet exemple, toutes les substances ont pour fraction thérapeutique l'ipratropium.

Form et Route

L'ensemble *Form* contient des formes pharmaceutiques et *Route* des voies d'administrations. Les caractéristiques de libération de la substance active (libération prolongée, gastrorésistant...) ne figurent pas dans le modèle mais peuvent être prises en compte dans les éléments de l'ensemble *Form*.

Unit

L'ensemble *Unit* contient les unités utilisées pour mesurer une quantité de substance. L'ensemble des unités est partitionné en unités standards (*Unit_Standard*) et non standards. Chaque unité possède une unité standard, la fonction $conv\_unit : Unit \times Unit\_Standard \rightarrow Q$ fournit le rapport de conversion. Par exemple conv_unit(g,mg) = 1000.

Strength et Strength_ref

Strength Les éléments de l'ensemble *Strength* expriment une quantité de matière. Cet ensemble est défini comme le produit cartésien entre l'ensemble des nombres rationnels Q et l'ensemble des unités : $Strength = \{(q, u) \mid q \in Q, u \in Unit\}$

La fonction $conv\_strength : (q1, u1) \rightarrow (q2, u2)$ lie chaque élément (q1,u1) de l'ensemble *Strength* à un autre élément (q2,u2) où u2 $\in$ *Unit_Standard*. Par exemple conv_strength(1,g) = (1000,mg). Cette fonction utilise la fonction *conv_unit* et permet de standardiser la quantité de

matière.

Strength_ref Une quantité de substance active est exprimée en fonction d'une référence : *"1mg de substance pour 10mg de crème"*, *"1mg de substance dans un millilitre de solution"* ou *"1mg de substance dans un comprimé"*. La référence correspond aux autres molécules dans la forme pharmaceutique du médicament. La référence peut avoir une unité de masse (mg, ml...) ou être une unité de présentation comme "30mg par comprimé" (figure 3.4). L'unité standard 'form' est une unité spéciale définie par Romedi pour représenter une unité de présentation. Contrairement à la SNOMED CT, Romedi ne fait pas la distinction entre les différentes unités de présentation (ampoule, flacon...).

$Strength\_ref = \{(q, u) \mid q \in Q, u \in (Unit \cup \{form\}\}$

Concentration

Une concentration est composée de deux éléments : la quantité de matière de substance active et la quantité de matière de la référence.

$C = \{((q1, u1), (q2, u2)) \mid (q1, u1) \in Strength, (q2, u2) \in Strength\_Ref\}$

La fonction de standardisation des concentrations $conv\_concentration : C \rightarrow C$ lie une concentration c1 à une concentration c2 où les unités de c2 de la substance et de la référence sont des unités standards. Cette fonction permet de normaliser les unités d'une concentration. Par exemple: conv_concentration((1,gramme), (1,litre)) = ((1,mg), (1,ml)).

CDC

Clinical Drug Component (CDC) est une classe de RxNorm décrivant la quantité d'une substance présente dans un médicament. Romedi définit l'ensemble *CDC* comme le produit cartésien entre l'ensemble des substances *S* et celui des concentrations *C* :

$CDC = S \times C = \{(s, c) \mid s \in S, c \in C\}$

Cet ensemble est partitionné en deux sous-ensembles selon que la substance soit une fraction thérapeutique ou une substance active:

$CDC\_IN = \{(s, c) \mid (s, c) \in CDC, s \in IN\}$,

$CDC\_PIN = \{(s, c) \mid (s, c) \in CDC, s \in PIN\}$

$CDC = CDC\_IN \cup CDC\_PIN$ et $CDC\_IN \cap CDC\_PIN = \varnothing$

CDCS

Un médicament peut contenir une à plusieurs substances actives donc un à plusieurs 'clinical drug component'. L'ensemble $CDCS = \mathcal{P}(CDC)$ décrit toutes les combinaisons possibles de 'clinical drug component'. Cette classe n'est pas définie par RxNorm ou la SNOMED CT.

Clinical Drug

Clinical Drug (CD) est une classe de RxNorm et de la SNOMED CT décrivant une combinaison de substance(s) active(s) dans une forme galénique. Par exemple 'abacavir 300 MG / lamivudine 150 MG / zidovudine 300 MG Oral Tablet' est le libellé d'un clinical drug dans RxNorm. Son concept correspondant dans la SNOMED CT est 'Product containing precisely abacavir (as abacavir sulfate) 300 milligram and lamivudine 150 milligram and zidovudine 300 milligram/1 each conventional release oral tablet'. Les termes 'Oral tablet' et 'conventional release oral tablet' correspondent à des libellés de formes pharmaceutiques. Le modèle de Romedi définit l'ensemble *CD* comme le produit cartésien entre l'ensemble *CDCS* et *Form*:

$CD = \{(cdcs, form) \mid cdcs \in CDCS, form \in FORM\}$

Il est important de noter que le mot 'oral' décrit le site prévu d'administration, caractéristique d'une forme pharmaceutique d'après l'EDQM, et non la voie d'administration du médicament. L'ensemble *Route* n'est donc pas utilisé pour décrire un clinical drug dans Romedi.

QuantCD

Cet ensemble représente une quantité de clinical drug, regroupée dans une unité de dispensation (ampoule, flacon, comprimé...), visant à être administrée à un patient. Ce concept est similaire à celui de l'unité commune de dispensation (UCD).

$QuantCD = \{(q, cd) \mid q \in Q, drug \in CD\}$

Le facteur q porte le nom de *quantity factor* dans RxNorm. Par exemple dans *"5 ml d'une solution contenant 1mg/ml de sulfate de morphine"*, *"1mg/ml de sulfate de morphine"* est le clinical drug et 5 est le facteur de quantité. Le facteur de quantité indique la quantité de la référence (ml

dans cet exemple). Lorsqu'une concentration est exprimée par unité de dispensation ("30 mg par comprimé"), q est égal à 1.

Pack

Generic Pack est une classe de RxNorm. *Pack* permet de décrire la composition d'une boîte de médicaments. Une boîte de médicaments peut contenir un seul médicament, plusieurs fois le même médicament ou des médicaments différents. Une boîte de pilules contraceptives peut contenir des comprimés avec des dosages différents d'œstrogène et de progestérone dans une même plaquette. Certaines boîtes de médicaments contiennent plusieurs solutions différentes comme 1500ml d'acides aminés dans un premier compartiment, 2500ml d'une autre solution dans un deuxième compartiment etc... Certains médicaments sont numérotés pour une prise séquentielle. Un élément de *Pack* est défini par Romedi comme une séquence ((z_1, quantCD$_1$), ..., (z_n, quantCD$_n$)) avec $z_i \in Z$ et quantCD$_i \in$ *QuantCD*. Dans la grande majorité des cas, un pack contient le même médicament.

Marketed Drug

Marketed Drug représente la commercialisation d'une boîte de médicaments. Ce concept correspond à celui de medicinal product dans la norme ISO 11615. Un marketed drug décrit sa composition par un élément de l'ensemble Pack et par des attributs spécifiques ou non d'un pays comme un nom commercial, un nom du laboratoire, un code CIS, un code CIP13, une date de début de commercialisation, une date de retrait etc.... La voie d'administration d'un médicament est une caractéristique d'un marketed drug et non d'une forme galénique.

La voie d'administration dépend de l'indication, elle est déterminée par le fabricant. La voie d'administration et l'indication déterminent la posologie, la quantité de substances à administrer. Par exemple *"l'enoxaparine 10 000 UI (100 mg)/1 mL, solution injectable en seringue préremplie"* peut être administré par voie sous-cutanée pour le traitement prophylactique d'une maladie thromboembolique, par voie intraveineuse dans le traitement d'un syndrome coronarien aigu et dans le circuit de circulation extracorporelle au cours de l'hémodialyse. Autre exemple, *"ESTIMA 100 mg, capsule molle orale ou vaginale"* est un médicament pouvant être administré par voie orale pour traiter des déficits en progestérone et par voie vaginale pour favoriser une

grossesse.

Ensembles navigationnels

Comme RxNorm, certains ensembles sont navigationnels pour faciliter la recherche d'information. Un élément d'un ensemble navigationnel est produit par une fonction appliquée à un élément d'un ensemble construit. Un ensemble navigationnel n'est pas construit directement à partir des éléments d'une base de données.

L'ensemble $MIN = \mathcal{P}(S)$, concept décrit par RxNorm, est l'ensemble contenant toutes les combinaisons de substances (MIN signifie multiple ingredients). En sélectionnant toutes les substances d'un élément de l'ensemble *CDCS* on obtient un élément de l'ensemble *MIN*.

L'ensemble $QuantD = \mathcal{P}(S \times Strength) \times Form$ est utilisé pour décrire la quantité totale de(s) substance(s) dans un médicament (*QuantD* signifie quantified drug). Chaque élément de cet ensemble est l'image d'un élément de l'ensemble *QuantCD* par une fonction qui, à partir d'une concentration de substance active (1mg/ml) et de la quantité de la référence (5ml), calcule la quantité totale de substance dans le médicament (5mg). Par exemple, *"5 ml d'une solution à 1mg/ml de sulfate de morphine"* (*QuantCD*) contient *"5 mg de sulfate de morphine dans une solution"* (*QuantD*).

Relations d'équivalence

Le rôle d'une relation d'équivalence est de regrouper des médicaments différents qui ont des propriétés pharmacologiques similaires comme les médicaments génériques. Une relation d'équivalence R sur un ensemble A est réflexive, symétrique et transitive [65]. Soit A un ensemble de Romedi et R une relation d'équivalence sur cet ensemble, chaque élément a ∈ A appartient à une classe d'équivalence notée $[a]_R$ par la relation R. Une classe d'équivalence $[a]_R$ est l'ensemble des éléments $x \in A$ tel que x est lié à l'élément a par la relation R :
$[a]_R = \{x \in A \mid (x, a) \in R\}$

Relation d'équivalence sur l'ensemble CDC La relation d'équivalence *CDCeq* sur cet ensemble regroupe les clinical drug component (ensemble *CDC*) qui ont des fractions thérapeutiques à la même concentration. Par exemple, un *CDC* contenant 702 mg de sulfate d'abacavir

par ml appartiendra à la même classe d'équivalence qu'un CDC contenant 600 mg d'abacavir par ml.

La relation *has_in* permet d'identifier le ou les fractions thérapeutiques d'un *CDC*. La quantité de fraction thérapeutique d'une substance active doit pouvoir être calculée. La fonction $mass\_ratio$ fournit le rapport de masse entre deux substances :

$mass\_ratio : (s_1, s_2) \rightarrow x$ avec (s_1,s_2) ∈ isPartOf et x ∈]0,1]. Son domaine est la relation *isPartOf* et son codomaine un nombre rationnel compris entre 0 et 1.

Par exemple, mass_ratio(abacavir, sulfate d'abacavir) ≈ 0,856.

La fonction $equiv\_cdc : CDC \rightarrow \mathcal{P}(CDC\_IN)$ lie tout élément de l'ensemble *CDC* à un ensemble de clinical drug component $\{cdc\_in_1, ..., cdc\_in_n\}$ où chaque élément cdc_in$_i$ ∈ CDC_IN. Cette fonction permet de calculer le(s) concentration(s) de fraction(s) thérapeutique(s) d'un *CDC*. L'algorithme de cette fonction est décrit en pseudocode (algorithme 1).

Algorithm 1 Fonction $equiv\_cdc$

Input: un élément cdc (s,c) ∈ CDC,
la relation has_in,
les fonctions conv_concentration et mass_ratio
Output: un ensemble de cdc $\{cdc\_in_1, ..., cdc\_in_n\} \in \mathcal{P}(CDC\_IN)$

```
CDC_IN_s := { }
c_standard := conv_concentration(c)                    ▷ Normalisation de la concentration
INs := {in | (s,in) ∈ has_in}   ▷ Identification des fractions thérapeutiques de la substance s
for ing in INs do
    q_s, u_s, q_ref, u_ref := c_standard                ▷ Déconstruction de c_standard
    c_in := ((conv_mass(ing,s) × q_s, u_s), (q_ref, u_ref))   ▷ Calcul de la concentration de FT
    cdc_in = (ing, c_in)
    Ajout cdc_in à CDC_IN_s
end for
retourne CDC_IN_s
```

Cette fonction est utilisée par la relation d'équivalence CDCeq :

$CDCeq = \{(cdc1, cdc2) \mid cdc1, cdc2 \in CDC, equiv\_cdc(cdc1) \approx equiv\_cdc(cdc2)\}$

Le calcul des concentrations de fractions thérapeutiques conduit le plus souvent à une différence minime de masse pour deux substances actives différentes. Par exemple, 702 mg de sulfate d'abacavir contient 599.3954 mg et pas exactement 600 mg d'abacavir. Il est donc nécessaire de réaliser une approximation de(s) concentration(s) équivalente(s) de frac-

tion(s) thérapeutique(s). Dans l'implémentation, la différence de dosage autorisée entre deux molécules a été fixée arbitrairement à 5%.

Relation d'équivalence sur l'ensemble des CDCS La fonction $equiv\_cdcs : CDCS \rightarrow \mathcal{P}(CDC\_IN)$ permet de normaliser tous les clinical drug component (*CDC*) d'un médicament en identifiant les fractions thérapeutiques et en calculant leur concentration normalisée.

L'algorithme de la fonction equiv_cdcs est décrit en pseudocode (algorithme 2).

Algorithm 2 Fonction $equiv\_cdcs$

Input: un ensemble de CDC $\{cdc_1, ..., cdc_n\} \in \mathcal{P}(CDC)$,
la fonction conv_cdc
Output: un ensemble de CDC $\{cdc\_in_1, ..., cdc\_in_m\} \in \mathcal{P}(CDC\_IN)$

```
CDC_IN_s := {}
for cdc in cdcs do
    cdc_ins := conv_cdc(cdc)          ▷ Fractions thérapeutiques de chaque substance active
    for cdc_in in cdc_ins do
        if cdc_in ∈ CDC_IN_s then     ▷ Deux substances actives ont une même fraction thérapeutique
            Modifier la quantité de matière de cdc_in ∈ CDC_IN_s
        else
            Ajouter cdc_in à CDC_IN_s
        end if
    end for
end for
retourne CDC_IN_s
```

On note *CDCSeq* cette relation d'équivalence :

$CDCSeq = \{(cdcs1, cdcs2) \mid equiv\_cdcs(cdcs1) \approx equiv\_cdcs(cdcs2)\}$

Relation d'équivalence sur l'ensemble CD La relation *CDeq* est une relation d'équivalence qui regroupe les clinical drug (*CD*) qui ont les mêmes concentrations de fractions thérapeutiques et une forme galénique similaire. Il est nécessaire de définir une relation d'équivalence, *R_form* sur l'ensemble des formes pharmaceutiques. Par exemple, les formes orales à libération immédiate (comprimé, gélule...) peuvent être considérées comme une forme galénique similaire pour les médicaments génériques [45]. Soit cd_1 ($cdcs_1$, $form_1$) et cd_2 ($cdcs_2$, $form_2$) deux clinical drug.

$CDeq = \{(cd_1, cd_2) \mid (cdcs1, cdcs2) \in CDCSeq, (form_1, form_2) \in R_{form}\}$

Implémentation du modèle Le modèle de Romedi décrit plus haut a été implémenté en Java version 1.8.

3.4.2 Identification des éléments primitifs

Le modèle décrit ci-dessus est indépendant du choix des terminologies utilisées pour identifier les éléments primitifs. Une implémentation requière de réaliser des choix, présentés ci-dessous.

Identification des substances La ressource Wikidata a été utilisée pour identifier les substances. Chaque substance est identifiée par un code Wikidata. Q304330 est par exemple le code Wikidata de l'abacavir. Wikidata est utilisée comme terminologie pivot pour aligner les substances de différentes bases de données. Wikidata contient déjà de nombreux alignements vers différentes terminologies (RxNorm, UMLS, SNOMED CT...) ce qui facilite les alignements vers les ressources anglophones. Aligner une substance vers Wikidata permet très souvent d'obtenir l'alignement vers RxNorm. L'autre avantage de Wikidata est qu'il est possible de créer un nouvel item pour décrire une substance si celle-ci n'existe pas et d'ajouter des alignements manquants via son interface en ligne ou ses API. La gestion collaborative de Wikidata facilite la maintenance des alignements vers les ressources externes.

Relation isPartOf Les relations de partonomie entre deux molécules, comme la relation entre "sulfate d'abacavir" et "abacavir", ont été extraites de PubChem et parfois ajoutées manuellement. Les formules chimiques et les masses moléculaires des substances ont été extraites de Wikidata. La fonction *mass_ratio* du modèle utilise les masses moléculaires pour calculer le rapport de masse entre deux molécules de la relation *isPartOf*.

Identification des unités La terminologie UCUM (Unified Code for Units of Measure) a été utilisée pour standardiser les unités de mesure. Les unités standards choisies sont celles de RxNorm : la masse d'une substance est exprimée en milligramme, celle d'une solution en millilitre. L'unité standard de certaines molécules est une unité internationale dont la définition dépend de chacune des substances. Ainsi 1 UI d'insuline correspond à 0,0347 mg d'insuline et 1 UI de vitamine A correspond à 0,3 ug de rétinol.

Identification des formes pharmaceutiques et des voies d'administration Les identifiants des formes pharmaceutiques fournits par l'EDQM ont été utilisés en retirant les formes pharmaceutiques multiples ou combinés. Par exemple, si la forme pharmaceutique était "poudre et solvant pour solution injectable" décrivant plusieurs formes, seul le code du terme "solution injectable" était conservé.

Relation d'équivalence entre formes pharmaceutiques Les formes pharmaceutiques ont d'abord été regroupées dans leur forme pharmaceutique basique (comprimé, gélule...). Dans les données de l'EDQM, chaque forme pharmaceutique possède une forme pharmaceutique basique. Ensuite, certaines formes pharmaceutiques solides orales ont été regroupées comme décrit par l'article L.5121-1 du code de la santé publique sur les médicaments génériques. La relation d'équivalence a regroupé les formes pharmaceutiques qui avaient les mêmes formes basiques ou qui étaient des formes solides orales.

3.4.3 Normalisation, instanciation et chargement des données

Chaque base de données médicamenteuses est normalisée, instanciée dans le modèle puis chargée dans une base de données graphe, un triplstore. La figure 3.8 montre ces étapes pour l'exemple d'un médicament de Medicabase du tableau 3.2.

Normalisation

La première étape de normalisation consiste à séparer les différents éléments d'un libellé : substance, dosage, forme galénique. La deuxième étape de normalisation consiste à aligner ces éléments vers les éléments primitifs du modèle de Romedi : les libellés des substances vers les codes Wikidata, les libellés des formes galéniques vers les codes EDQM etc.... Dans RxNorm, un groupe d'experts est en charge de normaliser les informations relatives aux médicaments. Le répertoire des médicaments de l'ANSM est moins normalisé et requière beaucoup plus d'efforts. La normalisation des données a été réalisée par des expressions régulières avec le langage R[66]. Les alignements des substances vers Wikidata ont été réalisés semi-automatiquement par méthodes de similarité de chaînes de caractères et en utilisant le SPARQL endpoint de Wikidata.

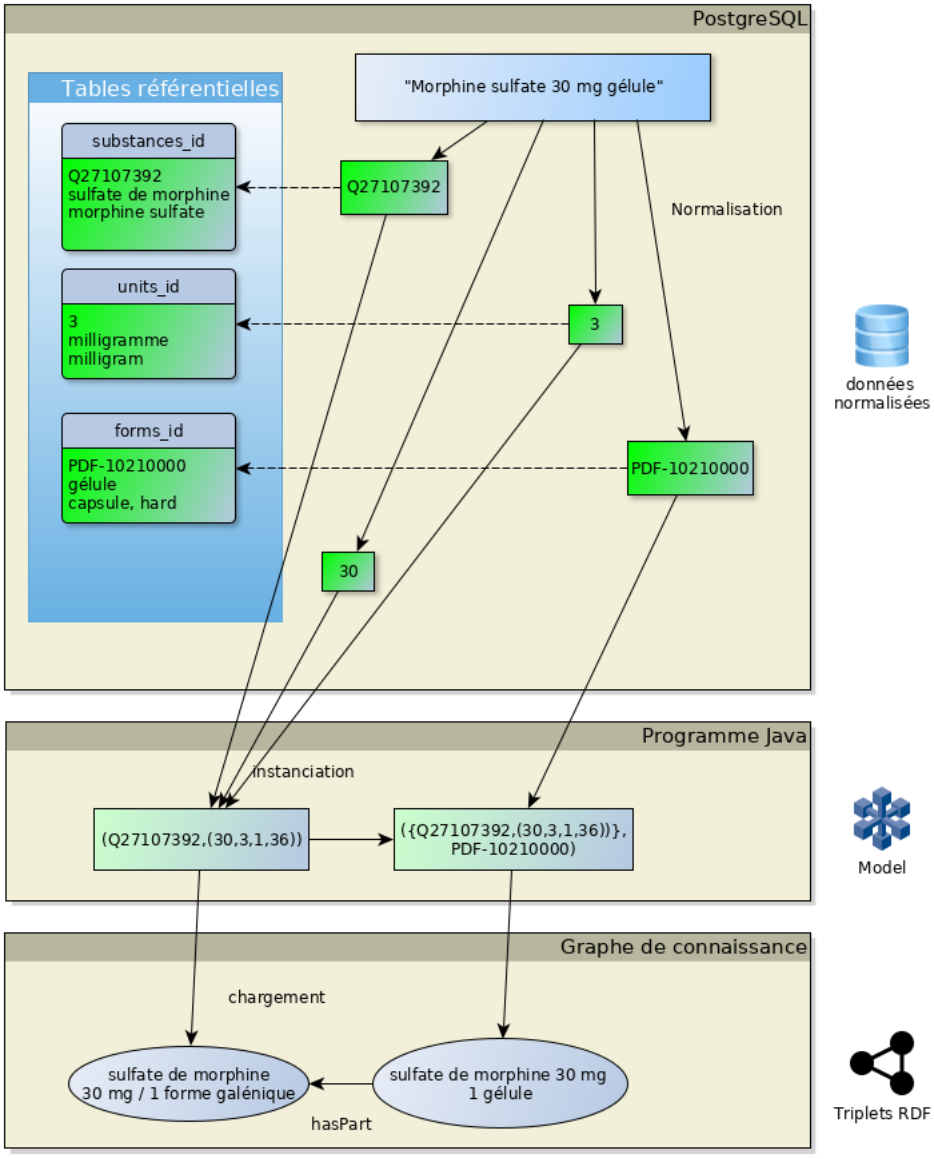


Figure 3.8: Exemple de normalisation d'un libellé issu de la base de données Medicabase, d'instanciation du modèle et de chargement dans un triplestore. Flèches en pointillé : clefs étrangères

Les éléments des ensembles primitifs (tableau 3.4) ont été stockés dans des tables d'une base de données PostgreSQL version 13.1. Les tables référentielles possèdent une structure similaire : un identifiant unique, un libellé préféré français et un libellé préféré en anglais (figure 3.8). Les données normalisées ont été chargées dans PostgreSQL en établissant des liens vers les tables référentielles par des clefs étrangères. Ces clefs étrangères assurent une intégrité référentielle : lors du chargement des données dans le modèle, les données normalisées sont consistantes avec les identifiants des éléments primitifs, c'est-à-dire qu'il n'existe aucun identifiant dans les bases de données qui ne soit pas un identifiant d'un élément primitif.

Instanciation

Les éléments primitifs (tableau 3.4) sont chargées en premier dans le modèle ainsi que les éléments de la relation *isPartOf* et la fonction de conversion d'unités permettant de normaliser les concentrations. Le programme Java charge ensuite une base de données médicamenteuses normalisée et l'instancie. L'instanciation consiste à identifier automatiquement les substances actives et les fractions thérapeutiques d'un médicament, à normaliser les dosages et les concentrations, à calculer la ou les concentrations de fraction(s) thérapeutique(s). Plusieurs instances de classes construites (tableau 3.5) sont générées pour chaque médicament. Le programme attribue à chaque instance un code unique qui correspond à la notation ensembliste de l'instance. Par exemple (Q27107392,((30,16),(1,36))) identifie un clinical drug component où la substance est Q27107392 (sulfate de morphine) et la concentration ((30,16),(1,36)) : (30,16) identifie le numérateur de la concentration (30 mg) et (1,36) le dénominateur (une unité de dispensation). Chaque identifiant d'un élément primitif possède un libellé préféré français ou anglais. Comme chaque élément construit est une combinaison d'éléments primitifs, il est possible de générer un libellé français et anglais pour chaque élément construit. Par exemple, l'élément (Q27107392,((30,16),(1,36))) a pour libellé français *sulfate de morphine 30 milligramme / 1 forme galénique* et pour libellé anglais *morphine sulfate 30 milligram / 1 dosage form*. Des libellés français et anglais sont ainsi créés automatiquement pour chaque élément créé permettant par la suite de visualiser une base de données médicamenteuses dans des langues différentes. Les instances créées par le programme Java sont ensuite transformées sous forme de triplets RDF qui sont chargés dans un triplestore. Le modèle génère les classes d'équivalence quand

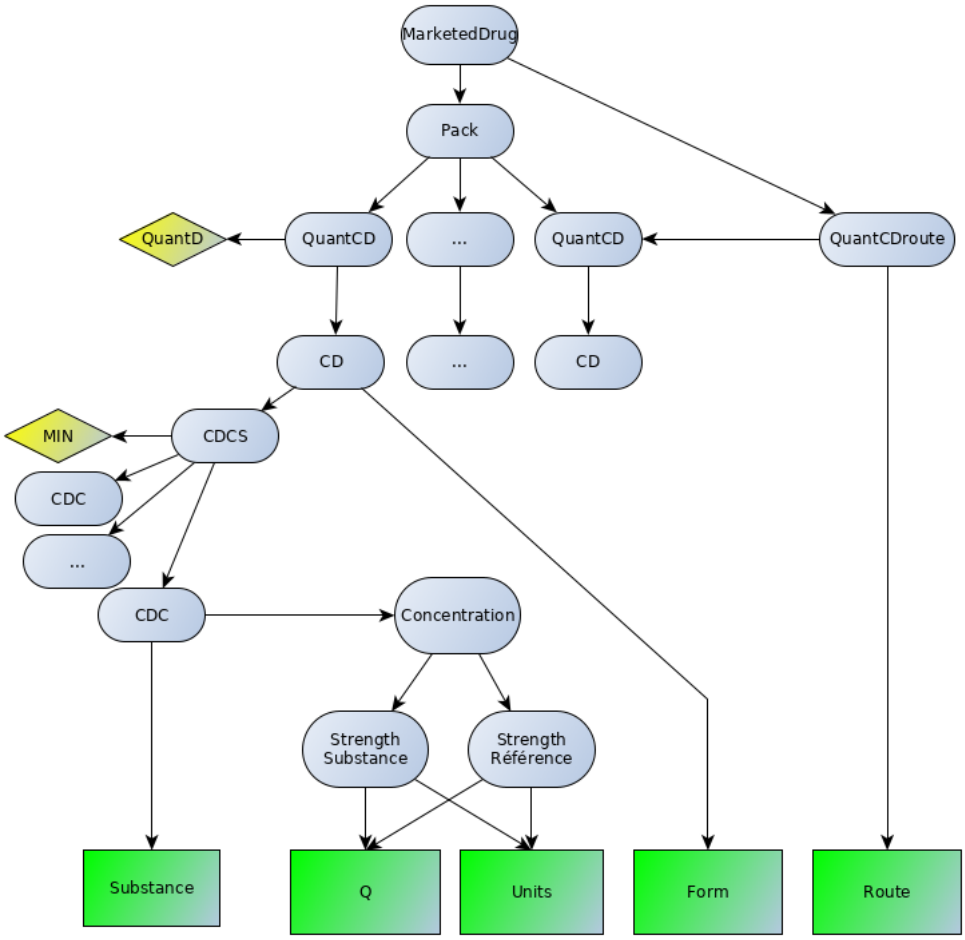


Figure 3.9: Schéma du graphe de connaissance de Romedi. Les concepts primitifs figurent dans des rectangles verts, les concepts construits dans des nœuds gris, les concepts navigationnels dans des nœuds jaunes. Chaque flèche correspond à la propriété *hasPart*. Une flèche entre un nœud A et un nœud B signifie que la paire (A,B) appartient à la relation *hasPart*. CDC : clinical drug component, CD : clinical drug, QuantCD : quantified clinical drug, QuantD : quantified drug, MIN : multiple ingredients

toutes les données ont été chargées dans le graphe de connaissance.

Graphe de connaissance

Un graphe de connaissance est une base de connaissance où les informations sont stockées dans un graphe [67]. Le graphe de connaissance généré par le modèle de Romedi forme un graphe orienté (figure 3.9). Les nœuds du graphe correspondent aux instances créées par le modèle. La propriété *hasPart*[12] est utilisée pour lier deux nœuds A et B qui représentent un élément A construit avec un élément B. Chaque nœud est relié

[12] `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/#hasPart`

par la propriété "creator" à au moins un nœud d'une base de données médicamenteuses dont est issue cette instance (figure 3.12). Chaque nœud du graphe est identifié par une URI. Les URI des nœuds primitifs sont générés avec leur identifiant. Par exemple, http://www.romedi.fr/romedi/Q27107392 est l'URI identifiant le sulfate de morphine dans Romedi. Les URI des nœuds construits sont créés par hachage de leur identifiant. Par exemple, le hash du code (Q27107392,((30,16),(1,36))) est j88pgd2m1lshg donc l'URI de ce clinical drug component est http://www.romedi.fr/romedi/CDC_PINj88pgd2m1lshg.

3.4.4 Recherche d'alignements terminologiques

Trouver des alignements correspond à rechercher des éléments communs à deux bases de données médicamenteuses ou des éléments liés par une relation d'équivalence.

Alignements par correspondance exacte

Le modèle de Romedi décrit un ensemble fini de n ensembles primitifs, construits et navigationnels. Chacun de ces ensembles est noté E_i avec i compris entre 1 à n. On définit RomediElements comme l'ensemble des éléments de chacun des ensembles E_i :
$RomediElements = \bigcup_{i=1}^{n} E_i$

Une base de données, notée BD, chargée dans le modèle génère un ensemble fini d'éléments noté BDelements : $BDelements \subset RomediElements$. Soit deux bases de données, BD1 et BD2, générant les ensembles BD1elements et BD2elements respectivement. Les éléments en commun correspondent à l'intersection de ces deux ensembles. $Alignements12 = BD1elements \cap BD2elements$. Ces éléments en commun sont des alignements par correspondance exacte.

Dans le graphe de connaissance, la propriété "dc:creator" associe une instance construite par le modèle à la base de données médicamenteuses dont elle est issue. Lorsqu'une instance est produite par plusieurs bases de données, elle possède plusieurs liens "dc:creator" (figure 3.12). Pour identifier l'intersection des instances de deux bases de données, il suffit de requêter le graphe de connaissance à la recherche de nœuds possédant plusieurs liens "dc:creator".

Relations d'équivalence

La deuxième façon de trouver des alignements est de trouver des équivalences pharmacologiques.

Dans le résumé des caractéristiques du produit du médicament *MORPHINE (SULFATE) LAVOISIER 1 mg/ml, solution injectable*[13], il est noté dans la section posologie :

"il est rappelé qu'UN mg de sulfate de morphine équivaut à UN mg de chlorhydrate de morphine".

Il existe toutefois une légère différence car 1 mg de sulfate de morphine et 1 mg de chlorhydrate de morphine contiennent respectivement 0,75 mg et 0.76 mg de morphine. Cette différence conduit à la création de deux nœuds différents dans le graphe de connaissance. En acceptant une différence de masse de 5% entre deux fractions thérapeutiques, une relation d'équivalence permet de regrouper ces deux clinical drug component.

Soit R une relation d'équivalence sur un ensemble de Romedi. Deux éléments e_1 et e_2, $e_1 \in BD1elements$ et $e_2 \in BD2elements$ sont dits équivalents s'il existe une relation R d'équivalence tel que e_1 appartient à la classe d'équivalence de e_2 par la relation R : $e1 \in [e2]_R$.

3.4.5 Détection et identification des médicaments dans les documents textuels

Dans le contexte des données hospitalières, les médicaments mentionnés sont très majoritairement des médicaments commercialisés en France. A de rares exceptions, il peut s'agir de médicaments étrangers et de nouvelles molécules utilisées dans un essai thérapeutique. Notre hypothèse était que le problème d'identification des médicaments dans un EDS pouvait être résolu avec une terminologie exhaustive des médicaments commercialisés en France. Comme le répertoire des médicaments de l'ANSM est une base de données exhaustive des médicaments commercialisés ou ayant été commercialisés en France, un médicament mentionné dans un DPI est probablement présent dans cette base de données et donc dans le graphe de connaissance de Romedi.

[13] `https://base-donnees-publique.medicaments.gouv.fr/affichageDoc.php?specid=60651453&typedoc=R`

Une étude a été menée au CHU de Bordeaux pour évaluer les performances de Romedi pour l'identification des médicaments dans les formulaires DxCare®. Cette étude a été menée sur des données textuelles du service des urgences adultes de l'hôpital Pellegrin de Bordeaux. Un formulaire informatisé est utilisé par les urgentistes pour recueillir les informations du patient à son arrivée. Le formulaire est composé d'une section "Traitements" pour saisir le traitement actuel du patient (figure 1.1).

Aucune aide à la structuration n'est proposée lors de la saisie. Les professionnels de santé entrent la liste de traitements pris par le patient, le plus souvent en notant seulement le nom commercial ou la substance en dénomination commune internationale. Un dictionnaire a été créé en extrayant du graphe de connaissance de Romedi les noms commerciaux et les fractions thérapeutiques. La méthode d'identification des médicaments en texte libre a consisté à annoter les documents avec IAMsystem, un annotateur sémantique présenté au chapitre 4.

Mille formulaires de patients âgés de plus de 75 ans ont été tirés au sort pour constituer un gold standard. Ce critère d'âge a été choisi car ces patients prennent souvent de multiples traitements et qu'une étude pharmacologique sur les interactions médicamenteuses était en cours. Deux personnes, un médecin et un pharmacien, ont annoté les médicaments dans le texte libre avec le logiciel Brat [68] sans inclure le dosage ou la fréquence de prise du médicament lorsque ces informations étaient mentionnées. Brat est un logiciel open source très utilisé en TAL pour annoter manuellement des documents afin de créer un gold standard. Une capture d'écran de Brat est disponible dans un autre chapitre à la figure 4.5.

Les performances de l'algorithme ont été évaluées par les mesures standards en recherche d'information : rappel, précision et F_1-score. Le rappel correspond au pourcentage de médicaments correctement identifiés dans les formulaires. La précision correspond au rapport du nombre de médicaments correctement identifiés sur le nombre total de médicament identifiés par l'algorithme. Le F_1-score est la moyenne harmonique du rappel et de la précision.

3.5 Résultats

Le référentiel est accessible en ligne sur le site `https://www.romedi.fr`. Une interface similaire à RxNav (figure 3.2) a été développée pour naviguer dans les données de Romedi.

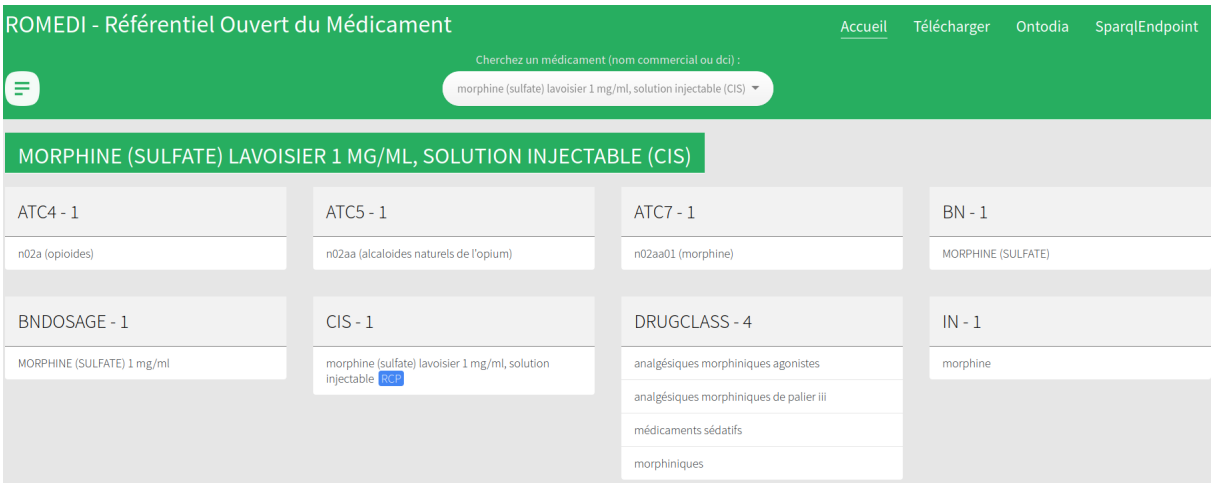


Figure 3.10: Interface de Romedi permettant de naviguer dans les données françaises. Cette interface ressemble à RxNav permettant de naviguer dans les données de RxNorm

La figure 3.10 montre la visualisation du médicament *morphine (sulfate) lavoisier 1 mg/ml, solution injectable (CIS)* commercialisé en France avec le code CIS 60651453. Ce médicament a l'URI `http://www.romedi.fr/romedi/CIS60651453`. Cette URI est déférencée par un navigateur et décrite dans un format json en ajouant le suffixe "?json" à l'url.

La figure 3.11 montre la visualisation de ce même médicament dans l'interface de Ontodia [69] qui permet de visualiser le graphe RDF et de naviguer dans son contenu. Un SPARQL endpoint est disponible pour interroger les données au format RDF.

3.5.1 Alignements terminologiques

Parmi les 4920 substances décrites dans le répertoire des médicaments 3785 (77%) ont été alignées vers un code Wikidata. Les substances non alignées sont majoritairement des médicaments à base de plantes ou des vaccins. Les relations vers RxNorm de Wikidata ont été utilisées pour normaliser les substances de OMOP. La figure 3.12 montre que le *1 gélule de sulfate de morphine 30mg dans une forme galénique* est présent à la fois dans la base de données de l'ANSM et dans Medicabase. Ce nœud permet d'établir un lien entre les données de l'ANSM et de Medicabase.

Des liens entre différentes terminologies sont parfois trouvés entre des concepts différents. Par exemple, *"3000 MG Dinoprostone 0.000667 MG/MG Vaginal Gel"* (QuantCD) défini par RxNorm est lié à *"Dinoprostone 2 mg gel vaginal"* (QuantD) défini par Medicabase car le pre-

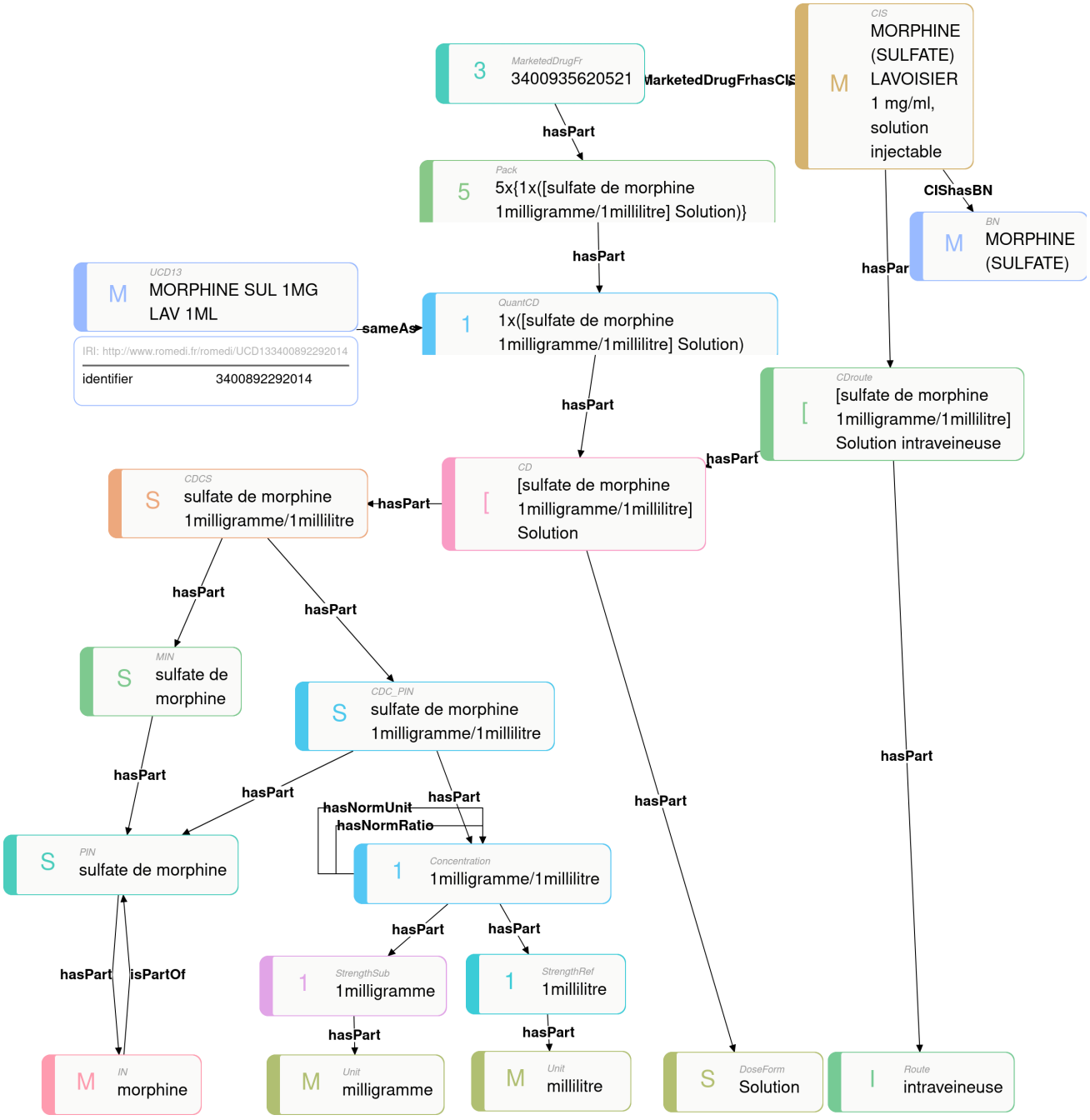


Figure 3.11: Visualisation dans l'interface Ontodia de triplets RDF issus du graphe de connaissance. Ces triplets décrivent le médicament *morphine (sulfate) lavoisier 1 mg/ml, solution injectable* commercialisé en France, il contient 5 ampoules de 1 ml de sulfate de morphine à la concentration de 1 mg par ml

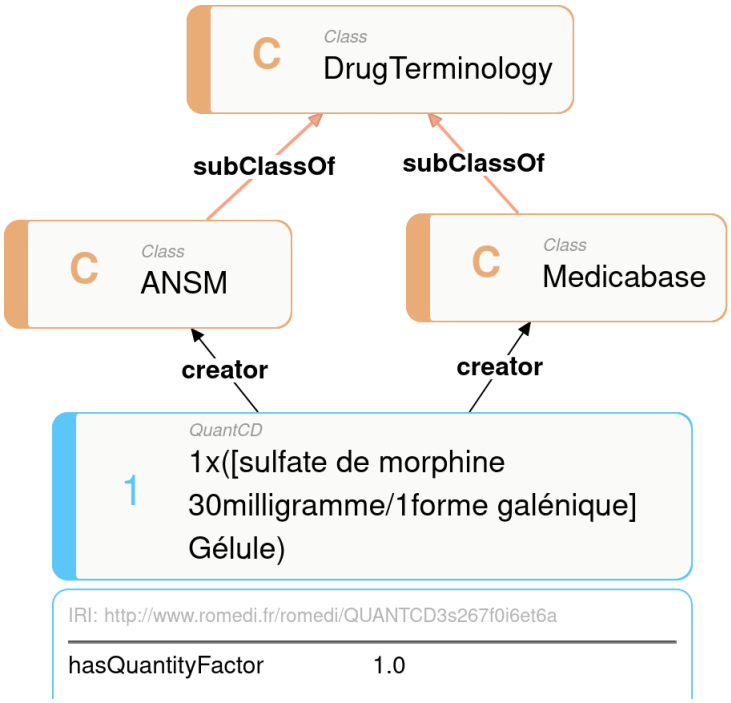


Figure 3.12: Visualisation du quantified clinical drug '1 gélule de sulfate de morphine 30mg dans une gélule' dans l'interface Ontodia. Cette instance est définie par les bases de données de l'ANSM et par Medicabase. Son URI est `http://www.romedi.fr/romedi/QUANTCD3s267f0i6et6a`

mier élément contient exactement 2mg de dinoprostone en gel vaginal. Ce lien est réalisé car le modèle de Romedi déduit du premier élément le même quantified drug. Dans cet exemple, RxNorm décrit le médicament par sa concentration tandis que Medicabase le décrit par sa quantité totale de substance.

La figure 3.13 montre un exemple de classe d'équivalence sur l'ensemble des clinical drug permettant d'identifier des médicaments ayant des concentrations proches de morphine dans une forme pharmaceutique similaire.

Alignements des médicaments de ville vers OMOP

Pour réaliser les alignements vers OMOP, seuls les médicaments présents dans la base de données OpenMedic décrivant le nombre de boîtes commercialisées en France ont été utilisés. Cette base contenait 12 508 médicaments. Le modèle de Romedi trouve un alignement potentiel pour 77% des médicaments français. Par exemple *"MOSCONTIN LP 10MG CPR"* est aligné à *"Morphine 7.5 MG Extended Release Oral Tablet"* par une relation d'équivalence car le moscontin contient 10mg de sulfate de morphine en substance active ce qui correspond à 7.5 mg de mor-

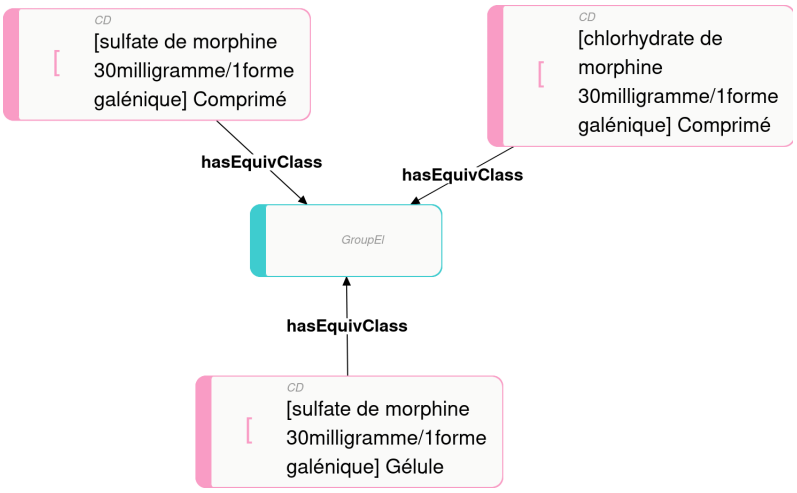


Figure 3.13: Une classe d'équivalence sur l'ensemble des clinical drug par la relation CDeq. Cette classe d'équivalence regroupe trois clinical drug distincts qui différent sur la forme pharmaceutique (comprimé, gélule) et sur la substance active (sulfate et chlorhydrate de morphine)

phine en fraction thérapeutique. Le médicament *"ZYMAD 80000UI SOL BUV AMP 1"* est aligné par correspondance exacte à *"2 ML Cholecalciferol 40000 UNT/ML Oral Solution"* car ces deux médicaments contiennent 2 millilitres d'une solution à 40 000 unités par millilitre. Le libellé *"ZYMAD 80000UI SOL BUV AMP 1"* indique la quantité totale de vitamine D contenue dans la solution tandis que les données de l'ANSM indiquent que l'ampoule a un volume de 2ml, donc la concentration est de 40000 UNT/ML. Aligner les données de remboursement l'assurance maladie vers OMOP nécessite l'utilisation du répertoire des médicaments de l'ANSM car l'information contenue dans les libellés est insuffisante.

Alignements des médicaments hospitaliers vers OMOP

La majorité des médicaments internes à l'hôpital possède un code UCD13. Un médicament hospitalier est donc un quantified clinical drug (QuantCD) dans le modèle de Romedi. Au total 8236 (72%) codes NIMED (code du référentiel interne) ont un alignement vers un code OMOP dont 84% sont des alignements exacts et 16% appartiennent à une même relation d'équivalence. Par exemple le médicament *COVERSYL 10MG CPR* contenant 10 mg de périndopril arginine n'est pas décrit dans RxNorm ni OMOP. Le modèle de Romedi identifie qu'il appartient à la même classe d'équivalence que le médicament *perindopril erbumine 8 MG Oral Tablet* décrit dans RxNorm car ils contiennent 6,79 mg et 6,68 mg de périndopril, respectivement. Ces quantités de fractions thérapeutiques sont déduites automatiquement du modèle de Romedi, via

la relation *isPartOf*, elles ne figurent pas dans RxNorm.

3.5.2 Identification des médicaments dans les données cliniques

Au total, le gold standard contenait 6581 médicaments (substance ou nom commercial) à détecter dans les 1000 formulaires. 6070 médicaments ont été détectés par notre programme (rappel : 0,92) pour 6 094 propositions (précision : 0,99 ; F-mesure : 0,95). La variation lexicale la plus fréquente concernait le "Diffu K", souvent orthographié "DiffuK". Parmi les médicaments non détectés, certaines erreurs étaient liées à des abréviations non prises en compte. Les cliniciens utilisent parfois "KDG" pour désigner le nom commercial "Kardegic", "ac" pour désigner le mot acide (ex : "ac acétylsalicylique") et "vit" pour le mot vitamine (ex : "vit D"). D'autres erreurs étaient liées à des défauts de normalisation. Par exemple, "SYMBICORT TURBUHALER 100/6 microgrammes par dose, poudre pour inhalation" a pour nom commercial "SYMBICORT TURBUHALER" dans notre référentiel alors que les professionnels de santé le dénomment "SYMBICORT".

3.6 Discussion

Nous proposons un nouveau modèle pour représenter un médicament et chercher des équivalences pharmacologiques. Une originalité du modèle de Romedi est que les fractions thérapeutiques (ingrédients dans RxNorm et la SNOMED CT) et les substances actives (precise ingredient dans RxNorm et la SNOMED CT) ne sont pas des concepts primitifs mais des concepts définis par la relation *isPartOf*. Ces concepts sont déduits des propriétés de la relation *isPartOf* sur l'ensemble des substances. Cette classification automatique des substances permet de gérer des conflits de classification inter-référentiels et certaines incohérences dans la base de données du répertoire de médicaments de l'ANSM où une substance peut être à la fois une fraction thérapeutique et une substance active. La relation *isPartOf* et les masses moléculaires permettent de calculer les quantités de fractions thérapeutiques d'un médicament et ainsi de compléter des données manquantes comme celles du tableau 3.1. Comme les résultats l'ont montré, compléter des données manquantes est nécessaire pour trouver des alignements entre bases de données différentes. A notre connaissance, les autres modèles basés sur les logiques de description

ne permettent pas de mettre en évidence automatiquement des équivalences pharmacologiques.

Une autre originalité de Romedi est l'utilisation de Wikidata, une ressource collaborative, pour identifier les substances. La maintenance des liens vers les ressources externes est ainsi assurée par la communauté des utilisateurs de Wikidata. Toute information incorrecte dans Romedi portant sur le libellé, sa formule chimique ou un lien externe à une base de données serait causée par une erreur dans Wikidata et donc facilement modifiable dans l'interface Wikidata.

L'instanciation du modèle avec plusieurs bases de données médicamenteuses, françaises et américaines, a permis de montrer que le modèle est suffisamment générique pour intégrer des données hétérogènes. Son modèle s'inspire du modèle américain RxNorm et celui de la SNOMED CT ; il existe donc de nombreuses similitudes. Certains concepts comme celui de substance, de dosage (strength), de forme pharmacologique, de clinical drug component, de clinical drug et de pack sont ré-utilisés. Comme dans la SNOMED CT, Romedi exprime un dosage en concentration (30mg/ml) ou en unité de présentation (30mg par comprimé). Contrairement à la SNOMED CT, l'unité de présentation est un élément unique tandis que la SNOMED CT décrit de nombreuses unités de présentation (comprimé, gélule, flacon, ampoule...). Pour les formes solides (comprimé, gélule...), l'information sur l'unité de représentation est redondante avec celle de la forme pharmaceutique ce qui n'entraîne pas de différence avec la SNOMED CT. Pour les solutions, l'unité de présentation permet de différencier des conteneurs différents (ampoule, flacon, seringue pré-remplie...). En n'intégrant pas cette information dans son modèle, Romedi regroupe des médicaments qui ont des présentations différentes. Ce défaut de granularité a peu d'importance pour les études pharmacologiques.

Une différence majeure avec la SNOMED CT est que Romedi n'utilise pas de logique de description. L'absence de formalisation par une logique de description ne permet pas à une machine de réaliser des raisonnements et de découvrir de nouveaux faits. Le rôle du modèle de Romedi est de normaliser des données hétérogènes, de compléter des données manquantes et d'identifier des relations d'équivalence. Le résultat final du modèle est un graphe de connaissance au format RDF sur lequel des règles logiques pourraient être ajoutées pour réaliser des raisonnements. Le modèle de Romedi n'est donc pas opposé au modèle de la SNOMED CT mais plutôt complémentaire. La normalisation des données avec des concepts similaires à ceux

de la SNOMED CT devrait faciliter l'instanciation des données françaises dans la SNOMED CT.

Une terminologie issue de Romedi, contenant les noms commerciaux et les ingrédients des médicaments en langue française extraits du graphe de connaissance, offre de bons résultats pour identifier les médicaments dans les formulaires DxCare®. La bonne sensibilité s'explique par le contenu exhaustif du répertoire des médicaments de l'ANSM qui contient l'intégralité des médicaments commercialisés ou ayant été commercialisés en France. L'excellente spécificité s'explique par une faible ambiguïté dans les données des formulaires. Dans les documents plus longs, une désambiguïsation est nécessaire car certains ingrédients sont aussi des examens biologiques (cholestérol, potassium, albumine...). L'identification automatique des médicaments dans un EDS permet leur structuration et facilite leur interrogation. Les URI du graphe de connaissance fournissent des identifiants pour annoter et indexer la mention d'un médicament dans les documents en texte libre.

Le modèle présente toutefois plusieurs limites. La principale limite est la normalisation automatique des données par des scripts conduisant à des erreurs. Certaines erreurs sont détectées au moment de son utilisation et corrigées manuellement. Une normalisation manuelle des données françaises nécessiterait une équipe d'experts dédiés comme pour RxNorm. L'agence du numérique en santé (ANS) est en train de construire un référentiel unique d'interopérabilité du médicament où les données de l'ANSM seront normalisées[14]. Ce référentiel pourra pallier les défauts de normalisation du répertoire de l'ANSM. La qualité actuelle des données intégrées dans Romedi est donc très inférieure à celle d'un référentiel comme RxNorm utilisé pour l'interopérabilité sémantique des systèmes d'information aux États-Unis. La maintenance de son contenu est aussi un problème car les différentes bases de données intégrées sont mises à jour régulièrement.

3.7 Conclusion

L'extraction d'information sur les médicaments est importante pour la réalisation d'études pharmacologiques. Le modèle de Romedi permet d'intégrer de multiples référentiels sur les médica-

[14] https://esante.gouv.fr/espace-presse/vers-le-referentiel-unique-dinteroperabilite-du-medicament

ments dans un graphe unifié. Romedi fournit aussi un dictionnaire exhaustif des médicaments pour les identifier dans les documents médicaux. Les traitements habituels d'un patient mentionnés dans des documents courts comme les formulaires DxCare® peuvent être structurés automatiquement. Le graphe de connaissance permet d'extraire des connaissances externes pour classifier les médicaments selon leurs propriétés pharmacologiques. Ainsi, les chercheurs peuvent rechercher des patients dans l'EDS qui prennent une certaine classe de médicaments comme des anti-diabétiques. Cette fonctionnalité facilite l'identification de cohortes de patients et l'extraction d'information dans un DPI.

Chapitre 4

Annotation sémantique

4.1 Introduction

La majorité des informations contenues dans un dossier patient informatisé (DPI) sont présentes sous forme de texte libre [4]. Le texte libre possède de nombreux avantages pour les professionnels de santé tels que la familiarité, la facilité d'utilisation et la liberté d'exprimer des concepts complexes [70]. Une étape fréquente dans une pipeline de traitement automatique de la langue (TAL) est la détection d'entités médicales (traitement, antécédents médicaux...) avec des algorithmes de reconnaissance d'entités nommées (NER). Il est nécessaire de lier chaque entité détectée à une terminologie ou à une ontologie pour tirer parti des graphes de connaissances qui apportent des informations supplémentaires et du sens aux termes détectés dans un texte [60]. L'action de relier le contenu d'un texte à un graphe de connaissance porte le nom d'annotation sémantique, *entity linking* en anglais [60]. Le chapitre précédent a illustré l'utilité d'une terminologie exhaustive sur les médicaments pour détecter et identifier la mention des médicaments dans les documents textuels. L'indexation de millions de documents avec des terminologies médicales contenant plusieurs centaines de milliers de termes est un défi technique pour tout entrepôt de données (EDS). Ce chapitre présente un nouvel algorithme d'annotation sémantique, IAMsystem, adapté aux contraintes d'un EDS.

La section suivante définit formellement l'annotation sémantique et décrit les travaux connexes. La section Méthodes détaille le fonctionnement d'IAMsystem. La section Résultats présente les performances de l'algorithme lors de sa participation à trois tâches partagées.

4.2 Travaux connexes

4.2.1 Formalisation du problème

Définitions

Les mots utilisés par la suite sont définis ici. Ces définitions sont issues de l'article de Tseylin et al. [71] :

- Un token est une séquence de chaînes de caractères d'un texte, en général un mot. Dans ce manuscrit, "token" et "mot" sont utilisés de façon interchangeable.
- Un terme est un ou plusieurs mots représentant un concept.
- Un concept est une représentation mentale d'un objet, d'une idée.
- Une terminologie est un ensemble de termes désignant des concepts dans un domaine précis.
- Un variant lexical est la variation d'un mot liée à la prononciation ou à la grammaire (singulier, pluriel, forme conjuguée...).
- Une abréviation est une forme raccourcie d'un mot.
- Un synonyme est un terme ayant le même sens qu'un autre terme.
- Une annotation est l'action de relier une séquence de mots d'un document à un concept d'une terminologie.

L'annotation sémantique consiste à relier des séquences de mots d'un document aux concepts d'une terminologie [72]. Une terminologie est composée d'un ensemble S de termes appelé dictionnaire, d'un ensemble C de concepts, chacun identifié par un code unique et d'une relation $R \subset S \times C$ entre les termes et les concepts.

Après une étape de normalisation et de tokenisation, un document *d* est une séquence de mots (w_1, ..., w_n). L'annotation sémantique consiste à identifier un ensemble de paires $((w_i, ..., w_j), c_k)$ où $(w_i, ..., w_j)$ est une séquence de un à plusieurs mots dans *d* et c_k un concept appartenant à C. Cette séquence de mots est continue ou discontinue. Une séquence de mots $(w_i, ..., w_j)$ est dite continue si et seulement si, pour tout nombre entier k, i ≤ k ≤ j, w_k appartient à la séquence $(w_i, ..., w_j)$.

Dans une approche à base de dictionnaire, l'ensemble des termes du dictionnaire S est utilisé pour identifier les séquences de mots du document représentant un concept. Un terme d'un dictionnaire peut être mentionné différemment dans un document pour de multiples raisons : abréviations, variations lexicales (singulier, pluriel, conjugaison...), fautes d'orthographe, utilisation de synonymes, présence de mots vides ou mots discontinus.

4.2.2 Algorithmes d'annotation sémantique

En 2017, Jovanovic et al. ont réalisé une revue des principaux outils d'annotation sémantique [60]. Chaque outil possède des fonctionnalités différentes, certains sont spécifiques d'un dictionnaire tandis que d'autres sont généralistes, la majorité des outils analysés sont modulaires et configurables. D'après les auteurs, il est difficile de comparer les différents outils sans benchmark commun et une voie de recherche à explorer est l'amélioration de la rapidité des annotateurs afin de traiter efficacement des millions de documents. Quelques annotateurs sémantiques sont présentés ci-dessous pour illustrer leurs différences et positionner IAMsystem dans ce contexte.

MetaMap [73] est l'un des annotateurs les plus connus et cités dans le domaine biomédical. Il a été développé par le NIH dans les années 1990 et ne fonctionne qu'en anglais et avec l'UMLS (Unified Medical Language System), un regroupement de nombreux vocabulaires contrôlés dans le domaine biomédical [74]. MetaMap réalise plusieurs traitements pour chaque document : étiquetage morpho-syntaxique, générations de variations lexicales pour chacun des mots (oculaire => (oeil, ophtalmique)), recherche de concepts candidats dans l'UMLS contenant au moins un mot en commun avec celui du document et enfin calcul d'un score pour chaque concept [75]. MetaMap favorise le rappel à la précision, son temps de traitement est relativement long par rapport aux autres approches, il est peu adapté à un grand corpus de documents [75].

cTakes est un logiciel de TAL spécialisé dans le domaine médical et développé par la Mayo Clinic [76]. cTakes est composé de plusieurs modules de TAL dont un module d'annotation sémantique. L'article de 2010 explique que l'annotation sémantique utilise une méthode à base de dictionnaire en se limitant aux syntagmes nominaux d'un document, c'est-à-dire qu'une

étape préalable d'étiquetage grammatical est réalisée. La documentation de la version actuelle de l'outil (version 4.0) explique que la méthode d'indexation des termes du dictionnaire a été modifiée et que la fenêtre de recherche a été étendue à la phrase plutôt qu'aux syntagmes nominaux. Pour chaque mot d'une phrase, des termes candidats sont recherchés dans le dictionnaire puis l'algorithme recherche la présence d'un terme candidat dans la phrase en prenant en compte des variations lexicales de chaque mot et des possibles permutations dans l'ordre des mots.

QuickUMLS est un annotateur basé sur la similarité entre deux chaînes de caractères [77]. Étant donné un document *d* de longueur n, un seuil de similarité α et une fenêtre w, QuickUMLS génère pour chaque token $d_i \in d$, toutes les séquences possibles de tokens $d_{ij} = \{d_i,..., d_j\}$ avec $j \in \{i,..., i + w - 1\}$. Pour chaque séquence générée, l'algorithme CPMerge de Okazaki et Tsujii [78] est appelé pour rechercher une chaîne de caractères similaire dans le dictionnaire. CPMerge est paramétré avec un seuil de similarité α compris entre 0 et 1, 1 correspondant à un alignement parfait. En diminuant α, la sensibilité augmente et la spécificité diminue. Les auteurs se sont intéressés à la rapidité de détection et revendiquent une vitesse 2,5 à 135 fois supérieure à MetaMap ou cTakes avec des performances similaires sur un jeu de données i2b2. Dans cette étude, le temps pour annoter un document clinique était d'environ 20 secondes pour MetaMap, 4 secondes pour cTakes et de 143 millisecondes ($\alpha = 1$) à 1.6 secondes ($\alpha = 2$) pour QuickUMLS.

ClinPhen est un algorithme conçu pour détecter des phénotypes en utilisant la terminologie HPO [79]. Ses auteurs affirment que leur algorithme est 15-20 fois plus rapide que MetaMap et cTakes. D'après les auteurs, la complexité algorithmique de cTakes serait exponentielle avec la longueur du document. L'algorithme ClinPhen est similaire à l'outil **NOBLE Coder** de Tseytlin et al. [71]. Chaque mot du dictionnaire est rattaché à son terme par une table de hachage. Le document est découpé en sous-séquences par un ensemble de délimiteurs comme la virgule ou le mot "and". L'algorithme recherche un terme du dictionnaire qui contient une combinaison des mots de la sous-séquence via la table de hachage. Cet algorithme détecte aussi la négation et si le concept réfère au patient ou à un membre de sa famille. ClinPhen est capable d'annoter, en moyenne, cinq notes cliniques en 3,64 secondes.

Le webservice du National Center for Biomedial Ontology (NCBO) utilise un algorithme appelé **Mgrep** pour annoter du texte libre avec les ontologies biomédicales [80]. Le code source

de Mgrep a été publié en 2019[1]. Mgrep utilise une structure de données appelée *radix trie* pour stocker les termes d'un dictionnaire et faciliter leur recherche [81].

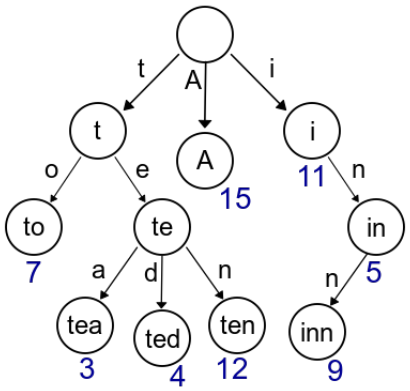


Figure 4.1: Un trie ou arbre préfixe est une structure de données utilisée pour de la correction orthographique ou de la complétion automatique.
source de l'image : `https://fr.wikipedia.org/wiki/Trie_(informatique)`

Un trie ou arbre préfixe est une structure de données utilisée pour de la correction orthographique ou de la complétion automatique (figure 4.1). Un trie est aussi utilisé pour vérifier rapidement la présence d'un terme dans un dictionnaire : en partant de la racine, les arêtes parcourues mises bout à bout forment le terme recherché. Un radix trie ou arbre PATRICIA[82] est une variante de trie qui optimise le stockage de l'arbre en mémoire : chaque nœud n'ayant qu'un seul fils est fusionné avec celui-ci. L'avantage d'un trie est sa rapidité : en stockant chaque nœud contenant un caractère dans une table de hachage, la recherche d'un mot s'effectue en temps linéaire avec la longueur de celui-ci. D'autres algorithmes utilisent cette structure de données. L'algorithme **FlashText** utilise un trie pour identifier et remplacer des termes d'un dictionnaire dans un document [83]. FlashText est lui-même inspiré de l'**algorithme Aho–Corasick**, inventé en 1975 par Alfred Vaino Aho et Margaret John Corasick, pour localiser des sous-chaînes de caractères (mots clefs) dans un document [84]. L'auteur de FlashText démontre que le temps d'annotation d'un document est indépendant de la taille du dictionnaire et donc que l'algorithme est très rapide. Le module *phrasematcher* de **spaCy**[2], une bibliothèque logicielle de TAL très connue et utilisée, développé pour identifier des entités par dictionnaire utilise la même approche. Le code de son implémentation explique que ce module est une implémentation de l'algorithme FlashText. Le principal inconvénient des algorithmes Mgrep et FlashText

[1] `https://github.com/daimh/mgrep`
[2] `https://spacy.io/`

est qu'un terme n'est détecté que par alignement parfait des chaînes de caractères [81]. Il est nécessaire d'augmenter le dictionnaire en générant des variations lexicales pour prendre en compte celles-ci.

En résumé, il existe un compromis entre rapidité et performance d'un algorithme d'annotation sémantique. Les algorithmes Mgrep et FlashText indexent efficacement un dictionnaire dans un trie et sont très rapides pour annoter un document mais ne prennent pas en compte les variations lexicales. Pour détecter des variations lexicales, les autres annotateurs sémantiques utilisent différentes stratégies de recherche dans un dictionnaire. Ces stratégies de recherche augmentent considérablement le nombre d'opérations réalisées et donc le temps de calcul. Ils sont capables d'augmenter la sensibilité mais aussi de diminuer la spécificité : d'après Stewart et al. [85] Mgrep est plus précis et plus rapide que MetaMap mais ce dernier retourne beaucoup plus de termes et est donc plus sensible.

L'algorithme IAMsystem présenté ci-dessous est une extension de l'algorithme Aho–Corasick dont Mgrep et FlashText sont des implémentations. Son originalité est d'intégrer des méthodes de comparaison de chaînes de caractères pour détecter dynamiquement des variations lexicales et de prendre en compte aisément les abréviations médicales. Nous montrons par une analyse de sa complexité algorithmique et des tests de rapidité que l'algorithme dépend peu de la taille du dictionnaire en entrées, comme Mgrep et FlashText. IAMsystem ajoute donc de la flexibilité dans la détection de termes d'un dictionnaire sans perte importante de rapidité. Ses performances en termes de rappel et de précision ont été évaluées dans trois tâches partagées.

4.3 Méthodes

IAMsystem est un algorithme d'annotation sémantique qui utilise une approche à base de dictionnaire. Il représente un dictionnaire sous forme d'arbre préfixe, appelé aussi trie. Contrairement à un trie classique (figure 4.1), chaque nœud de l'arbre n'est pas un caractère mais un token normalisé (figure 4.2).

Une première étape de normalisation (texte en minuscule, retrait des accents) et de tokenisation est réalisée sur la terminologie et sur le document. Comme FlashText, l'algorithme cherche à faire correspondre une suite de mots du document avec un chemin dans l'arbre.

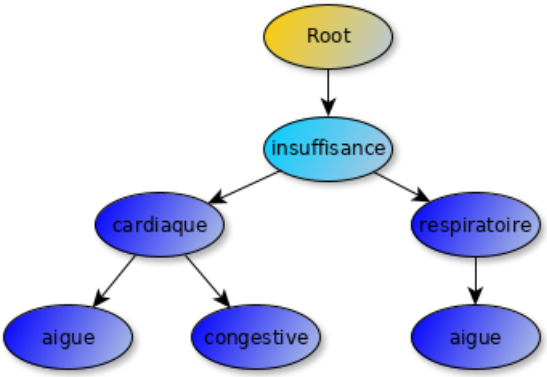


Figure 4.2: Les termes du dictionnaire sont normalisés et stockés dans une structure de données appelée trie. Chaque nœud bleu foncé correspond à un terme. Dans cet exemple, cinq termes sont stockés : "insuffisance cardiaque", "insuffisance cardiaque aigue", "insuffisance cardiaque congestive", "insuffisance respiratoire" et "insuffisance respiratoire aigue". Le token "insuffisance" est un nœud du trie mais n'est rattaché à aucun concept et n'est donc pas un terme du dictionnaire (bleu clair).

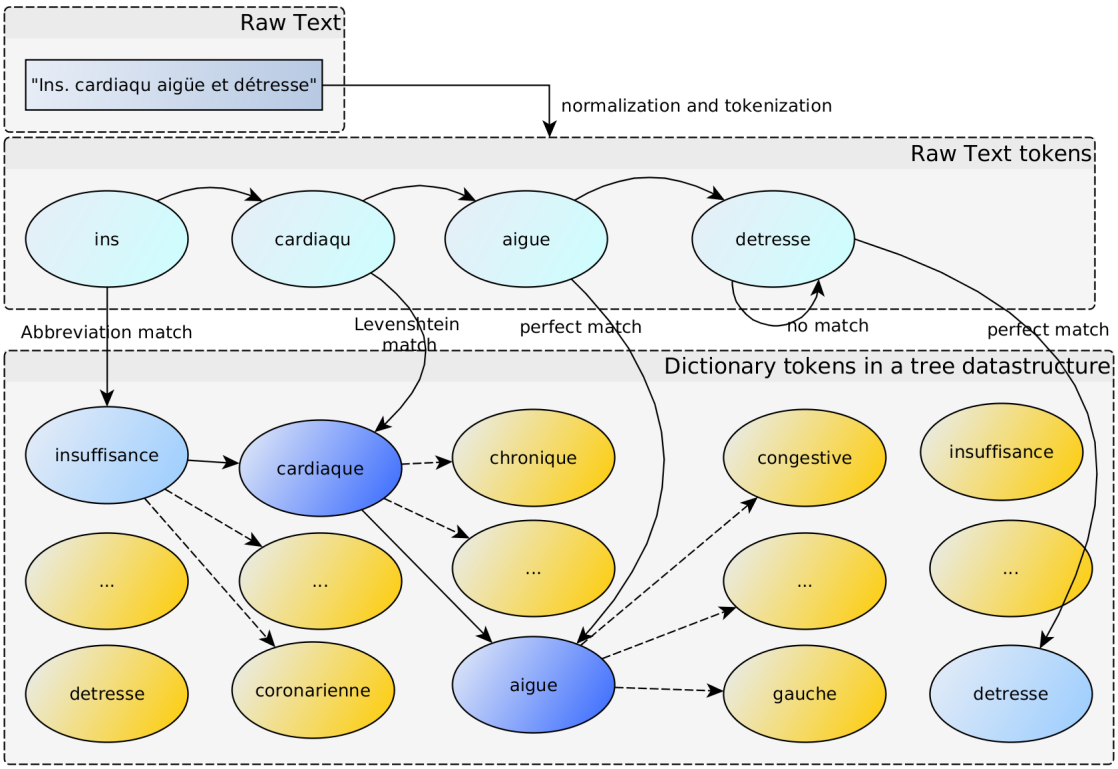


Figure 4.3: Exemple d'annotation sémantique avec IAMsystem. Le texte est normalisé et tokenisé. L'algorithme utilise des méthodes de comparaison (correspondance parfaite, distance de Levenshtein, dictionnaire d'abréviations...) pour faire correspondre un token du texte avec un token du dictionnaire. Les tokens candidats dépendent de la position actuelle dans l'arbre de la terminologie. Raw Text : texte à annoter. Flèches pleines : chemin pris par l'algorithme dans l'arbre de la terminologie. Flèches en pointillé : chemins disponibles. Noeud bleu foncé/clair : un token dans la terminologie associée à un concept (bleu foncé) ou non (bleu clair).

Dans l'exemple de la figure 4.3, le premier token du document est "ins" et l'algorithme est positionné à la racine de l'arbre. Le token "ins" est aligné au token "insuffisance" à l'aide d'un dictionnaire d'abréviations. L'algorithme modifie sa position dans l'arbre en se déplaçant sur le token "insuffisance" et passe au token suivant du document. Arrivé au token "détresse", la position dans l'arbre est le nœud "aigue" dont les parents sont "cardiaque" et "insuffisance". Aucun token suivant dans l'arbre ne correspond à ce mot : c'est une impasse. L'algorithme annote la séquence de mots (ins, cardiaqu, aigue) du document avec le terme du dictionnaire "insuffisance cardiaque aigue" et poursuit en se repositionnant à la racine de l'arbre sur le nœud *root*.

Lorsque plusieurs chemins sont possibles, l'algorithme se positionne sur plusieurs nœuds de l'arbre pour les explorer simultanément. Un exemple est donné sur la figure 4.4. Le token "meningoencephalite" peut être aligné vers le nœud "meningoencephalite" et vers le nœud "meningo" puis "encephalite".

4.3.1 Méthodes de comparaison de chaînes de caractères

Une méthode de comparaison reçoit en entrée le token du document et retourne en sortie un ou plusieurs tokens que l'algorithme utilisera pour modifier sa position dans l'arbre. Les méthodes suivantes ont été utilisées dans les tâches d'annotation :

- Méthode par correspondance parfaite.

  Elle renvoie simplement le token reçu pour rechercher un alignement parfait.

- Méthode basée sur la distance de Levenshtein

  Elle renvoie tous les tokens de la terminologie qui ont un nombre minimum (paramétré à un) d'éditions (insertions, suppressions ou substitutions) avec le token du document. Cette méthode renvoie par exemple le terme "meningo encephalite" (deux tokens) lorsque le token en entrée est "meningoencephalite" car l'insertion d'un espace sépare ces deux termes. Cette méthode permet notamment de prendre en compte des fautes d'orthographe.

- Méthodes reposant sur les abréviations ou la lemmatisation

  Ces deux méthodes utilisent un dictionnaire.

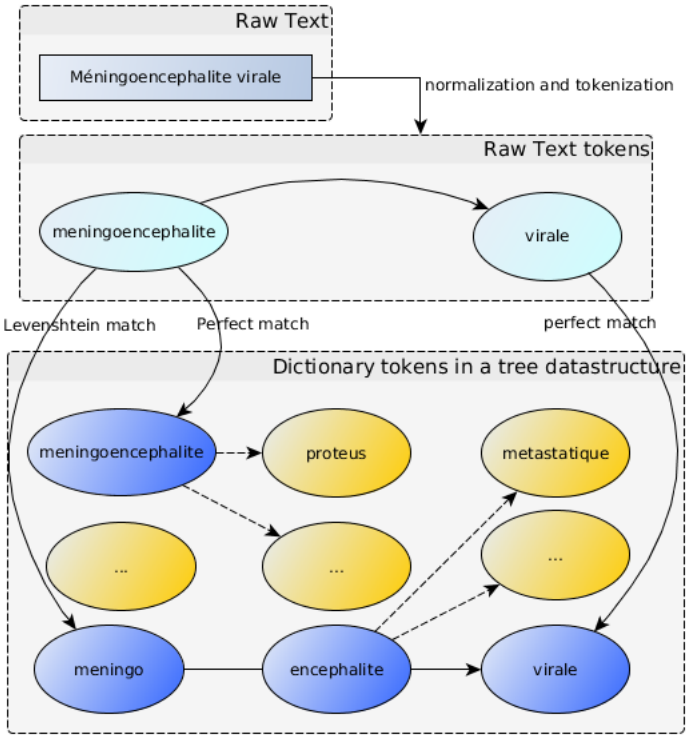


Figure 4.4: Le token "meningoencephalite" a été aligné vers le nœud "meningoencephalite" par la méthode de correspondance parfaite et vers le nœud "encephalite", situé après le nœud "meningo", par méthode de Levenshtein. En se positionnement sur plusieurs nœuds de l'arbre, l'algorithme explore dynamiquement plusieurs chemins. Après le token "virale", le premier chemin s'avère être une impasse et est donc ignoré. L'algorithme détecte le terme "meningo encephalite virale" bien que "meningoencephalite virale" n'existait pas dans la terminologie.

La lemmatisation consiste à remplacer un mot par sa forme canonique. Par exemple, le lemme petit renvoie à quatre formes fléchies : petit, petite, petits, petites.

Une abréviation consiste à remplacer la forme longue d'un terme par une forme plus courte. Par exemple, elle renvoie la séquence de tokens ("accident","vasculaire","cérébral") si le token en entrée est "avc".

- Méthode basée sur la racinisation

  La racinisation consiste à appliquer des règles pour désuffixer un mot et conserver sa racine. Par exemple, la racine de "diabetique" est "diabet". Comme la lemmatisation, cette méthode permet de prendre en compte les variations lexicales d'un mot.

Dans son utilisation, IAMsystem est paramétré par un ensemble de méthodes de comparaison. La méthode par correspondance parfaite est la méthode par défaut. Les autres méthodes sont ajoutées en fonction de la tâche d'annotation. D'autres méthodes de similarité de chaînes de caractères sont disponibles et peuvent être ajoutées par l'utilisateur.

4.3.2 Description formelle

Ci-dessous, l'algorithme d'IAMsystem est décrit comme un automate fini déterministe (AFD) [86], de la même façon qu'Aho et Corasick décrivent leur algorithme en 1975 [84]. Le diagramme de cet automate forme un trie, structure de données utilisée pour stocker et rechercher efficacement des séquences de caractères ou des séquences de mots dans notre cas. L'originalité d'IAMsystem est de recourir à des méthodes de comparaison de chaînes de caractères dans la transition d'états qui est sous la responsabilité de la fonction *next* dans un AFD. IAMsystem est donc composé d'un automate fini déterministe (AFD) et de méthodes de comparaison de chaînes de caractères.

Automate fini déterministe

Un AFD est défini par un quintuplet (Q, Σ, δ, q_0, F) [86] :

- un ensemble fini d'états Q
- un ensemble fini de symboles Σ

- une fonction de transition $\delta : Q \times \Sigma \rightarrow Q$
- un état initial $q_0 \in Q$
- un ensemble d'états finaux $F \subseteq Q$

Soit S le dictionnaire composé d'un ensemble de termes. La fonction de tokenisation transforme une séquence de caractères en une séquence de mots. En appliquant cette fonction sur chacun des termes du dictionnaire S, on obtient un dictionnaire S' composé d'un ensemble de séquences de mots.

L'ensemble des symboles $\Sigma = \{w_1^r\}$ de l'automate est l'ensemble des mots uniques du dictionnaire S' de cardinalité r. Un trie associé à un dictionnaire S' est défini récursivement [87] : $trie(S') = \{trie(S' \mid w_1), ..., trie(S' \mid w_r)\}$ où $trie(S' \mid w)$ signifie le sous-ensemble de S' (pouvant être vide) qui commence par le mot w. Un arbre enraciné, noté G, est construit à partir de cette définition ensembliste d'un trie. Soit G = (V, A) un ensemble de nœuds V et d'arêtes A. Chaque arête est composée de deux nœuds et d'un mot w : $A \subset \{(x, y, w) \in V \times V \times \Sigma, x \neq y\}$. On commence par créer la racine de G, le nœud *root*, correspondant à $trie(S')$ puis, de façon récursive, pour chaque trie x ayant pour élément un trie y commençant par le mot w $\in \Sigma$, on ajoute au graphe G un nœud x, un nœud y et une arête (x,y,w). L'ensemble fini des états de l'automate Q correspond à l'ensemble des nœuds V du graphe G. Ainsi, chaque état q $\in$ Q est un nœud de G et l'état initial q_0 est la racine de G.

La fonction de transition $\delta : Q \times \Sigma \rightarrow Q$ est définie par l'ensemble des arêtes A : $\delta = \{((q1, w), q2) \mid (q1, q2, w) \in A\}$. Une transition est possible entre deux nœuds $q1$ et $q2$ via le mot w si et seulement si il existe une arête $(q1, q2, w) \in$ A. Les états et les transitions entre états forment un arbre enraciné, c'est-à-dire un graphe acyclique orienté possédant une unique racine nommée *root*. Graphiquement, notre automate correspond à la figure 4.2 où chaque état est un nœud et chaque transition une arête entre deux nœuds.

Soit q_n un nœud de l'arbre G. Il existe un chemin unique, une séquence d'arêtes, ((q_0, q_i, w_i),(q_i, q_j, w_j),..., (q_z, q_n, w_n)) de la racine q_0 au nœud q_n. En prenant le 3^e élément de chaque triplet de cette séquence, on obtient la séquence de mots (w_i, w_j, ..., w_n) associée à l'état q_n. Soit f : Q $\rightarrow$ $(w_1, w_2, ..., w_n)$ la fonction faisant correspondre à chaque état $q_n \in Q$ une séquence de mots. Un état q_n est un état final si et seulement si la séquence de mots par la fonction f appartient à S', c'est-à-dire si cette séquence est la tokenisation d'un terme du dictionnaire.

$F \subset Q$ est l'ensemble des états finaux défini par $F = \{q \mid q \in Q, f(q) \in S'\}$. Sur la figure 4.2, les états finaux de l'automate correspondent aux nœuds bleus foncés.

Transitions d'états

On note Σ^* l'ensemble infini de séquences de mots qu'il est possible de former à partir de l'ensemble de symboles Σ (mots du dictionnaire). La fonction $next : Q \times \Sigma^* \rightarrow Q$ permet de rechercher une transition d'états à partir d'une séquence de mots. Elle reçoit en entrées un état de l'automate $q_1 \in Q$ et une séquence de mots $(w_1, w_2, ..., w_n) \in \Sigma^*$. Elle renvoie un état $q_2 \in Q, q_2 \neq q_1$ s'il existe une transition (un chemin dans l'arbre) entre q_1 et q_2 via la séquence de mots reçue. Cette fonction *next* utilise la fonction δ pour chaque transition intermédiaire entre deux états. Si aucune transition d'états n'est possible, la fonction renvoie l'ensemble vide $\varnothing \in Q$. L'originalité d'IAMsystem est que la séquence de mots $(w_1, w_2, ..., w_n) \in \Sigma^*$ reçue par la fonction *next* est produite par une méthode de comparaison de chaînes de caractères.

Méthodes de comparaison de chaînes de caractère

Pour chacun des mots du document w_{input}, un ensemble de séquences de mots est produit par chacune des méthodes de comparaison de chaînes de caractères. Soit g une méthode de comparaison, $g : w_{input} \rightarrow SEQ$ avec $SEQ \subset \Sigma^*$. La fonction g reçoit en entrée un mot w_{input}, pouvant appartenir à l'ensemble des symboles Σ ou non, et renvoie un ensemble de séquences de mots où chaque mot appartient à Σ.

Algorithme

L'automate décrit ci-dessus est utilisé par IAMsystem pour détecter un terme d'un dictionnaire. L'algorithme se positionne initialement sur l'état q_0, à la racine de l'arbre. Pour chaque mot d'un document, chacune des méthodes de comparaison de chaînes de caractères est appelée et les séquences de mots retournées sont regroupées dans un ensemble $SEQ \subset \Sigma^* = \bigcup_1^n g(w_{input})$. Pour chaque élément $seq \in SEQ$, la fonction *next* est appelée pour rechercher une transition d'état. Si une transition est possible, l'algorithme se positionne sur un nouvel état de l'automate. Si plusieurs transitions sont possibles (via plusieurs séquences de mots seq), l'algorithme se positionne sur plusieurs états afin d'explorer plusieurs chemins simultanément dans l'arbre (fig-

ure 4.4). Si aucune transition n'est trouvée, l'automate retourne à l'état initial q_0 et recherche à nouveau une transition (exemple du mot "détresse" figure 4.3). Si aucune transition n'est trouvée à l'état initial, l'algorithme passe au token suivant du document.

Lorsque l'algorithme se positionne sur un état final, ceci signifie qu'un terme du dictionnaire a été détecté et une annotation sémantique est produite. Par défaut l'algorithme renvoie le terme le plus long détecté correspondant au chemin le plus long dans l'arbre.

L'algorithme IAMsystem est décrit en pseudocode ci-dessous.

Algorithm 3 IAMsystem

Input: un document d,
un automate $\mathcal{A}$,
les fonctions next et tokenize
des méthodes de comparaison de chaînes de caractères g_i
Output: une séquence d'annotations (Annots)

```
Annots = []
doc_tokens := tokenize(d)
etats := [q_0]                                                    ▷ 1)
w_input := premier token de doc_tokens
while w_input ≠ e do
    if w_input est un mot vide then
        aller à la prochaine itération
    end if
    SEQ = ⋃_{i=1}^{n} g_i(w_input)                                 ▷ 2)
    for seq in SEQ do
        nouveaux_etats := []
        for etat in etats do
            nouvel_etat := next(etat, seq)
            ajout nouvel_etat à nouveaux_etats
        end for
    end for
    if longueur de nouveaux_etats = 0 then                        ▷ 3)
        for etat in etats do
            if etat ∈ F then                                      ▷ 4
                créer une annotation et l'ajouter à Annots
            end if
        end for
        if etats = [q_0] then                                     ▷ 3.a)
            w_input := prochain token
        else                                                      ▷ 3.b)
            etats := [q_0]
        end if
    else
        etats := nouveaux_etats                                   ▷ 4)
        w_input := prochain token
    end if
end while
retourner Annots
```

Commentaires du pseudocode :

1. L'algorithme enregistre les états dans lesquels il se trouve dans un tableau. A l'initialisation, l'algorithme est positionné sur l'état initial q_0. Ce tableau peut contenir plusieurs états correspondant à plusieurs chemins explorés dans l'arbre. Par exemple, (meningoencephalite) et (meningo,encephalite) de la figure 4.4.

2. Chaque méthode de comparaison renvoie aucune ou plusieurs séquences de mots. Pour chaque séquence et chaque état (boucle *for* imbriquée), la fonction next est appelée. Si un chemin existe dans l'arbre alors la fonction *next* renvoie un nouvel état, si aucun chemin n'existe elle renvoie ∅.

3. Si aucune transition d'état n'est possible pour le mot en cours w_{input} l'algorithme sauvegarde le ou le(s) termes éventuellement détectés. Puis:

   (a) Si l'algorithme est dans l'état initial, il passe au token du document suivant.

   (b) Si l'algorithme n'est pas dans l'état initial, il revient à l'état initial q_0 pour rechercher une transition avec ce token.

4. Si une transition est trouvée, l'algorithme change d'état(s) et passe au token suivant.

4.3.3 Analyse de la complexité algorithmique

Dans cette sous-section, nous nous intéressons à la complexité algorithmique d'IAMsystem, c'est-à-dire l'analyse du nombre d'opérations réalisées pour annoter un document et la manière dont cette complexité évolue avec la taille du document et de la terminologie.

La notation Big-O de Bachmann–Landau est la méthode la plus utilisée pour exprimer la complexité d'un algorithme [88, 65]. Elle permet de comparer différents algorithmes de façon indépendante aux caractéristiques matérielles d'une machine. Elle représente la borne supérieure asymptotique de la complexité. Pour une fonction g(n), l'expression O(g(n)) représente un ensemble de fonctions:
O(g(n)) = {f(n): ∃ c, n0 ∈ Z^+ tel que 0 ≤ f(n) ≤ cg(n) pour tout n ≥ n0 }.
Une fonction f(n) appartient à l'ensemble O(g(n)) s'il existe une constante positive c tel que f(n) ≤ cg(n) pour un nombre n suffisamment grand. On note f(n) ∈ O(g(n)) ou f(n) = O(g(n)).

IAMsystem est une fonction qui prend en entrée un document et une terminologie puis produit en sortie un ensemble d'annotations. Soit n le nombre de mots dans un document et m le nombre de termes dans la terminologie. On cherche à comprendre la complexité d'IAMsystem, notée O(IAMsystem), quand n et m augmentent. Tout algorithme d'annotation sémantique doit parcourir l'ensemble des mots d'un document, la complexité est donc au moins O(n). Si pour

chaque mot du document, l'algorithme extrait des séquences de mots dans une fenêtre w autour de ce mot puis les compare naïvement avec les m termes de la terminologie, la complexité est O(nm) ce qui n'est pas scalable pour annoter les documents d'un EDS avec une grande terminologie.

Nombre d'opérations réalisées

Les étapes de tokenisation et de normalisation d'un document sont réalisées en temps linéaire, elles sont indépendantes de la taille de la terminologie.
Pour chaque token d'un document, chacune des méthodes de comparaison est appelée. Comme chacune des méthodes est indépendante, celles-ci peuvent être appelées de façon parallèle plutôt que séquentiellement. La complexité de l'ensemble de ces méthodes dépend donc de la plus complexe d'entre elles, notée g_{max}, qui peut agir comme un goulot d'étranglement.
La fonction *next* est ensuite appelée autant de fois qu'il y a d'états sur lesquels est positionné l'algorithme et de séquences de mots retournées par les méthodes de comparaison.

La complexité algorithmique d'IAMsystem est donc dominée par deux étapes : les méthodes de comparaison de chaînes de caractères et la transition d'états:
$O(IAMsystem) \approx n \times (O(g_{max}) + O(next) \times E[|SEQ|] \times E[n_{etats}])$
où $O(g_{max})$ est la complexité de la plus complexe des fonctions g, O(next) la complexité de la fonction next, $E[|SEQ|]$ la moyenne du nombre de séquences de mots produites par les méthodes de comparaisons et $E[n_{etats}]$ la moyenne du nombre d'états sur lesquels se trouve l'algorithme. Chacune de ces étapes est analysée séparément ci-dessous.

O(g_{max}) La complexité des principales méthodes de comparaison est décrite ci-dessous:

- Méthode par correspondance parfaite

  Elle renvoie le token reçu donc une seule opération est réalisée.

- Méthode par abréviations ou lemmatisation

  Les abréviations / lemmes sont fournis dans un dictionnaire, ils ne sont pas détectés à la volée. Tous les couples (abréviation, forme longue) et (lemme, forme fléchie) sont stockés dans une table de hachage. Rechercher un mot dans une table de hachage est réalisé en

temps constant O(1) si la fonction de hachage est parfaite. Des collisions peuvent se produire en utilisant une mauvaise fonction de hachage. Dans son implémentation en Java 1.8, les clefs des tables de hachage utilisées par IAMsystem sont des chaînes de caractères, la fonction de hachage utilisée est celle de la classe String qui appartient au noyau de Java. La table de hachage est implémentée avec les paramètres par défaut de la classe java.util.HashMap. Les abréviations et les lemmes d'un mot sont trouvés en temps constant ≈ O(1)

- Méthode de Levenshtein

  IAMsystem utilise l'implémentation de Apache Lucene[3]. Étant donné un dictionnaire de mots w_1,..., w_d, la distance de Levenshtein entre un mot v et chaque mot w_i est calculé en temps O(|v|) où |v| est la longueur du mot [89]. La complexité de cette méthode est indépendante de la taille de la terminologie.

  Comme cet algorithme est déterministe, une table de hachage est utilisée pour mettre en cache les résultats et diminuer le temps de calcul de O(|v|) à ≈ O(1). Le gain de temps est d'autant plus important qu'un mot apparaît fréquemment dans le corpus. Soit n' le nombre de tokens unique d'un document contenant n tokens. Plus un document est long, plus n' est petit par rapport à n. A l'échelle d'un corpus de documents, n' ≪ n.

  La complexité est $\sum_{i=1}^{n'}(O|v_i|)$ pour chaque token unique d'un document et O(1) pour (n - n') tokens avec le système de cache.

- Alignement par racinisation

  Chaque token de la terminologie est racinisé et le résultat est stocké dans une table de hachage contenant la racine et son token. Chaque token du document est racinisé et si cette racine est présente dans la table de hachage, le ou les tokens de la terminologie sont renvoyés. La racinisation d'un token du document est réalisée par un ensemble de règles en temps linéaire. Comme pour la méthode de Levenshtein, un système de cache sauvegarde en mémoire les résultats. Cette méthode fonctionne en temps linéaire et est indépendante de la taille de la terminologie.

[3] `https://lucene.apache.org/`

En résumé, la complexité de chacune des méthodes de comparaison est indépendante de la taille de la terminologie. O(g_{max}) est bornée par une constante.

O(next) × E[|SEQ|] × E[n$_{etats}$] Les transitions entre états sont stockées dans une table de hachage. Rechercher un mot dans une table de hachage est de l'ordre de O(1). Etant donné un état q et un mot w, O(next) ≈ O(1) pour trouver une transition d'état.

$E[|SEQ|]$ est difficile à estimer. Le pire des scénarios est que chaque mot du document soit similaire, via une méthode de comparaison, à tous les mots du dictionnaire. Le meilleur des scénarios est que chaque mot du document ne soit similaire à aucun mot du dictionnaire mais les méthodes de comparaison ne seraient alors d'aucune utilité. Il est intéressant de noter qu'ajouter des termes au dictionnaire qui n'ont aucune similarité avec les mots d'un document n'augmentera pas la complexité car le sous-ensemble de l'arbre contenant ces termes ne sera jamais exploré par l'algorithme. Intuitivement $E[|SEQ|]$ dépend du nombre de mots dans la terminologie et de leur similarité avec les mots d'un document : $E[|SEQ|]$ dépend donc de la tâche d'annotation. Dans la sous-section suivante "tests de performance", $E[|SEQ|]$ est environ égale à 3 pour des documents médicaux annotés avec une terminologie contenant plus de 300 000 termes médicaux et environ 1 pour une terminologie relativement petite. Dans ce scénario réel, les méthodes de comparaison renvoient, en moyenne, un petit nombre de séquences de mots.

Dans le pire des scénarios, $E[n_{etats}]$ est égal au nombre d'états de l'automate qui est supérieur ou égal à m. En pratique $E[n_{etats}] \approx 1$ car il est rare que l'algorithme explore plusieurs chemins dans l'arbre.

En résumé, la complexité de $O(next) \times E[|SEQ|] \times E[n_{etats}]$ dépend de la tâche d'annotation. Dans le pire des scénarios, cette complexité est de l'ordre de O(m^2) : l'algorithme est positionné sur tous les états de l'automate et doit explorer autant de séquences de mots qu'il y a de termes dans le dictionnaire.

En pratique, $O(next) \times E[|SEQ|] \times E[n_{etats}] \approx 5$ pour une grande terminologie (m = 300 000) et plusieurs méthodes de comparaison, c'est-à-dire que l'algorithme est positionné en moyenne sur un seul état de l'automate et explore quelques séquences de mots. Pour annoter des documents médicaux avec une terminologie médicale, cette complexité appartient à O(log(m)).

O(IAMsystem)

En résumé, dans le pire des scénarios la complexité d'IAMsystem est O(nm^2) et dans le meilleur des scénarios O(n). En pratique $O(IAMsystem) \in O(nlog(m))$: le nombre d'opérations dépend peu du nombre m de termes d'un dictionnaire. En comparaison, la complexité des algorithmes Mgrep et FlashText est O(n) : le nombre d'opérations ne dépend pas de la taille du dictionnaire. IAMsystem ajoute donc de la flexibilité pour détecter un terme au prix de diminuer très légèrement la rapidité d'exécution.

4.3.4 Tests de performance

Des tests de performance ont été conduits pour évaluer la rapidité d'IAMsystem pour détecter des termes dans un corpus de documents sur un ordinateur portable Intel Core i7-5700HQ @2.70GH x 8CPUs et 8 Go de RAM. Un corpus de documents a été créé avec 2279 articles Wikipedia en français identifiés par un code CIM-10 dans Wikidata. Ce corpus était composé de $2\,832\,502$ mots et $84\,420$ mots uniques.

Trois terminologies de taille différente ont été créées pour évaluer le temps d'annotation sémantique en fonction de la taille de la terminologie. Le tableau 4.1 fournit des informations sur chaque terminologie. La terminologie "UMLS" correspond à l'ensemble des terminologies en langue française extraites de l'UMLS [74]. La terminologie "Diseases" correspond à un sous-ensemble de la terminologie "UMLS" où l'identifiant du concept appartient à la catégorie "Disease or Syndrome". La terminologie "Romedi" correspond aux ingrédients et noms commerciaux de Romedi. Romedi est relativement petite, elle est composée de 8018 termes différents, la terminologie "Diseases" est environ 5,6 fois plus grande et l'UMLS contient environ 41 fois plus de termes.

	UMLS	Diseases	Romedi
Nombre de termes distincts	331 592	45 346	8018
Ordre de grandeur	x41	x5.6	1
Nombre de concepts distincts	178 591	18 317	7607
Nombre de nœuds/états	673 064	99 228	11 512

Table 4.1: Trois terminologies de taille différente "UMLS", "Diseases" et "Romedi" ont été générées pour comparer les différences de temps de détection.

Les temps de détection sont fournis dans le tableau 4.2. Plus il y a de méthodes de comparaison, plus le temps de détection augmente : de 3,4 secondes de détection en utilisant uniquement la méthode de correspondance parfaite à 42 secondes en utilisant la méthode de Levenshtein avec la méthode de racinisation.

	Parfait	Levenshtein	Racinisation	Racinisation et Levenshtein
Nombre de concepts détectés :				
UMLS	597 317	1 221 539	1 068 675	1 536 288
Diseases	53 726	91 692	84 792	104 242
Romedi	12 091	27 798	15 766	30 166
Temps en secondes :				
UMLS	3.4	40.3*	5.2	41.9
Diseases	2.8	22.3	4.3	25
Romedi	2.8	19.3	3.8	20.1
$E[\|SEQ\|]$				
UMLS	1	2.43	1.99	3.06
Diseases	1	1.58	1.45	1.82
Romedi	1	1.05	1.03	1.05

Table 4.2: Nombre de concepts détectés, temps d'annotation et nombre moyens de séquences de mots renvoyées ($E[|SEQ|]$) par les méthodes de comparaison en fonction de trois terminologies de taille différente et trois paramétrages différents des méthodes de comparaison. Le corpus était composé de 2279 articles médicaux en français extraits de Wikipedia contenant un total de 2 832 502 mots.
*: 5 secondes lors du 2ème passage.

La mémoire utilisée par le programme Java était d'environ 700 megaoctets pour la terminologie UMLS, 300 megaoctets pour "Diseases" et 200 megaoctets pour "Romedi". La mémoire utilisée dépendait peu de l'utilisation de méthodes de comparaison.

Lorsque la taille de la terminologie augmente, le temps d'annotation augmente. Avec une terminologie 41x plus grande, il faut 2,1 fois (40,3 versus 19,3 secondes) plus le temps pour détecter les concepts avec la méthode de Levenshtein et 1,36 fois (5,2 versus 3,8 secondes) plus de temps pour la méthode de racinisation. La méthode utilisant la distance de Levenshtein est beaucoup plus longue car elle appelle une librairie externe. Cependant le système de cache permet un gain de temps significatif : s'il faut 40.3 secondes pour annoter les documents avec l'UMLS avec la méthode de Levenshtein lors du premier passage, il faut seulement 5 secondes lors du 2ème passage car les séquences de mots sont placées en cache et la librairie externe n'est plus appelée. L'algorithme est lent sur les premiers documents puis de plus en plus rapide car il sollicite de moins en moins les méthodes de comparaison et de plus en plus le système de

cache.

Pour la terminologie UMLS, le nombre de séquences de mots renvoyées par les méthodes de comparaison va de 1 (alignement parfait) à 3 (Levenshtein et racinisation). Ce nombre augmente très peu même pour une grande terminologie comme l'UMLS. Ceci peut s'expliquer par la dissimilarité entre les mots qui augmente peu même lorsque la terminologie augmente. La méthode de Levenshtein était configurée pour rechercher une différence d'un seul caractère, avec une différence de deux caractères $E[|SEQ|]$ augmenterait.

En résumé, le temps d'annotation n'est pas proportionnel à la taille de la terminologie et est plus lent pour les premiers documents que pour les derniers documents d'un corpus.

4.3.5 Complexité en termes de mémoire

La complexité de la mémoire est dominée par le stockage de l'automate (le trie) et le système de cache.

Le nombre d'états de l'automate est fixé par la taille de la terminologie. Soit m la taille de la terminologie, le nombre d'états est égal à environ 2m d'après les tests de performance (tableau 4.1).

Le système de cache enregistre, pour chaque mot unique d'un corpus, les mots similaires de la terminologie. Soit n' le nombre de mots uniques d'un corpus, la complexité est donc O($E[|SEQ|]\times n'$) avec $E[|SEQ|]$ la moyenne du nombre de séquences de mots produites par les méthodes de comparaisons. La complexité de la mémoire est donc $O(E[|SEQ|]\times n' + 2m)$.

Dans les tests de performance, le nombre d'états de la terminologie UMLS était de $673\,064$ ce qui était plus élevé que le nombre de mots uniques ($84\,420$). La majorité de la mémoire utilisée était dédiée au stockage de l'automate. Lorsqu'il s'agit d'indexer un très grand corpus, n' peut potentiellement devenir très grand. Pour fixer une limite à la taille de la mémoire utilisée, le système de cache d'IAMsystem fixe le nombre de mots en cache à 100 000. Au-delà, le système de cache n'accepte plus de mots : les méthodes de comparaison sont appelées plusieurs fois si le mot apparaît plusieurs fois dans le document. D'après la loi de Zipf, la fréquence d'un mot est inversement proportionnelle à son rang dans la table de fréquence. Un mot qui ne serait pas dans le système de cache serait donc un mot très rare dans le corpus et

posséderait donc une faible probabilité d'apparaître plusieurs fois. Avec ce système de cache, le terme $E[|SEQ|] \times n'$ possède une borne supérieure.

La complexité de la mémoire augmente linéairement avec la taille de la terminologie. La complexité de la mémoire d'IAMsystem est donc O(m), c'est-à-dire qu'elle augmente proportionnellement à m et peut devenir très grande. Pour une grande terminologie comme l'UMLS, la mémoire utilisée est inférieure à 1 gigaoctect, ce qui est relativement peu à l'heure actuelle.

4.3.6 Evaluation

Les performances d'IAMsystem ont été évaluées dans le cadre de trois tâches partagées dont deux organisées par l'initiative CLEF (Conference and Labs of the Evaluation Forum). Cette initiative a pour mission principale de promouvoir la recherche, l'innovation et le développement de systèmes d'accès à l'information en mettant l'accent sur l'information multilingue et multimodale[4]. L'objectif d'une tâche partagée est de comparer les performances de plusieurs algorithmes appliqués à un même problème et d'améliorer les connaissances sur l'état de l'art. La performance des systèmes était évaluée par les métriques habituelles d'extraction d'information : précision, rappel et F-mesure (plus précisément, β_1=1 était utilisé).

$$Rappel = \frac{\text{Vrais positifs}}{\text{Vrais positifs + Faux négatifs}}$$

$$Precision = \frac{\text{Vrais positifs}}{\text{Vrais positifs + Faux positifs}}$$

$$Fmesure = \frac{(1 + \beta^2) \times precision \times rappel}{\beta^2 \times precision + rappel}$$

[4] http://www.clef-initiative.eu/

4.4 Résultats

4.4.1 Annotation des certificats de décès français

Cette sous-section présente la tâche 1, extraction d'informations multilingues, du défi CLEF eHealth 2018 [90]. Cette tâche consistait à coder automatiquement les certificats de décès à l'aide de la Classification Internationale des Maladies, 10^e révision (CIM-10).

Corpus

Le jeu de données s'appelle le CépiDC corpus. Plusieurs fichiers CSV (comma-separated values) ont été fournis par les organisateurs contenant des certificats de décès annotés entre 2006 et 2014. Le jeu de données d'entraînement contenait $125\,383$ certificats. Chaque certificat contenait une à plusieurs lignes de texte (cause médicale ayant conduit au décès), ainsi que des métadonnées (âge, sexe, lieu de décès). Chaque fichier CSV contenait une colonne "Raw Text" entrée par un médecin, une colonne "Standard Text" entrée par un codeur humain qui doit justifier la sélection d'un code CIM-10 dans la dernière colonne. Le tableau 4.3 présente un extrait de ces fichiers. Zero à plusieurs codes CIM-10 pouvaient être attribués à chaque ligne d'un certificat de décès.

Raw Text	Standard Text	ICD-10 code
SYNDROME DE GLISEMENT AVEC GRABATISATION DEPUIS OCTOBRE 2012	syndrome glissement	R453
SYNDROME DE GLISEMENT AVEC GRABATISATION DEPUIS OCTOBRE 2012	grabatisation 2 mois	R263

Table 4.3: Un extrait du fichier CSV fourni par les organisateurs de la tâche 1, extraction d'informations multilingues, du défi CLEF eHealth 2018. Raw Text: texte entré par un médecin dupliqué dans le fichier car plusieurs codes lui sont assignés.
Standard Text: texte entré par un codeur humain pour justifier la sélection d'un code CIM-10.

Dictionnaires

Chaque équipe pouvait réaliser jusqu'à cinq soumissions, appelées run. Deux dictionnaires ont été construits pour annoter les certificats de décès avec IAMsystem. Le premier dictionnaire a été créé en sélectionnant tous les termes de la colonne "Standard Text" (2ème run). Le 2ème

dictionnaire a été créé en ajoutant au premier dictionnaire l'ensemble des termes de la CIM-10 fournie par les organisateurs. Quand un terme était associé à plusieurs codes CIM-10, le code le plus fréquent était gardé (tableau 4.4).

Le premier dictionnaire contenait 42 439 termes et 3539 codes CIM-10 et le deuxième 148 448 termes et 6392 codes CIM-10. Les métadonnées n'ont pas été utilisées.

Les certificats de décès contenaient de nombreuses fautes d'orthographe et des abréviations. IAMsystem a été configuré avec une méthode de comparaison de chaînes de caractères basée sur la distance de Levenshtein, paramétrée à 1, et avec un dictionnaire d'abréviations créé à partir des données fournies par les organisateurs. Un exemple d'annotation d'un certificat de décès avec IAMsystem est présenté figure 4.3.

Standard Text	ICD-10 code	number of occurrences
avc	F179	1
avc	I64	260
avc	I640	1,635
avc	T821	1
avc	Z915	1
avc	I489	1

Table 4.4: Certains termes comme "avc" étaient associés à plusieurs codes CIM-10. Nous avons gardé le plus fréquent, I640 dans cet exemple. Standard Text: text entré par un code humain.

System	Precision	Rappel	F-mesure
run1	0.782	0,772	0.777
run2	0.794	0.779	0.786
average score	0.712	0.581	0.634
median score	0.771	0.545	0.641
best score	0.841	0.835	0.838

Table 4.5: Performance de IAMsystem sur le jeu de données du CépiDC.

Le tableau 4.5 montre les performances du système avec la médiane et la moyenne des autres participants. Douze équipes ont participé à cette tâche. Les meilleurs performances ont été obtenues par l'équipe catalanne Ixamed [91] qui a obtenu une F-mesure de 0,838. Cette équipe a utilisé une méthode de classification par réseau de neurones Seq2Seq. IAMsystem se classe deuxième avec une F-mesure de 0,786.

4.4.2 Détection de maladies et d'actes dans des notes cliniques

Cette sous-section présente la tâche 1, extraction d'informations multilingues, du défi CLEF eHealth 2020 [92]. La tâche consistait à attribuer automatiquement les codes de diagnostic et d'acte de la CIM-10 à des notes cliniques issues de dossiers médicaux électroniques espagnols.

Corpus

Le jeu de données s'appelle le CodiEsp corpus. Ce corpus comprenait 1 000 cas cliniques espagnols sélectionnés et annotés par un médecin. Chaque cas clinique était un fichier texte brut avec le nom du fichier comme identifiant. Le jeu d'entraînement était composé de 750 cas cliniques. Les participants devaient annoter automatiquement 3 001 cas cliniques parmi lesquels 250 formaient le jeu de test et celui-ci n'était pas connu des participants.

Les versions espagnoles des terminologies CIM10-CM (International Classification of Diseases, Tenth Revision, Clinical Modification) et CIM10-PCS (International Classification of Diseases, Tenth Revision, Procedure Coding System) ont été fournies par les organisateurs. Ces terminologies contenaient respectivement 98 288 et 75 789 codes différents. Seulement 2 921 codes distincts (1,7%) étaient présents dans le jeu de données d'entraînement. Aucun exemple n'était disponible pour une grande majorité de codes tandis que certains codes étaient fréquents.

La tâche était composée de trois sous-tâches:

1. Annotation des diagnostics : les systèmes devaient détecter les symptômes et les maladies mentionnés dans les notes cliniques en prédisant les codes diagnostics CIM10-CM.

2. Annotation des actes : les systèmes devaient détecter les actes mentionnés dans les notes cliniques en prédisant les codes ICD10-PCS.

3. Explicabilité : les systèmes devaient fournir des annotations textuelles, c'est-à-dire la séquence de mots pour chaque annotation avec un code ICD10-CM et ICD10-PCS des sous-tâches 1 et 2.

Le format d'annotation était un fichier séparé par des tabulations avec deux champs pour les sous-tâches n°1 et n°2 correspondant à l'identifiant du cas clinique et un code ICD10-CM

ou ICD10-PCS. Trois autres champs étaient attendus pour la sous-tâche n°3 : la position du premier caractère de la séquence de mots, la position du dernier caractère (sépararation par ";" si plusieurs début/fin d'un terme discontinu) et si le code était un diagnostic ou un acte.

Un script a été développé pour transformer les fichiers fournis par les organisteurs en fichiers visualisables dans Brat [68]. Un exemple de note clinique est fourni figure 4.5.

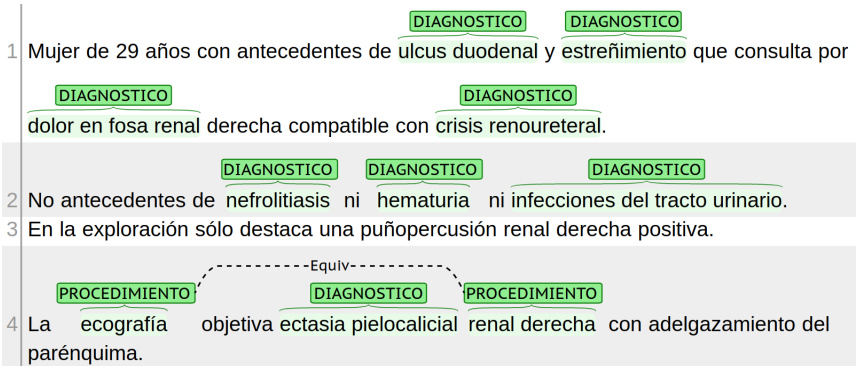


Figure 4.5: L'interface Brat a été utilisée pour visualiser les annotations faites par l'expert médical. Chaque annotation était liée à un code ICD10-CM ou ICD10-PCS (non représenté ici).

Tous les diagnostics et actes dans une note clinique devaient être codés. Le système n'avait pas besoin de détecter la négation qui était souvent présente. La mention de l'absence d'un diagnostic ou d'un acte était également codée. Le nombre d'annotations constituées de séquences de mots discontinues était de 14,6% et 39,7% pour les diagnostics et les actes, respectivement.

Deux dictionnaires ont été créés et deux soumissions ont été réalisées. Le premier dictionnaire (run1) ne contenait que les termes des annotations de l'expert médical, soit 6 316 termes. Le second dictionnaire (run2) était la combinaison du premier dictionnaire et des termes normalisés de la terminologie CIM-10. Il contenait un total de 94 386 termes.

Résultats

22 participants (78 runs) ont participé à la première sous-tâche, 17 participants (64 runs) à la deuxième et seulement 8 participants à la troisième. De multiples approches ont été employées par les différents participants [92]. Les organisateurs les ont classifiées dans trois catégories : les systèmes à base de dictionnaire, à base de machine learning et de modèle de langage. IAMsystem a obtenu la meilleure F-mesure aux trois sous-tâches, la meilleure MAP (mean average

Sous-tâche	Run	MAP	Precision	Rappel	F-mesure
1	1	0.52	0.82	0.59	0.69
	2	0.51	0.79	0.59	0.68
2	1	0.43	0.66	0.37	0.48
	2	0.49	0.69	0.42	0.52
3	1	-	0.005	0.003	0.005
	2	-	0.006	0.004	0.005
3 (non official)*	1	-	0.75	0.52	0.61
	2	-	0.73	0.52	0.61

Table 4.6: Performance d'IAMsystem sur le jeu de données du Codiesp. MAP: Mean Average Precision.
*: A la soumission officielle, la position du dernier caractère d'un terme a été extrait mais la position du dernier caractère + 1 était attendue. Les résultats non officiels correspondent à la correction de cette erreur de format. Les organisateurs ont publié les résultats 'non officiels' d'IAMsystem à la sous-tâche 3 mais ils n'ont pas été pris en compte dans le calcul des prix.

precision) aux sous-tâches 2 et 3 et la 2ème meilleure MAP à la sous-tache 1. Les deuxièmes et troisièmes meilleurs systèmes ont utilisé un modèle de langage de type BERT [93, 94]. Les organisateurs ont montré que combiner les résultats d'IAMsystem et du troisième meilleur système (Fujitsu [94]), par simple union, permettait d'améliorer la F-mesure de la sous-tâche 1 (de 0.687 à 0.703).

4.4.3 Détection des espèces dans les documents médicaux

Cette sous-section présente la tâche 1, LivingNER-Species NER, de la tâche partagée LivingNER [95]. La tâche consistait à détecter automatiquement la mention d'espèces dans des notes cliniques en espagnol. Les organisateurs de cette tâche étaient les mêmes que celle présentée ci-dessus mais cette tâche ne faisait pas partie de l'initiative CLEF.

Corpus

Le jeu de données s'appelle le LivingNER corpus. Le corpus a été annoté manuellement par des experts médicaux suivant un guide d'annotation spécifiquement créé pour cette tâche. Le corpus était très varié puisqu'il contenait des annotations d'animaux, de plantes et de micro-organismes et les comptes-rendus cliniques provenaient de 20 disciplines médicales (cardiologie, oncologie, radiologie, etc.). Le gold standard consistait en une collection de 2000 comptes-rendus en espagnol, échantillonné de manière aléatoire en trois sous-ensembles : développement, validation

filename	mark	label	off0	off1	span
caso_clinico_radiologia526	T1	SPECIES	18	21	VIH
caso_clinico_radiologia526	T2	HUMAN	0	5	Varón

Table 4.7: Extrait d'un fichier d'annotation fourni par les organisateurs. La colonne filename contient l'identifiant d'un document ; mark est un identifiant d'annotation ; label prend deux valeurs : SPECIES ou HUMAN ; off0 et off1 sont les positions de début et de fin de l'entité dans le document, respectivement ; span correspond au texte annoté.

et test, composés respectivement de 1000, 500 et 500 documents. L'ensemble de test a été publié sans annotation avec une grande collection de comptes-rendus supplémentaires pour éviter des annotations manuelles. Les participants devaient annoter un total de 13 467 documents, dont seulement 500 faisaient partie de l'ensemble de test et étaient connus des organisateurs.

Chaque compte-rendu était fourni dans un fichier unique en UTF8. Par exemple, le fichier "caso_clinico_radiologia526.txt" commence par la phrase suivante : *Varón de 49 años, VIH+, al que se solicita TC abdominal urgente por sospecha de obstrucción intestinal.* Les mots "Varón" (homme) et "VIH" (HIV) désignent respectivement une personne humaine et un virus. Ces termes ont été annotés manuellement par des experts médicaux et inclus dans les fichiers d'annotation publiés par les organisateurs. Les fichiers d'annotation étaient des fichiers TSV (tab-separated values) contenant les positions de début et de fin des entités ainsi que le texte annoté. Un extrait est fourni dans le tableau 4.7.

L'objectif des participants était d'annoter automatiquement les documents de l'ensemble de test dans un format similaire à celui des ensembles de validation et de développement (table 4.7).

Les ensembles de développement et de validation ont été combinés en un seul ensemble, appelé par la suite "ensemble d'entraînement". Cet ensemble d'entraînement contenait 1 500 cas cliniques pour lesquels 23 203 annotations étaient disponibles. Ces annotations contenaient 3 418 valeurs uniques dans la colonne span après normalisation. Le tableau 4.8 affiche les 10 annotations les plus fréquentes avec leur fréquence et le pourcentage cumulé. Le tableau montre qu'environ 21% des termes annotés dans les documents étaient le mot "paciente".

Un dictionnaire temporaire a été créé en sélectionnant toutes les valeurs de la colonne "Span" des fichiers d'annotation et a été utilisé par IAMsystem pour annoter l'ensemble d'entraînement. Le rappel et la précision sur l'ensemble d'entraînement avec ce dictionnaire temporaire étaient de 0,97 et 0,53 ; respectivement. Le fichier de sortie ne correspondait pas

span	fréquence	pourcentage cumulé
paciente	4861	0.209
vih	576	0.234
varón	521	0.257
personales	424	0.275
mujer	397	0.292
pacientes	292	0.305
familiares	270	0.316
madre	259	0.328
cmv	197	0.336
sars-cov-2	197	0.345

Table 4.8: Les 10 termes les plus fréquents avec leur fréquence et le pourcentage cumulé dans les fichiers d'annotation.

parfaitement aux fichiers d'annotation pour deux raisons. Tout d'abord, IAMsystem ne sélectionne que le terme le plus long détecté, mais un expert humain ne fait pas nécessairement le même choix. Par exemple, si le document contient "VIH 1" et que le dictionnaire contient "VIH" et "VIH 1", l'algorithme retourne "VIH 1" bien que l'humain puisse annoter "VIH". Deuxièmement, si un humain annote un terme dans un document, il peut ne pas être annoté dans d'autres documents, alors que l'algorithme l'annotera dans tous les documents. Afin de maximiser le score F1 sur l'ensemble d'apprentissage, certains termes ont dû être supprimés du dictionnaire temporaire. Ce faisant, le rappel a diminué mais la précision et le F1-score a augmenté sur l'ensemble d'apprentissage. Pour identifier les termes à supprimer, le fichier de sortie a été comparé aux fichiers d'annotation. La fréquence de chaque terme dans les fichiers d'annotation et dans le fichier de sortie a été comparée : si le rapport entre les deux fréquences était supérieur à 2, le terme était retiré du dictionnaire. Par exemple, le terme "covid-19" est apparu 231 fois dans l'ensemble d'apprentissage, ce qui a entraîné 231 annotations par IAMsystem dans le fichier de sortie, mais il n'a été annoté qu'une seule fois par les experts humains. La suppression de ce terme a conduit à la suppression d'un vrai positif et de 230 faux positifs. Au total, 109 termes ont été retirés. Le dictionnaire final contenant 3 683 termes et a été utilisé pour annoter l'ensemble de test. Le rappel et la précision sur l'ensemble d'entraînement avec ce dictionnaire personnalisé étaient respectivement de 0,96 et 0,97. IAMsystem a été configuré sans méthode de correspondance approximative des chaînes de caractères, de sorte que seule la méthode de correspondance exacte a été appliquée.

Système	Micro-Precision	Micro-Rappel	Micro-F1 score
Meilleur système	0.9583	0.9438	0.951
IAMsystem	0.9209	0.8733	0.8965
Moyenne	0.876	0.807	0.824

Table 4.9: Performances du meilleur système, d'IAMsystem et moyenne des participants sur l'ensemble test de la tâche 1, LivingNER-Species NER, de la tâche partagée LivingNER [95]

Résultats

Il a fallu environ 6 secondes pour annoter les 13 467 documents de l'ensemble test et générer 107 651 annotations sur un ordinateur portable Intel Core i7-5700HQ @2.70GH x 8CPUs. Avec un F1-score de 0.8965, IAMsystem se classe onzième sur vingt participants [95]. Le tableau 4.9 montre les résultats du meilleur système, d'IAMsystem et la moyenne des participants. Les meilleures approches ont utilisé un réglage fin et non standard d'un modèle de langage de type transformer [95]. Le meilleur F1-score est obtenu par la société Vicomtech qui a utilisé un modèle BERT pré-entraîné sur un large corpus de textes espagnols et configuré avec une méthode personnalisée de fenêtre glissante pour prendre en compte le contexte d'un paragraphe [96].

4.5 Discussion

IAMsystem est un nouvel algorithme d'annotation sémantique généraliste à base de dictionnaire. L'algorithme ne s'intéresse qu'à trouver une transition d'états en recherchant une similarité de chaînes de caractères entre un mot du document et un mot de la terminologie. Il ne génère pas de variations lexicales comme MetaMap ou des nouvelles séquences de mots comme QuickUMLS. Il ne calcule pas de mesure de similarité entre plusieurs séquences de mots comme QuickUMLS, le calcul d'une similarité n'est réalisé qu'au niveau d'un seul mot. Il ne réalise pas d'étiquetage morphosyntaxique préalable comme cTakes ou MetaMap qui est une étape souvent chronophage dans une pipepline. Par sa simplicité, IAMsystem est l'un des annotateurs les plus rapides. Par rapport aux algorithmes Mgrep [80] et FlashText [83] qui utilisent une structure de données similaire pour la rapidité, il ajoute de la flexibilité en prenant en compte les abréviations, les variations lexicales et les fautes d'orthographe d'un ou plusieurs mots. La détection dynamique d'une variation d'un mot lui permet de prendre en compte un

nombre très élevé de variations d'un terme, c'est-à-dire d'une séquence de mots. L'analyse de sa complexité algorithmique et les tests de performance ont montré que l'ajout de cette flexibilité impactait peu sa rapidité. Cette flexibilité se traduit par des bons résultats aux tâches partagées. IAMsystem est donc adapté aux contraintes d'un EDS : il est capable d'annoter très rapidement un grand corpus de documents, sans annotation préalable, avec des performances satisfaisantes.

Les performances d'IAMsystem étaient satisfaisantes à trois tâches partagées d'annotation sémantique de documents médicaux, en français et en espagnol. La tâche d'annotation des certificats de décès ressemble à celle d'annoter des antécédents médicaux dans une section "antécédents" d'un formulaire de l'EDS : les professionnels de santé listent les antécédents médicaux et chirurgicaux les uns à la suite des autres sans rédiger de phrase. Le séparateur de termes est généralement un saut de ligne ou une virgule.
La tâche de détection des diagnostics et des actes dans des notes cliniques en espagnol est similaire à l'annotation des documents longs comme les comptes-rendus de l'EDS. Cette tâche est plus complexe car les termes à détecter peuvent être composés de séquences de mots discontinus et le contexte à prendre en compte autour d'un mot est beaucoup plus grand.

L'avantage des tâches partagées comme celles organisées par l'initiative CLEF est une évaluation objective et une comparaison des performances de différents systèmes. La comparaison d'annotateurs sémantiques exige que chaque outil soit configuré en fonction des spécificités de la tâche d'annotation [60]. La compétition entre chercheurs permet la comparaison de nombreux systèmes paramétrés pour obtenir leurs meilleures performances. En dehors de tâches partagées et de benchmark commun, il est plus difficile de comparer les performances de différents annotateurs [60]. Une difficulté pour comparer des annotateurs est de prendre en compte leur évolution. Un annotateur comme cTakes [76] correspond à des annotateurs très différents en fonction de la version utilisée. Les auteurs ne mentionnent pas toujours la version de l'outil comparé.
Les tâches partagées ont cependant des limites. Le nombre de systèmes évalués et comparés dépend du nombre de participants. Les résultats obtenus par les participants dépendent du temps consacré au développement et au paramétrage de leur système. De nombreux autres systèmes, certains open source, ne sont pas évalués et comparés. Une seule tâche n'est pas suffisante pour

se comparer car les performances d'un système peuvent varier en fonction des spécificités d'un corpus, de la taille du jeu d'entraînement et de la terminologie. Notre participation à trois tâches partagées a montré l'hétérogénéité des performances d'IAMsystem qui obtient une F-mesure la plus basse à 0,52 pour la détection des actes médicaux et la plus haute à 0,897 pour la détection des espèces dans les documents médicaux. De plus, en comparant les systèmes par leur F-mesure, les tâches partagées favorisent autant le rappel que la précision. Le classement des systèmes serait différent si le rappel ou la précision étaient privilégiés. MetaMap favorise par exemple le rappel à la précision [85].
D'autre part, le choix d'un annotateur pour une utilisation quotidienne ne dépend pas uniquement de ses performances. D'autres caractéristiques comme son caractère open source, sa facilité d'utilisation ou sa rapidité doivent être prises en compte. La rapidité d'un outil d'annotation permet son intégration à un logiciel comme SmartCRF qui sera présenté au chapitre 6.

Il convient de discuter les limites d'IAMsystem. Sa principale limite est son incapacité à prendre en compte le contexte et donc à détecter les termes d'un dictionnaire lorsque la séquence de mots est discontinue dans un document. Par exemple, dans le contexte d'un patient atteint d'un cancer de la prostate, le terme IRM réfère au concept de "IRM prostatique" même si l'adjectif "prostatique" n'est pas mentionné explicitement ou à distance du mot "IRM". Ce problème était fréquent dans la tâche CodiEsp consistant à détecter des diagnostics et des actes dans des documents longs. La prise en compte du contexte nécessite de recourir à des algorithmes plus complexes. Les récents modèles de langage basés sur une architecture de type BERT obtiennent des performances à l'état de l'art dans de nombreuses tâches de TAL [97]. Un modèle de langage s'entraîne de façon non supervisée sur de grands volumes de données textuelles pour apprendre des représentations contextualisées des mots d'un corpus. Les modèles pré-entrainés sont ensuite peaufinés (fine-tuned) pour réaliser des tâches spécifiques qui requièrent peu de données d'entraînement grâce aux connaissances acquises lors de la phase de pré-entraînement [98]. Dans le domaine médical, la confidentialité des données de santé est un frein au pré-entraînement et au partage de ces modèles. Les modèles pré-entraînés peuvent mémoriser des données sensibles qu'un attaquant pourrait exploiter pour extraire des informations confidentielles [99]. Un autre frein à l'utilisation d'algorithmes d'apprentissage automatique dans le cadre d'un EDS est de disposer d'un corpus d'annotation suffisamment grand.

Dans la tâche Codiesp le faible nombre d'exemples annotés, environ 1,7% de la CIM-10, peut expliquer les performances sous optimales obtenues par ce type de modèle BERT. Les méthodes à base de dictionnaire ont l'avantage de ne pas avoir besoin de données annotées et leurs performances peuvent être satisfaisantes lorsque le jeu d'entraînement est insuffisant pour un apprentissage supervisé. Les modèles à base de dictionnaire peuvent aussi être combinés à des modèles d'apprentissage supervisé en pré-annotant des données pour produire des données d'entraînement[100]. Par rapport à une approche par dictionnaire, l'explicabilité d'une annotation est plus difficile à produire et la correction d'une erreur plus complexe à réaliser. Dans le cadre d'un EDS ne disposant pas d'un modèle de langage pré-entraîné et de données annotées par des experts médicaux pour une tâche spécifique, l'utilisation d'un algorithme à base de dictionnaire comme IAMsystem est justifiée.

Une deuxième limite est que l'ordre des mots dans le document est important. IAMsystem ne recherche pas de permutations des mots. Un médecin peut par exemple noter "humerus: fracture" et IAMsystem ne sera pas en mesurer de détecter le terme "fracture de l'humérus". La grammaire est cependant souvent respectée dans les documents cliniques et cette perte de sensibilité est souvent négligeable.

4.6 Conclusion

En résumé, IAMsystem est un annotateur généraliste à base de dictionnaire dont la complexité algorithmique dépend peu de la taille de la terminologie. Il est rapide et donc adapté aux contraintes d'un EDS où il est nécessaire de détecter des centaines des milliers de concepts médicaux dans des millions de documents sans annotation. Ses performances sont satisfaisantes lorsque la tâche d'annotation est relativement simple : le dictionnaire contient les termes à annoter et l'annotation ne dépend pas d'éléments contextuels. Lorsque le dictionnaire est exhaustif, une approche par dictionnaire est suffisante pour extraire les informations de documents courts comme les formulaires DxCare® et une approche plus complexe n'est pas nécessaire.
IAMsystem prend facilement en compte les abréviations mais requière un dictionnaire. Très peu d'abréviations sont utilisées pour mentionner les médicaments tandis qu'elles sont fréquentes pour mentionner des maladies. La création d'un dictionnaire d'abréviations était donc néces-

saire pour structurer les antécédents médicaux. Ce travail est présenté au prochain chapitre.

Chapitre 5

Détection des abréviations dans les dossiers patients informatisés

Il n'est pas nécessaire de créer une terminologie médicale des maladies pour structurer les antécédents médicaux car les terminologies CIM-10, MedDRA (Medical Dictionary for Regulatory Activities) et MeSH intégrées dans l'UMLS [74] contiennent déjà de nombreux concepts pour les identifier et les structurer. Des essais pour structurer les antécédents médicaux avec l'UMLS ont montré des mauvaises performances en termes de rappel à cause d'abréviations absentes de ces terminologies. Afin d'extraire les informations sur les antécédents médicaux, il était nécessaire de créer un inventaire des abréviations médicales en langue française. L'algorithme IAMsystem présenté au chapitre 4, configuré avec un dictionnaire d'abréviations, devrait permettre de structurer ces informations. La principale contribution de cette partie a été de construire un dictionnaire d'abréviations et un nouvel algorithme de détection des abréviations dans les dossiers patients informatisés.

5.1 Introduction

Les abréviations permettent de raccourcir les termes et sont largement utilisées en médecine [101, 102]. Une étape commune à de nombreuses applications de traitement automatique de la langue (TAL) consiste à reconnaître et à identifier les entités biomédicales dans un texte biomédical [103]. Les abréviations ajoutent de la complexité à la tâche de recon-

naissance des entités nommées car elles ne sont pas souvent couvertes par un lexique et elles sont souvent ambiguës. Une fréquence élevée d'abréviations dans un corpus peut entraîner une diminution des performances des applications en aval telles que les moteurs de recherche et les systèmes d'extraction d'information. Dans un DPI, la fréquence des abréviations dépend du type de document [104]. Stetson et al. ont constaté un pourcentage élevé d'abréviations utilisées dans les notes de transmission (26,88%), les notes cliniques (20,07%) et les résumés de sortie (3,87%). Comme la plupart des informations médicales contenues dans les DPI sont fournies sous forme de texte libre [70], il est important de traiter les abréviations lors des étapes de prétraitement d'une pipeline de TAL. Une solution classique consiste à construire un inventaire d'abréviations, c'est-à-dire une liste d'abréviations et leurs sens [105]. Un dictionnaire d'abréviations contient des paires d'abréviations et leurs expansions sous la forme <abréviation, expansion> [106]. La détection d'abréviations et la détection de leurs sens sont les deux tâches nécessaires pour construire un dictionnaire d'abréviations [107]. La première tâche vise à détecter et à répertorier toutes les abréviations dans un corpus, la seconde tente à identifier leurs sens.

La prochaine section donne une définition d'une abréviation et présente un état de l'art des méthodes de détection des abréviations. La section "Méthodes" présente les algorithmes utilisés pour détecter les abréviations dans un EDS. La section "Résultats" présente le dictionnaire d'abréviations et une évaluation de son contenu.

5.2 Définitions et travaux connexes

5.2.1 Définitions

Les abréviations sont des dérivés abrégés d'un mot et sont généralement prononcées comme leurs formes développées (ex: <mm, millimètre>) [108, 109]. Les acronymes et les sigles sont formés à partir des premières lettres des mots. Les acronymes sont prononcés comme des mots (ex: <SEP, Sclerose En Plaques>), les sigles sont prononcés lettre par lettre (ex: <AVC, Accident Vasculaire Cerebral>, prononcé A-V-C). Ainsi, une abréviation est une représentation courte d'un seul mot et un acronyme ou un sigle une représentation courte de plusieurs

Forme courte	Forme longue	Type
Neg	**Nég**atif	Premières lettres d'un mot
AVC	**A**ccident **V**asculaire **C**érébral	Premières lettres de chaque mot
Hb	**H**émoglo**b**ine	Première lettres de syllabes
CIP	**C**hambre **im**p**l**antable	Mixte 1ère lettres de mots / syllabes
KT	**Cath**éter	Début phonétique

Table 5.1: Les abréviations médicales peuvent prendre plusieurs formes. Exemples.

mots [108].

Dans les travaux de recherche, les auteurs définissent souvent l'abréviation de manière large pour inclure toutes les chaînes de caractères qui sont des formes abrégées de séquences de mots [110]. Sauf indication contraire, la définition large de l'abréviation sera utilisée dans ce chapitre.

Les acronymes et les abréviations sont souvent appelés "formes courtes". Développer une forme courte signifie la remplacer par sa forme longue [111].

Une abréviation est dite ambiguë si elle a plusieurs significations. Par exemple, <PR> peut signifier <professeur> ou <polyarthrite rhumatoïde>. La signification d'une abréviation dépend toujours du contexte dans lequel celle-ci apparaît. Une lettre unique comme <e> est toujours extrêmement ambiguë.

Les abréviations peuvent prendre plusieurs formes [108, 110]. Le tableau 5.1 présente plusieurs formes d'abréviations médicales. Les différentes manières de former une abréviation complexifient leur détection dans un document médical.

Les abréviations peuvent être classées en fonction de leur domaine d'utilisation. Les abréviations courantes sont comprises par tous les professionnels de la santé (<HTA, hypertension>), parfois même par le grand public (<ADN, acide désoxyribonucléique>). Une abréviation peut être spécifique à une spécialité, par exemple uniquement utilisée en cardiologie (<SEES, sonde d'entraînement électrosystolique>). Une abréviation peut également être créée de manière ad-hoc dans un contexte ou une tâche spécifique. Plus une abréviation est rare, plus elle est difficile à comprendre, non seulement pour les profanes mais aussi pour les cliniciens de différentes disciplines [107].

Les abréviations sont classées en fonction de la manière dont elles apparaissent dans un document. Les abréviations globales apparaissent dans les documents sans que la forme longue

soit explicitement mentionnée, tandis que les abréviations locales apparaissent avec leur forme longue dans le document [112]. La littérature médicale (articles scientifiques, recommandations médicales...) définissent très souvent une abréviation la première fois qu'elle apparaît dans le texte [110, 113, 114]. Sa forme longue est alors fournie entre parenthèses, par exemple *"Les AVC (accident vasculaire cérébral) sont la première cause d'handicap..."*. Les documents textuels d'un dossier patient informatisé, quant à eux, ne fournissent presque jamais le sens de l'abréviation [113, 115]. Cette distinction conduit au développement d'algorithmes différents pour la détection et l'identification des sens d'une abréviation selon que les abréviations soient globales ou locales.

5.2.2 Thématiques de recherche

Les travaux de recherche sur les abréviations tentent de résoudre trois problèmes distincts:

1. L'identification des abréviations dans un document

   L'objectif est de détecter les mots d'un texte qui sont des abréviations, par exemple le mot RCP dans "*Compte rendu de RCP*".

   Les méthodes d'identification des abréviations sont classifiées en trois groupes : méthodes basées sur des heuristiques, méthodes statistiques et méthodes à base d'apprentissage supervisé [116]. Les règles heuristiques les plus fréquentes pour la détection des abréviations sont la longueur des mots (par exemple, moins de 6 caractères), la présence d'une première lettre majuscule et le pourcentage de lettres majuscules [117, 118]. Les méthodes statistiques calculent des métriques pour identifier une abréviation. Par exemple, les acronymes sont souvent écrits en majuscule et peuvent être identifiées par la fréquence plus élevée de leur écriture majuscule que leur écriture en minuscule. En apprentissage automatique, la tâche est une classification binaire pour prédire si un mot est une abréviation ou non. Des annotations sont nécessaires pour entraîner un modèle. Les métriques statistiques et les règles heuristiques sont des caractéristiques souvent utilisées par les modèles pour réaliser une prédiction [119, 115]. Lorsqu'un sens potentiel (une forme longue) est détecté, les méthodes d'apprentissage automatique peuvent être utilisées pour prédire la probabilité d'une paire <forme courte, forme longue> [120].

2. La création d'un inventaire d'abréviations

   L'objectif est de recenser l'ensemble des abréviations et leurs sens dans un corpus donné. Par exemple, les deux sens de l'abréviation RCP sont :

   - <RCP, Réunion de Concertation Pluridisciplinaire>
   - <RCP, Réflexe Cutané Plantaire>

   Le sens d'une abréviation peut être découvert automatiquement ou saisi manuellement par un expert humain. Schwartz et al. [121] ont proposé un algorithme à base de règles pour vérifier si un terme candidat pouvait être l'expansion d'une abréviation :

   (a) Le premier caractère de la forme courte et de la forme longue doit être le même

   (b) Chaque caractère de la forme courte doit être présent dans la forme longue

   (c) Les caractères de la forme courte doivent apparaître dans le même ordre que ceux de la forme longue

3. Désambiguïser une abréviation

   La désambiguïsation est réalisée seulement si l'abréviation est connue pour avoir plusieurs sens. L'objectif est de prédire le sens d'une abréviation dans un contexte donné:

   - Réunion de Concertation Pluridisciplinaire dans "*Compte rendu de RCP*"
   - Réflexe Cutané Plantaire dans "*RCP en flexion*"

   Les méthodes d'apprentissage supervisées sont utilisées pour prédire la probabilité de chaque paire <forme courte, forme longue> selon des caractéristiques du contexte. Il est nécessaire de disposer d'exemples validés manuellement pour chaque abréviation ambiguë.

5.2.3 Algorithmes

De nombreux algorithmes de détection d'abréviation ont été développés sur des articles biomédicaux [110, 112, 122]. Zhou et al. [122] ont créé la base de données ADAM (another database of abbreviations) en extrayant 59 405 paires <forme courte, forme longue> des titres

et des résumés de MEDLINE. Le grand volume de résumés d'articles biomédicaux disponibles sur le web explique le fort intérêt des chercheurs pour ce corpus.

Ciosici et al. [117] ont proposé une méthode pour extraire automatiquement les formes courtes et leurs possibles formes longues à partir de grands corpus de textes non structurés. Leur algorithme repose sur la co-localisation de la forme longue et de la forme courte dans le corpus. Les algorithmes basés sur la co-occurrence des formes courtes et longues ne fonctionnent qu'avec des abréviations locales.

Les chercheurs se sont également intéressés à la détection des abréviations dans les dictionnaires biomédicaux, notamment dans le système de langage médical unifié (UMLS) [101, 113]. Les dictionnaires constituent une ressource importante pour détecter le sens d'une abréviation. Liu et al. [101] ont développé un algorithme qui extrait automatiquement les abréviations des termes de l'UMLS en utilisant un ensemble de règles heuristiques. Dans l'UMLS, les formes courtes et longues peuvent être reliées par un tiret ("AV - aortic valve"), le sens de l'abrévation peut être donné dans le libellé d'un terme ou identifié via les multiples termes associés à un concept [101].

Dans les DPI, l'identification des sens d'une abréviation, des paires <forme courte, forme longue> est plus compliqué. La forme longue peut être absente du corpus. Moins de méthodes ont été développées pour identifier automatiquement le sens des abréviations dans les dossiers médicaux possiblement en raison des problèmes de confidentialité pour accéder à ces données [113]. Moon et al. [113] ont détecté des abréviations candidates à partir d'un corpus de 352 267 notes cliniques en utilisant un ensemble de règles heuristiques. Les abréviations candidates les plus fréquentes (N=440) et leur texte environnant, avec une fenêtre de 12 mots, ont été montrées à des cliniciens pour trouver leur signification.

Xu et al. [115] ont détecté des abréviations dans des notes d'admission à l'hôpital en utilisant des listes de mots, des règles heuristiques et un arbre de décision basé sur les caractéristiques lexicales et statistiques des mots. Ils ont utilisé l'UMLS pour proposer des significations possibles des abréviations détectées. Ces méthodes n'utilisent pas le corpus pour détecter le sens des abréviations.

Hua Xu et al. [115, 103] ont proposé quant à eux une méthode de clustering pour détecter et regrouper les sens multiples d'une abréviation. Les exemples de chaque cluster sont présentés

à un expert médical qui détermine manuellement le sens de chaque cluster. Leur méthode a été implémentée dans CARD, un framework open source pour la détection et la désambiguïsation des abréviations cliniques [123].

Oleynik et al. [107] se sont intéressés pour leur part aux abréviations terminant par un point. Leur algorithme s'appuie sur la co-occurrence des tokens adjacents dans une liste de bigrammes pour détecter une abréviation et sa forme développée. Par exemple, "a. subclavia" peut être correctement résolu en "arteria subclavia" car cette dernière est la forme développée la plus courante. Leur méthode peut détecter automatiquement la forme longue d'une abréviation mais est limitée aux abréviations d'un seul mot.

Terada et al. ont aussi cherché à extraire des abréviations dans un corpus de rapports d'incidents d'aviation où leur définition n'était pas mentionnée explicitement [118]. Pour trouver les formes courtes et longues, ils ont combiné deux corpus : un corpus riche en abréviations et un corpus pauvre en abréviations. L'idée était d'utiliser le corpus riche en abréviations pour détecter des formes courtes et le corpus pauvre en abréviations pour détecter les formes longues. Pour détecter automatiquement le sens d'abréviations, leur méthode consistait à :

1. Extraire des formes courtes et longues candidates sur la base de règles heuristiques
2. Extraire le contexte de mots autour d'une abréviation et de ses formes longues candidates
3. Mesurer la similarité en utilisant la mesure du cosinus

Par exemple, les mots "flight" et "flat" ont satisfait les règles heuristiques pour être une forme longue candidate du mot "FLT". Comme "flight" a un contexte plus similaire (similarité cosinus = 0.43) que le second (similarité cosinus = 0.25), "flight" est retenu comme la forme longue de "FLT".

5.3 Méthodes

Nous avons sélectionné dans l'entrepôt de données du CHU de Bordeaux $186\,475$ (8%) comptes-rendus d'hospitalisation pseudonymisés et $3\,070\,352$ réponses à des formulaires Dx-Care® sur les antécédents du patient. Comme proposée par Terada et al. [118], l'idée était

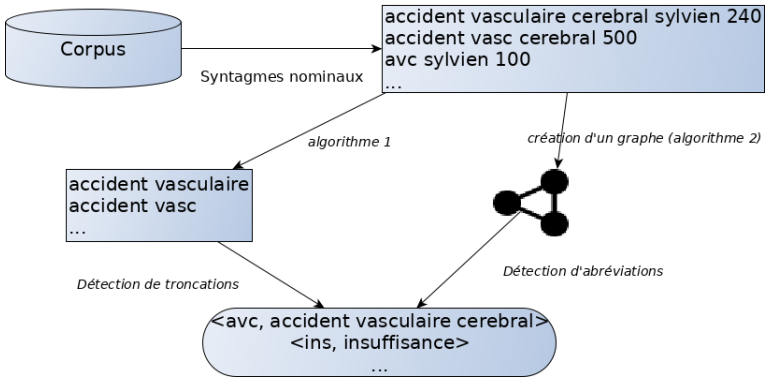


Figure 5.1: Présentation de la pipeline utilisée. Premièrement, les syntagmes nominaux et leur fréquence ont été extraites d'un corpus de documents cliniques. Ensuite, deux algorithmes ont détecté les abréviations. Le premier algorithme recherche la troncation d'un mot dans un terme composé de plusieurs mots (un n-gramme). Le second algorithme a utilisé des règles heuristiques et a mesuré la similarité du contexte pour détecter des abréviations et leurs sens.

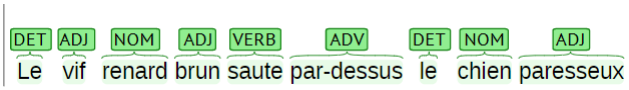


Figure 5.2: Visualisation de l'étiquetage morpho-syntaxique réalisé par TreeTagger. Suite à cette étape, les syntagmes nominaux "renard", "renard brun", "chien", "chien paresseux" ont été extraits avec une expression régulière et les formes lemmatisées ont été conservées.

de combiner deux corpus contenant respectivement une fréquence élevée (antécédents du patient) et faible (comptes-rendus de sortie) d'abréviations. Les notes cliniques contenaient des listes d'antécédents médicaux. Notre motivation pour sélectionner la section des antécédents du patient était de détecter les abréviations de maladies pour structurer ces informations. Dans les DPI, les abréviations sont largement utilisées sans mention de leurs sens. Notre hypothèse était que si une abréviation était utilisée, sa forme longue était utilisée quelque part dans le corpus. Pour détecter automatiquement les sens possibles d'une abréviation, l'idée était de réduire l'espace de recherche aux syntagmes nominaux du corpus. Un syntagme nominal est composé d'un ou plusieurs mots et a pour noyau un nom, il est identifié par étiquetage grammatical par un programme d'analyse morphosyntaxique. La figure 5.1 présente la pipeline.

5.3.1 Syntagmes nominaux

TreeTagger [124], un étiqueteur morpho-syntaxique (POS, Part-of-Speech), a été utilisé pour attribuer une catégorie grammaticale (nom, verbe, adjectif...) à chaque token d'un document. La figure 5.2 montre un exemple.

Ensuite, une expression régulière a filtré tous les termes multi-mots qui commençaient par un nom et contenaient un nombre maximum de 5 tokens. Le programme d'extraction des syntagmes nominaux était similaire à l'outil BioTex [125] : il utilise TreeTagger pour déterminer la catégorie grammaticale de chaque mot mais emploie une expression régulière au lieu d'une liste de motifs pour extraire les syntagmes nominaux d'un document. Le résultat de cette étape est un fichier tabulaire contenant tous les termes extraits avec leur fréquence. Les syntagmes nominaux se chevauchant ont aussi été extraits.

Formellement, soit *d* un document et E un ensemble d'étiquettes grammaticales. Pour le français, TreeTagger dispose de 33 étiquettes grammaticales différentes. La fonction de tokenisation de TreeTagger transforme le document en une séquence de mots (w_1, ..., w_n) puis TreeTagger associe à chaque mot w_i une étiquette $e_i \in E$ et un lemme $w_i^{'}$ par un module de lemmatisation. L'expression régulière recherche un motif dans la séquence d'étiquettes. Le programme extrait toutes les sous-séquences de lemmes ($w_j^{'}$, ..., $w_{j+w}^{'}$), avec une fenêtre w fixée à 5, où e_j est l'étiquette NOM (noun). Certaines séquences se chevauchent, c'est-à-dire que certains mots appartiennent à plusieurs séquences extraites. Après extraction des syntagmes nominaux dans chacun des documents, leur fréquence dans le corpus est dénombrée.

Cette première étape prend en entrée un corpus de documents et renvoie un ensemble de syntagmes nominaux avec leur fréquence noté SYN $\subset \Sigma^* \times Z^+$ avec Z^+ l'ensemble des nombres entiers positifs, Σ l'ensemble des lemmes et Σ^* l'ensemble des séquences de lemmes. Ci-dessous un exemple de 10 syntagmes nominaux avec leur fréquence.

data.txt

```
avc        100
accident vasculaire cerebral        100
accident vasc cerebral        30
avc sylvien        20
accident vasculaire cerebral sylvien        40
accident vasculaire cerebral ischemique        40
idm        100
idm ischemique        30
infarctus du myocarde        30
infarctus du myocarde ischemique        30
```

5.3.2 Algorithmes de détection

Deux algorithmes différents ont été développés pour rechercher des abréviations dans les syntagmes nominaux extraits. Le premier algorithme recherche uniquement la troncation d'un mot, le deuxième algorithme recherche tout type d'abréviations.

Algorithme 1 La stratégie de recherche par bigrammes d'Oleynik et al. [107] décrite ci-dessus a été utilisée pour détecter les troncations d'un seul mot. Par exemple, les termes "ins aortique" et "insuffisance aortique" sont rapprochés car "aortique" est un token commun et "ins" correspond aux premières lettres de "insuffisance". Cet algorithme présente plusieurs différences avec celui décrit par Oleynik et al.:

- La détection d'une troncation ne se limite pas aux mots terminant par un point. Tout token dans la liste des syntagmes nominaux est potentiellement la troncation d'un mot.
- La détection d'une troncation ne se limite pas aux bigrammes. Par exemple, la troncation "vasc" est détectée comme l'abréviation de "vasculaire" avec les termes "accident vasc cerebral" et "accident vasculaire cerebral".

Pour réaliser cette détection, l'algorithme utilise le sous-ensemble $SYN_{2+} \subset SYN$ contenant tous les syntagmes nominaux qui ont au moins 2 mots. IAMsystem, présenté au chapitre 4, a été utilisé pour rechercher les troncations. Il utilise le dictionnaire SYN_{2+} pour rechercher un syntagme nominal syn1 dans un autre syntagme nominal syn2. SYN_{2+} est à la fois le dictionnaire et le corpus d'annotation. IAMsystem est paramétré avec la méthode de comparaison *prefix* conçue pour cette tâche. Pour un lemme w_{input}, la méthode *prefix* renvoie tous les lemmes de l'ensemble Σ qui ont pour préfixe w_{input}. Par exemple, si le lemme est "ins", la méthode renvoie tous les lemmes débutant par "ins" comme "insuffisance". La détection par IAMsystem génère des couples ((syn1,n1), (syn2,n2)) $\in SYN_{2+} \times SYN_{2+}$, où syn2 est un terme du dictionnaire d'IAMsystem et syn1 un terme détecté dans le texte.
La fonction *check_conditions* : $\Sigma^* \times \Sigma^* \rightarrow \{Vrai, Faux\}$ vérifie les conditions suivantes:

- Les deux syntagmes nominaux sont différents : syn1 ≠ syn2
- Ils contiennent le même nombre de mots : la longueur de syn1 est à égale à la longueur de syn2
- Ils ont un seul mot de différence. Soit l la longueur de syn1 et syn2. Il existe un et un seul entier i, $1 \leq i \leq l$, tel que $syn1_i \neq syn2_i$.

Pour chaque couple qui respecte ces règles, la forme courte et la forme longue d'une abréviation peuvent être extraites. Soit $w_1' = syn1 \setminus syn2$ le lemme présent dans syn1 et absent de syn2 et $w_2' = syn2 \setminus syn1$ le lemme présent dans syn2 et absent de syn1. w_1' est le préfixe (forme courte) de w_2' (forme longue) puisque IAMsystem a détecté syn1 avec le terme du dictionnaire syn2. La fonction *extract_abb* : $\text{SYN}_{2+} \times \text{SYN}_{2+} \rightarrow (\Sigma \times \Sigma) \times Z^+$ transforme chaque couple ((syn1,n1), (syn2,n2)) en une paire ((forme_courte, forme_longue), n).
Le nombre entier n = min(n1, n2) correspond à la plus petite fréquence entre syn1 et syn2.
Des couples différents ((syn1,n1), (syn2,n2)) peuvent produire la même paire (forme_courte, forme_longue). La fonction *agreger* somme les valeurs n de chaque paire (forme_courte, forme_longue). Cet algorithme est décrit ci-dessous (pseudocode 4).

Algorithm 4 Algorithme 1 de détection des abréviations

Input: l'ensemble SYN des syntagmes nominaux extraits du corpus,
IAMsystem paramétré avec la méthode *prefix*,
les fonctions *check_conditions*, *extract_abb* et *agreger*
Output: un ensemble d'abréviations A ((forme_courte, forme_longue), n)

```
A := { }
abreviations := []
SYN2+ := sous-ensemble de SYN contenant les syntagmes nominaux qui ont au moins 2 mots
Charger le dictionnaire SYN2+ dans IAMsystem
for (syn1,n1) in SYN2+ do                    ▷ Itération sur chacun des syntagmes nominaux
    termes_detectes := IAMsystem(syn1)
    for (syn2,n2) in termes_detectes do      ▷ Les termes détectés sont des syntagmes nominaux
        if check_conditions(syn1,syn2) then
            abreviation := extract_abb((syn1,n1),(syn2,n2))   ▷ (forme courte, forme longue, n)
            Ajouter abreviation à abreviations
        end if
    end for
end for
A := agreger(abreviations)
retourne A
```

Algorithme 2

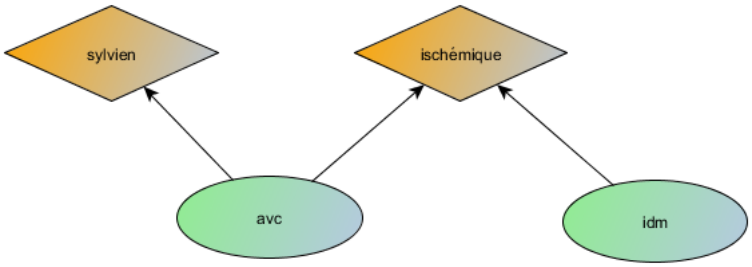
Le 2^{e} algorithme est basé sur l'hypothèse distributionnelle qui stipule que les mots qui apparaissent dans les mêmes contextes ont des significations similaires ou apparentées [126, 127]. Il permet de rechercher des abréviations de un à plusieurs mots. L'algorithme construit un graphe contenant des abréviations potentielles et leurs formes longues puis les paires <forme courte; forme longue> sont identifiées par les règles de Schwartz précédentes. Pour chaque paire, une mesure de similarité est calculée pour mesurer la similarité du contexte entre la forme courte et la forme longue. Son fonctionnement étape par étape est décrit ci-dessous.

1. Identifier les formes courtes potentielles

   Une abréviation candidate était définie comme tout premier mot d'un syntagme nominal contenant moins de 5 caractères. Dans l'exemple donné, les mots "avc" et "idm" sont des candidats potentiels. Le graphe est initialisé par ces deux nœuds.

2. Identification des mots du contexte

   Les mots contextuels sont ceux qui se trouvent à droite des abréviations candidates, à l'exception des déterminants et des prépositions (par exemple, du, de, le...). Dans notre exemple, les mots sylviens et ischémiques sont des mots contextuels. Le graphe est orienté, le sens de la flèche indiquant que ces mots contextuels sont dans le contexte de droite des abréviations potentielles.



3. Identification des formes longues candidates

Une forme longue candidate était tout terme, composé d'un ou plusieurs mots, situé à gauche d'un mot contextuel. Dans notre exemple, les termes "accident vasculaire cérébral" et "infarctus du myocarde" sont des formes longues potentielles.

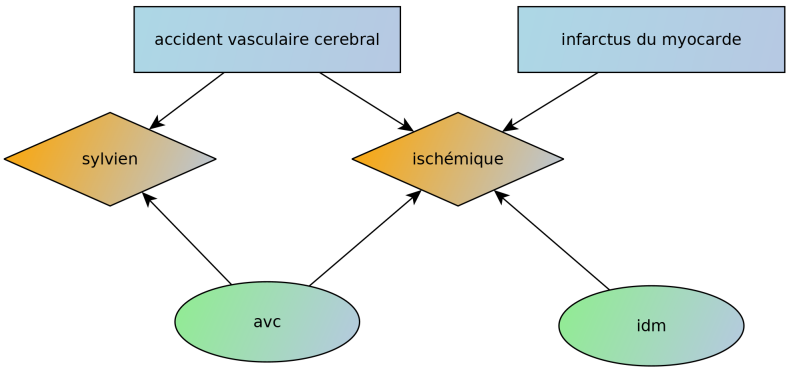


Figure 5.3: Représentation graphique du graphe créé pour la détection d'abréviations

4. Extraction des paires potentielles.

   A cette étape, le graphe contient trois types de nœuds : abréviations candidates, mots de contexte et formes longues candidates, comme le montre la figure 5.3. Chaque arête avait une valeur numérique qui correspond à la fréquence du mot contextuel. L'algorithme extrait toutes les paires <abréviation candidate, forme longue candidate> qui ont au moins un mot de contexte en commun et qui satisfont les règles heuristiques de Schwartz, décrites plus haut. Dans notre exemple, les paires <idm, infarctus du myocarde> et <avc, accident vasculaire cérébral> sont ainsi extraites.

5. Calcul de similarité

   Comme une abréviation a la même signification que sa forme longue, on s'attend à une distribution similaire de leurs mots contextuels. Chaque abréviation et forme longue candidate peut être représentée par un vecteur de mots contextuels.

   Il existe plusieurs métriques pour mesurer la similarité entre deux densités de probabilité [128]. Pour chaque paire, l'algorithme a calculé la similarité cosinus. Cette métrique a donné les meilleurs résultats lors des tests préliminaires et a également été choisie par Terada et al. [118].

Formellement, soit G = (V,A) un graphe formé d'un ensemble de nœuds V et d'arêtes A.

L'ensemble V est partitionné en trois sous-ensembles de nœuds V_s (s pour short form), V_c (c pour contexte) et V_l (l pour long form). Chaque nœud possède un libellé.

L'ensemble A ($A \subset V \times V$) est partitionné en deux sous-ensembles d'arêtes: A_s ($A_s \subset V_s \times V_c$, forme courte et son contexte) et A_l ($A_l \subset V_l \times V_c$, forme longue et son contexte). Par conséquent, $A \subset (V_s \cup V_l) \times V_c$.

Chaque arête, relation binaire entre deux nœuds, est issue d'un unique syntagme nominal. La fonction $comes\_from : A \rightarrow SYN$ associe une arête à un syntagme nominal. La fonction $has\_weight : A \rightarrow Z^+$ associe à chaque arête la fréquence du syntagme nominal :
$has\_weight = \{(a, n) \mid (a, (syn, n)) \in comes\_from\}$.
On numérote les nœuds contextuels de 1 à m, m = $|V_c|$, tel que $V_c = \{v_{c_1}, ..., v_{c_m}\}$.
La fonction $vectorize : (V_s \cup V_l) \rightarrow Z^m$ associe à tout nœud $v_{sl} \in (V_s \cup V_l)$ un vecteur de nombre entier de longueur m. Chaque élément i, $1 \leq i \leq m$, de ce vecteur est égal à $has\_weight(v_{sl}, v_{c_i})$ si la fonction est définie (donc si une arête existe) pour ce couple (v_{sl}, v_{c_i}) ou 0 sinon. Ainsi chaque nœud représentant une forme candidate courte ou longue est associé à un vecteur de nombre entier de longueur m. La représentation vectorielle des nœuds d'un graphe porte le nom de *graph embedding*. Le vecteur associé à chaque nœud $v_{sl} \in (V_s \cup V_l)$ représente simplement la fréquence des mots contextuels. Cette vectorisation permet de mesurer la similarité contextuelle de deux nœuds. Soit $\boldsymbol{x_s}$ le vecteur d'un nœud V_s et $\boldsymbol{x_l}$ le vecteur d'un nœud V_l. La cosinus similarité entre deux vecteurs $\boldsymbol{x_s}$ et $\boldsymbol{x_l}$ est égale à leur produit scalaire divisé par la longueur de chacun :

$$cosine\_similarity = \frac{\boldsymbol{x_s}.\boldsymbol{x_l}}{||\boldsymbol{x_s}||\,||\boldsymbol{x_l}||}$$

Cette cosinus similarité est égale à 0 si une forme courte et une forme longue n'ont aucun mot contextuel en commun. Cette cosinus similarité est calculée uniquement si les règles de Schwartz sont vérifiées entre deux nœuds.

Cet algorithme est décrit ci-dessous (pseudocode 5). Les numéros des étapes décrits dans les commentaires correspondent aux étapes décrites plus haut. Les autres fonctions utilisées par l'algorithme sont : *getNode* qui permet de trouver un nœud par son libellé, *getLabels* qui retourne les libellés d'un ou plusieurs nœuds, *validSchwartzRules* qui renvoie Vrai si les règles

de Schartwz sont vérifiées, Faux sinon.

Algorithm 5 Algorithme 2 de détection des abréviations

Input: l'ensemble SYN des syntagmes nominaux extraits du corpus,
les fonctions *validSchwartzRules, getNode, getLabels, vectorize*
Output: un ensemble d'abréviations AB ((forme_courte, forme_longue), similarité cosinus)

▷ Etape 1 et 2

SYN$_2$:= sous-ensemble de SYN contenant les syntagmes nominaux qui ont exactement 2 mots
$V_s, V_c, V_l, A, has\_weight$:= {}
for (syn,n) in SYN$_2$ **do**
  $[w_1, w_2]$:= syn
  if $|w_1| < 5$ **then**
    Ajouter un nœud avec le libellé w_1 à V_s s'il n'existe pas
    Ajouter un nœud avec le libellé w_2 à V_c s'il n'existe pas
    v_s := getNode(V_s, w_1)
    v_c := getNode(V_c, w_2)
    Ajouter une arête (v_s, v_c) à A
    Ajouter $((v_s, v_c), n)$ à has_weight
  end if
end for

▷ Etape 3

for (syn,n) in SYN **do**
  $[w_1, ..., w_n]$:= syn
  if $w_n \in getLabels(V_c)$ **then**
    $long\_form := w_1 \oplus ... \oplus w_{n-1}$ ▷ concaténation des lemmes précédents
    if $long\_form \notin$ labels(V_s) **then** ▷ Vérifier qu'il ne s'agit pas d'une forme courte
      Ajouter un nœud avec le libellé $long\_form$ à V_l s'il n'existe pas
      v_l := getNode($V_l, long\_form$)
      v_c := getNode(V_c, w_n)
      Ajouter une arête (v_l, v_c) à A
      Ajouter $((v_l, v_c), n)$ à has_weight
    end if
  end if
end for

▷ Etape 4 et 5

AB := {}
for v_s in V_s **do**
  for v_l in V_l **do**
    if validSchwartzRules(getLabels(v_s), getLabels(v_l)) **then**
      cossim := cosine_similarity(vectorize(v_s), vectorize(v_l))
      ajouter ((v_s, v_l), cossim) à AB si cossim ≠ 0
    end if
  end for
end for
retourne AB

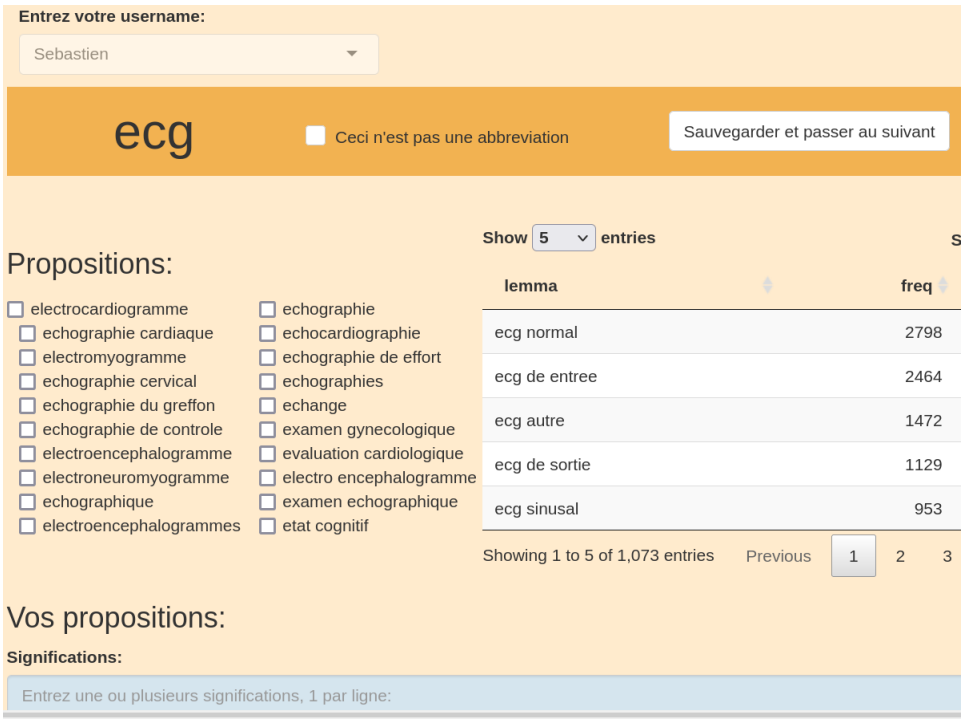


Figure 5.4: Interface de validation des abréviations. L'expert humain pouvait ignorer un mot si ce dernier n'était pas une abréviation, sélectionner le sens d'une abréviation et ajouter des sens. Les syntagmes nominaux contenant l'abréviation étaient montrés à l'utilisateur pour l'aider à comprendre le sens.

5.3.3 Validation manuelle

Une interface web a été développée pour valider les abréviations et les sens détectés (figure 5.4). L'utilisateur avait le choix entre ignorer les propositions si le mot n'était pas une abréviation, valider la signification d'une abréviation détectée par l'algorithme et ajouter des significations manuellement si le sens n'avait pas été détecté.

Dans cette approche semi-automatique, le système présentait aux utilisateurs jusqu'à 20 sens, classés selon leur fréquence de co-occurrence (algorithme 1) ou selon leur cosinus similarité (algorithme 2). Quatre experts médicaux ont examiné manuellement les résultats. Les syntagmes nominaux ont été affichés pour montrer les mots du contexte et faciliter la compréhension du sens. Si une forme longue n'était pas trouvée, les annotateurs devaient la saisir manuellement. L'utilisateur était libre d'utiliser tout type de ressource pour rechercher la signification d'une abréviation si nécessaire. Si aucune signification n'était trouvée, l'abréviation était ignorée.

Enfin, chaque abréviation validée a été revue une dernière fois pour numéroter les différents

sens d'une abréviation et sélectionner un libellé préféré à chaque sens. Ce procédé améliore la qualité d'un inventaire d'abréviations [129].

5.3.4 Jeux de données pour l'évaluation

Les abréviations médicales françaises provenant de la page web Wikipédia[1] (N=1 651) et de la page web d'un codeur médical[2] des hôpitaux universitaires de Genève (N=3 901) ont été scrapées pour construire un dictionnaire externe. Ensemble, ils comprenaient 3 393 formes courtes normalisées distinctes et 5 119 formes longues. Ce dictionnaire a été utilisé pour évaluer les performances de notre approche et identifier des abréviations non détectées (manque de sensibilité). Les abréviations qui n'ont pas été détectées par nos algorithmes mais par le dictionnaire d'abréviations externe ont été montrées à des experts médicaux, ainsi que leur fréquence dans le corpus, afin de vérifier les abréviations non détectées.

5.4 Résultats

Les statistiques descriptives du corpus utilisé dans cette étude sont données dans le tableau 5.2. Comme prévu, le pourcentage d'abréviations était élevé dans les notes cliniques, contrairement aux comptes-rendus d'hospitalisation. Les cinq abréviations les plus courantes étaient <mg, milligramme>, <kg, kilogramme>, <Dr, docteur>, <cp, comprimé> <ATCD, antécédents> dans les comptes-rendus d'hospitalisation et <HTA, hypertension>, <chir, chirurgical>, <FA, fibrillation atriale>, <med, médical> et <PTH, prothèse totale de hanche> dans les notes cliniques.

Le nombre d'abréviations détectées par algorithme est montré sur la figure 5.5.

[1] https://fr.wikipedia.org/wiki/Liste_d%27abr%C3%A9viations_en_m%C3%A9decine

[2] http://abreviationsmedicales.ch/

	Comptes rendus d'H°	section antécédents
Nombre de documents	186 475	3 070 352
Nombre de mots uniques	319 807	55 011
Nombre moyens de mots	606	3,6
Pourcentage d'abréviations	13,5%	45,3%

Table 5.2: Statistiques descriptives du corpus de documents utilisé pour la détection d'abréviations

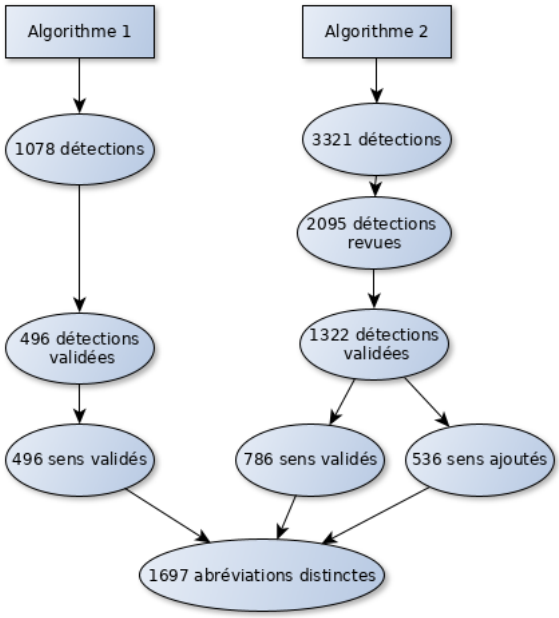


Figure 5.5: Nombre d'abréviations détectées et validées par les experts humains.

5.4.1 algorithme 1

L'algorithme 1 a détecté 1 078 abréviations dont 496 (46%) ont été validées manuellement. Sur ces 496 abréviations, 166 (33%) avaient plusieurs sens. Le mot "neuro" était l'un des mots les plus ambigus ; il peut désigner le neurologue ("avis neuro"), le type de douleur ("douleur neuro"), un trouble neurologique ("trouble neuro") et un service hospitalier ("transfert en neuro"). De nombreuses abréviations détectées à tort (faux positifs) étaient des mots de la langue française. Par exemple, "créatinine" n'est pas une abréviation de "créatininémie" bien qu'il ait la même signification dans certains contextes cliniques.

5.4.2 algorithme 2

L'algorithme 2 a détecté 3 321 abréviations potentielles. La validation manuelle a été arrêtée à la 2 095ème abréviation car la fréquence de l'abréviation (N<33) était trop faible pour fournir suffisamment de mots de contexte pour trouver sa signification. Sur les 2 095 abréviations validées, 1 322 (63%) ont été classées comme abréviations, dont 786 (37,5%) ont été trouvées par l'algorithme. Seulement 378 (18%) abréviations étaient des acronymes.

Plusieurs raisons expliquent pourquoi certains sens n'ont pas été détectés automatiquement par cette approche :

- De nombreuses formes longues n'apparaissaient pas dans le corpus, par exemple : <VAC, fermeture assistée par le vide>, <DAS, Disease Activity Score>, <PICC, cathéter central inséré en périphérie>.

- Les règles de Schwartz ont exclus certaines paires, par exemple : <KT, cathéter> (règle 1, la première lettre ne correspond pas) et <Rx, radiographie> (règle 2, la lettre x manque dans la forme longue).

- Certaines paires <forme courte, forme longue> ne partageaient pas le même contexte. Par exemple : <IEC, inhibiteur de l'enzyme de conversion>, les formes courte et longue étaient toutes les deux présentes dans le dictionnaire des syntagmes nominaux mais leur contexte local n'était pas similaire.

5.4.3 Evaluation

En comparant les abréviations détectées semi-automatiquement et le dictionnaire d'abréviations créé à partir de ressources externes, on observe un chevauchement de 571 paires <forme courte, forme longue>, soit 33.6% des abréviations détectées. De nombreuses abréviations non présentes dans les ressources externes ont donc été détectées par notre approche.

Pour 538 formes courtes, un nouveau sens a été détecté. Par exemple, l'abréviation "DR" avait pour seule signification "docteur" dans nos données alors que les ressources externes ne fournissent qu'une seule signification : "dernières règles".

Un total de 215 abréviations supplémentaires, non détectées par notre approche, ont été ajoutées grâce aux ressources externes. La plupart d'entre elles étaient des abréviations médicales courantes, par exemple <ELISA, enzyme-linked immunosorbent assay>. La majorité des abréviations non détectées avaient une forme longue qui n'apparaissait jamais dans le corpus. Leur sens était donc impossible à détecter dans les syntagmes nominaux.

Le dictionnaire d'abréviations a été créé en combinant les 1 697 formes courtes distinctes détectées par notre approche et les 215 abréviations ajoutées par le dictionnaire d'abréviations externe.

Ces résultats montrent que les dictionnaires d'abréviations extraits de ressources externes ne sont ni exhaustifs ni généralisables à d'autres corpus.

5.5 Discussion

À notre connaissance, il s'agit du premier inventaire d'abréviations issues de DPI français. Le dictionnaire contient 1 912 abréviations distinctes : 1 697 formes courtes détectées par notre approche semi-automatique et 215 détectées avec un dictionnaire externe d'abréviations médicales.

Dans ce chapitre, nous avons également présenté une nouvelle méthode pour détecter des abréviations et leur signification à partir d'un large corpus de documents médicaux. Comme les abréviations ne sont pas définies localement dans les DPI, le champ de recherche pour trouver une expansion peut être le corpus entier. Les approches décrites dans la littérature ne permettaient pas la détection d'abréviations multi-mots à partir d'un grand corpus de texte ou s'appuyaient sur des annotateurs humains pour trouver manuellement le sens. Notre idée principale était de limiter le champ de recherche aux syntagmes nominaux. Il avait déjà été démontré que la restriction de la recherche aux syntagmes nominaux pour la détection des abréviations dans les articles biomédicaux améliorait la précision [130]. En utilisant le contexte local, 378 acronymes ont été détectés.

Un autre avantage de notre approche par rapport à l'annotation manuelle de données textuelles cliniques est que les syntagmes nominaux sont des données agrégées qui ne posent pas de problèmes de confidentialité. L'agrégation de syntagmes nominaux, issue de documents pseudonymisés, rend très peu probable la présence d'une information directement identifiante. Cependant ces syntagmes nominaux comportaient parfois le nom de médecins hospitaliers.

La méthode proposée présente toutefois plusieurs limites. Premièrement, elle est entièrement basée sur les données cliniques et repose sur l'hypothèse forte que l'expansion d'une abréviation est présente ailleurs dans le corpus. Les résultats montrent que de nombreux acronymes, notamment issus de la langue anglaise (VAC, ELISA...) ne sont presque jamais exprimés par leur forme longue. Un travail antérieur, en langue autrichienne, avait montré que de nombreux acronymes étaient rarement développés dans les DPI [107].

Deuxièmement, une abréviation a été définie comme tout mot, étiqueté comme un nom, contenant moins de 5 caractères. Cette définition large a conduit à une faible précision et un

grand nombre de faux positifs. Les exemples positifs et négatifs d'abréviations annotés dans l'interface pourront permettre d'entraîner des modèles supervisés pour prédire si un mot est une abréviation ou non.

Troisièmement, la méthode n'est pas entièrement automatique car elle nécessite une validation manuelle. La revue manuelle peut être jugée comme un avantage car elle garantit la qualité d'un inventaire d'abréviations [131].

Quatrièmement, pour développer une abréviation, les utilisateurs s'appuient fortement sur les informations contextuelles. Une fenêtre courte n'est pas toujours suffisante, pour un humain ou une machine, pour développer une abréviation, en particulier lorsque l'abréviation est peu fréquente. L'approche proposée a bien fonctionné pour les abréviations fréquentes mais a échoué à plusieurs reprises pour découvrir la signification des abréviations peu fréquentes.

En plus des limites des algorithmes présentés ci-dessus, l'étude présente d'autres limites. Premièrement, l'évaluation a été réalisée en utilisant un "silver standard", un inventaire d'abréviations scrapé sur Internet. Aucun gold standard n'a été créé en annotant manuellement les abréviations dans leur contexte. Par conséquent, le rappel n'a pas pu être mesuré. En plus des problèmes de confidentialité, la création d'un gold standard se heurte à la difficulté d'annoter manuellement un très grand nombre de documents car certaines abréviations sont très peu fréquentes. Cela est particulièrement vrai pour les abréviations ambiguës qui ont plusieurs significations et une fréquence très déséquilibrée de leurs formes longues.

Deuxièmement, l'importance de chaque abréviation pour la finalité de recherche d'information n'a pas été prise en compte. Une forme longue qui n'apparaît jamais dans un corpus a peu d'importance. En effet, il n'est pas nécessaire de réaliser une expansion de sa forme courte pour augmenter les résultats de la recherche de documents.

Troisièmement, la désambiguïsation des abréviations n'a pas été abordée. Lorsqu'une abréviation a plusieurs significations, un algorithme de désambiguïsation est nécessaire pour identifier son sens dans un contexte donné. Les abréviations non ambiguës sont néanmoins majoritaires et n'ont pas besoin d'algorithme de désambiguïsation [117]. La principale limite de validité externe de cette étude est le type de documents sélectionnés. D'autres documents textuels peuvent contenir d'autres abréviations. Par exemple, les abréviations des infirmières pourraient être totalement différentes de celles des médecins. Ainsi, ces résultats ne peuvent

pas être extrapolés à d'autres documents.

5.6 Conclusion

Dans ce chapitre, nous avons décrit la construction d'un inventaire d'abréviations médicales françaises en utilisant une approche data-driven et un dictionnaire externe d'abréviations médicales scrapé sur le web. Nous avons proposé un nouvel algorithme basé sur les syntagmes nominaux pour détecter et développer les abréviations à partir d'un large corpus de textes cliniques. Cet inventaire a permis d'améliorer les performances de la structuration des antécédents médicaux dans les formulaires DxCare®. Une étude sera menée pour évaluer objectivement le rappel et la précision de la structuration de ces informations.

Chapitre 6

SmartCRF

De nombreuses informations sont trop complexes à extraire automatiquement ou nécessitent des algorithmes spécifiques trop chronophages à créer et déployer. En routine, les chercheurs ont besoin de disposer d'une interface interactive pour faciliter la revue des dossiers et l'extraction d'information. L'objectif de SmartCRF était de remplacer la revue manuelle des dossiers dans DxCare®, très chronophage, par une revue assistée par ordinateur. SmartCRF réutilise les briques développées aux chapitres précédents pour faciliter la recherche d'information dans un dossier : la terminologie des médicaments Romedi, les abréviations médicales, les syntagmes nominaux extraits au chapitre 5 et l'annotateur IAMsystem.

6.1 Introduction

Pour rechercher des patients éligibles à une étude, des outils de requêtage sont fournis aux utilisateurs pour interroger les données d'un EDS (figure1.3). Cependant, ces outils offrent rarement de bonnes performances en termes de sensibilité et de spécificité. Une première raison est que la majorité des informations cliniques est présente dans le texte libre [4] et donc difficilement requêtable. Une deuxième raison est que les données structurées sont limitées par le nombre de codes des terminologies utilisées. Certaines maladies ou évènements indésirables n'ont par exemple pas de codes CIM-10 spécifiques [132]. Une étude a montré que le texte libre était nécessaire pour vérifier 60% des critères d'inclusion d'un essai clinique sur la leucémie lymphoïde chronique et près de 80% des critères d'inclusion d'un essai clinique sur le cancer de

la prostate [133]. Une autre étude a montré que 20% des patients prenaient un médicament qui ne figurait pas dans les données structurées de prescription [134]. Les informations médicales d'un DPI ne sont pas toutes structurées et ne sont pas toutes structurables.

La recherche dans le domaine de traitement automatique de la langue (TAL) visant à exploiter les données en texte libre évolue rapidement. L'utilisation en routine d'algorithmes de TAL requière cependant des efforts importants pour les évaluer et les déployer. Dans de nombreuses situations, leurs performances ne sont pas encore suffisantes pour remplacer un expert humain afin de vérifier automatiquement l'éligibilité d'un patient dans une étude ou pour remplir l'eCRF d'une étude [15].

La norme en matière de recherche clinique reste la revue manuelle des dossiers médicaux par des personnes formées à cet effet [135]. Cette activité est très chronophage car il peut être nécessaire de parcourir l'intégralité du dossier patient pour trouver une information. Compte tenu du temps et des efforts nécessaires à l'examen manuel des dossiers, des outils ont été développés pour faciliter cette tâche [135]. Par exemple, de nombreuses études ont montré que l'utilisation d'un moteur de recherche permettait des gains de temps significatifs pour trouver une information [135]. Une machine est capable de rechercher et d'identifier des termes d'intérêt beaucoup plus rapidement qu'un humain. Ces outils permettent de combiner la compréhension du langage naturel d'un humain et la puissance de calcul d'une machine dans une interface où se déroule les interactions homme-machine. L'objectif est de remplacer la revue manuelle des dossiers par une revue assistée par ordinateur afin de gagner du temps dans les tâches d'identification de patients éligibles et de complétion d'un eCRF.

En plus d'augmenter l'efficacité à l'accomplissement de ces tâches, l'utilisation d'une interface dédiée peut permettre d'enrichir un EDS. En effet, une étude génère des données structurées de qualité qui pourraient être réutilisées par d'autres études et l'éligibilité d'un patient à une étude constitue aussi une nouvelle information intéressante pour un EDS. Les cohortes rétrospectives créées manuellement par les chercheurs après revue des dossiers pourraient alimenter l'EDS. Ces données peuvent aussi servir de données d'entraînement et d'évaluation à des algorithmes de TAL qui sont généralement chronophages et coûteux à produire. Une interface dédiée pour revoir les dossiers présente donc un triple intérêt :

1) gagner du temps pour la réalisation d'une étude particulière

2) générer des données structurées de qualité pour la réalisation d'études ultérieures

3) générer des données d'entraînement pour des algorithmes de classification automatique des dossiers et d'extraction d'information

On définit un *entrepôt de données apprenant* comme un EDS capable de s'enrichir avec les interactions des utilisateurs dans une interface dédiée. L'idée est qu'un utilisateur peut activement contribuer à la mise en qualité des données d'un EDS en classifiant un dossier ou en annotant un document. Chaque interaction est une opportunité de structurer une information et d'alimenter l'EDS avec des données structurées de qualité. L'augmentation du nombre d'utilisateurs devrait permettre d'augmenter la structuration des données d'un EDS et donc de faciliter l'utilisation secondaire des données. Ce concept est proche du *système de santé apprenant* (learning health system) introduit par Etheredge en 2007 [136]. Dans ce système, chaque interaction entre le patient et les services de santé apporte une information pouvant permettre d'améliorer les connaissances en clinique et en santé publique [137].

Objectif

Notre objectif était de développer un outil de visualisation d'un DPI permettant à un ingénieur de recherche de vérifier rapidement l'éligibilité d'un patient et de trouver une information dans un dossier. Notre objectif secondaire était de montrer la faisabilité de réutiliser les données générées par ces ingénieurs à travers l'interface pour enrichir un EDS.

La suite de ce chapitre est organisée de la façon suivante. La partie état de l'art présente les travaux similaires réalisés dans ce domaine. La partie méthodes présente l'implémentation réalisée au CHU de Bordeaux et l'étude menée pour évaluer les performances de l'interface.

6.2 Etat de l'art

6.2.1 Visualisation de dossiers patients

L'organisation des données en silos dans un SIH nécessite de naviguer dans divers modules pour comprendre les informations d'un DPI. Il faut par exemple ouvrir un onglet *Résultats bactériologiques* pour consulter un résultat de microbiologie et un antibiogramme, un deux-

ième onglet *Prescriptions médicamenteuses* pour rechercher les antibiotiques prescrits et comparer manuellement l'adaptation de l'antibiothérapie aux résultats, un troisième onglet *Résultats biologiques* pour comprendre une adaptation de dose (insuffisance rénale) puis un quatrième onglet pour rechercher la justification d'une antibiothérapie dans les notes médicales et d'éventuelles allergies médicamenteuses. La visualisation fragmentée de l'information dans différents onglets rend difficile la compréhension du dossier [138]. Cette visualisation est loin d'être efficace car elle oblige les professionnels de santé à reconstituer mentalement la séquence d'évènements.

La visualisation de données est une discipline qui étudie les représentations visuelles et les techniques d'interaction pour permettre aux utilisateurs de voir, d'explorer et de comprendre de grandes quantités d'information. Elle vise à combiner la puissance de traitement des ordinateurs modernes avec la cognition et les capacités visuelles de l'homme [2]. L'interaction de l'utilisateur avec l'interface est centrale dans la visualisation de données.

Dans une revue de la littérature, West et al. [139] liste les difficultés de la visualisation d'un dossier patient : la quantité importante d'information, la petite taille d'un écran d'ordinateur, la présence de nombreux types de données (quantitative, qualitative, textuelle), la temporalité de chaque information au cours du temps et les liens implicites dans les informations.

Plusieurs prototypes ont été développés pour proposer des visualisations intuitives de DPI. Ainsi, développé en 1996 aux Etats-Unis, LifeLines [140] est l'un des premiers prototypes proposant une ligne de vie et une vue d'ensemble d'un dossier patient. LifeLines permet la visualisation de nombreuses variables médicales sur le même écran (figure 6.1). Les segments de ligne sont répartis le long de l'axe horizontal du temps qui peut être agrandi et déplacé pour révéler plus ou moins de détails. LifeLines regroupe les éléments d'une même source de données dans des groupes. Par exemple, tous les tests biologiques appartiennent à un même groupe, et tous les traitements médicaux appartiennent à un autre groupe. Les groupes sont dépliables et repliables afin de choisir les plus importants et éviter d'encombrer l'écran d'informations inutiles.

Selon Ben Shneiderman, auteur de LifeLines, une visualisation se déroule en trois étapes : une vue d'ensemble, la possibilité de zoomer et de filtrer les informations recherchées et enfin l'affichage d'éléments précis sur demande de l'utilisateur. Ce principe porte le nom Mantra de

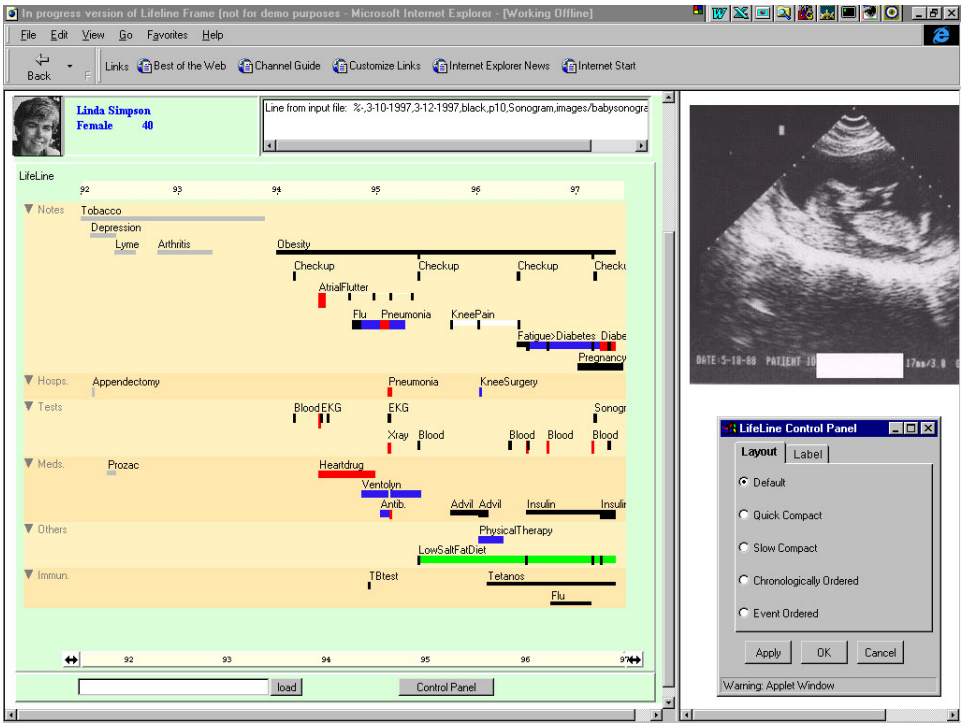


Figure 6.1: Interface du logiciel LifeLines. Les lignes horizontales représentent les événements et les épisodes d'un même dossier patient. La couleur, la hauteur et les légendes des lignes sont utilisées pour transmettre les informations. Un panneau de détails à droite montre des informations supplémentaires sur un élément sélectionné, tel qu'une image échographique. D'après Rind et al. [2].

Shneiderman : overview first, zoom and filter, details-on-demand [141]. LifeLines a été une source d'inspiration de nombreuses autres interfaces [2].

En 2011, Rind et al. [2] ont réalisé un état de l'art sur les méthodes de visualisation et d'exploration d'un DPI. L'objectif de leur revue de la littérature était d'étudier les idées des interfaces existantes pour fournir un support au développement des interfaces futures. Les auteurs ont étudié et recensé quatorze systèmes différents. Les interfaces de ces systèmes ont été réalisées pour de multiples situations cliniques : Midgaard [142] a été développée pour visualiser des données de soins intensifs, WBIVS [143] pour la surveillance de patients transplantés, VIE-VISU[144] pour la néonatologie et VisuExplore [145] pour suivre l'évolution du diabète (figure 6.2).

La visualisation combinée de variables catégorielles et quantitatives au fil du temps est une caractéristique de nombreux systèmes. Certains systèmes utilisent un glyphe, objet graphique possédant différents attributs géométriques et visuels utilisé pour coder des informations complexes. Par exemple le glyphe de VIE-VISU encode les informations de quinze variables dif-

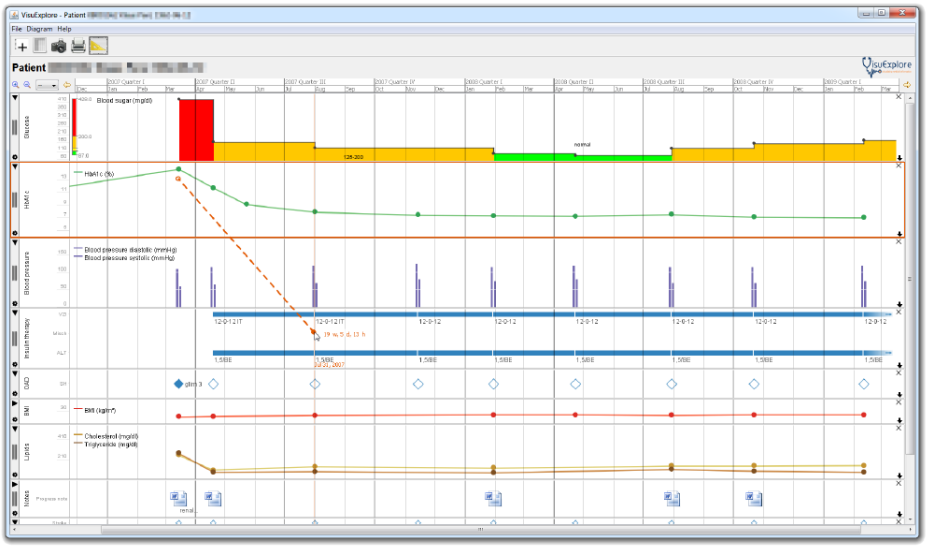


Figure 6.2: Interface du logiciel VisuExplore pour suivre l'évolution du diabète. D'après Rind et al. [2].

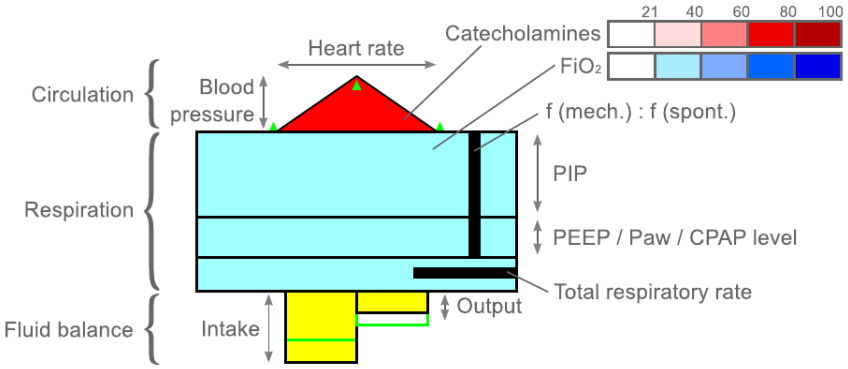


Figure 6.3: Glyphe du logiciel VIE-VISU utilisé en néonatologie. D'après Rind et al. [2].

férentes (figure 6.3).

Un DPI est souvent représenté verticalement sur une ligne de temps horizontale appelé ligne de vie ou timeline qui permet de représenter l'intégralité des données d'un patient et d'identifier certaines séquences d'évènements. Les différentes variables sont souvent représentées par des couleurs ou des icônes différentes. La manière la plus courante de visualiser des données catégorielles est d'utiliser des icônes pour les événements ponctuels et des segments de ligne pour les événements qui ont une durée.

Rind et al. [2] distinguent les différentes interfaces par les fonctionnalités d'interaction offertes à l'utilisateur. La majorité des systèmes utilisent une représentation temporelle pour afficher les données et permettre à l'utilisateur de les explorer. Certains outils permettent de fil-

trer les éléments d'un dossier. Par exemple la fonctionnalité de recherche de LifeLines permet de mettre en sur-brillance les éléments recherchés, rechercher "diabète" met en sur-brillance les glycémies et les traitements diabétiques par exemple. Certains outils permettent à l'utilisateur de configurer la visualisation pour l'adapter à ses besoins. Afficher le lien entre différents items, trouver des relations implicites dans les données est une tâche complexe à réaliser et peu implémentée dans les interfaces.

Rind et al. [2] jugent difficile la comparaison de différentes interfaces car elles n'ont pas les mêmes finalités ni les mêmes données à représenter. Toutefois, l'un des critères important d'évaluation est son utilisation en situations réelles. Certains systèmes n'ont pas dépassé le prototypage ou sont encore au stade de développement et ne sont pas encore déployés dans un environnement de soins. Les vraies données peuvent être différentes des données utilisées lors du développement et certains systèmes ont besoin de s'adapter aux vraies données d'une structure de soins. D'après les auteurs, les systèmes doivent être personnalisés pour chaque établissement de santé à cause de l'hétérogénéité des différents environnements.

6.2.2 Outils similaires

Cette section aborde les outils similaires de visualisation d'un dossier patient à partir de données hospitalières et pour des finalités de recherche clinique.

EMERSE

L'Electronic Medical Record Search Engine (EMERSE) est une interface de visualisation d'un dossier patient muni d'un moteur de recherche développé dans le centre hospitalier universitaire du Michigan. De nombreux papiers ont été publiés par les auteurs d'EMERSE et permettent de comprendre son évolution. La première version a été développée en 2005 par le docteur David Hanauer, médecin spécialisé en informatique médicale avec une formation initiale en pédiatrie. Dans le premier papier décrivant EMERSE [146] les auteurs expliquent qu'un moteur de recherche est un outil précieux pour naviguer dans les grandes quantités de données en texte libre d'un dossier patient. Paradoxalement, ils sont souvent absents des logiciels hospitaliers. Le déploiement d'EMERSE a été un succès immédiat avec des retours très positifs des utilisateurs.

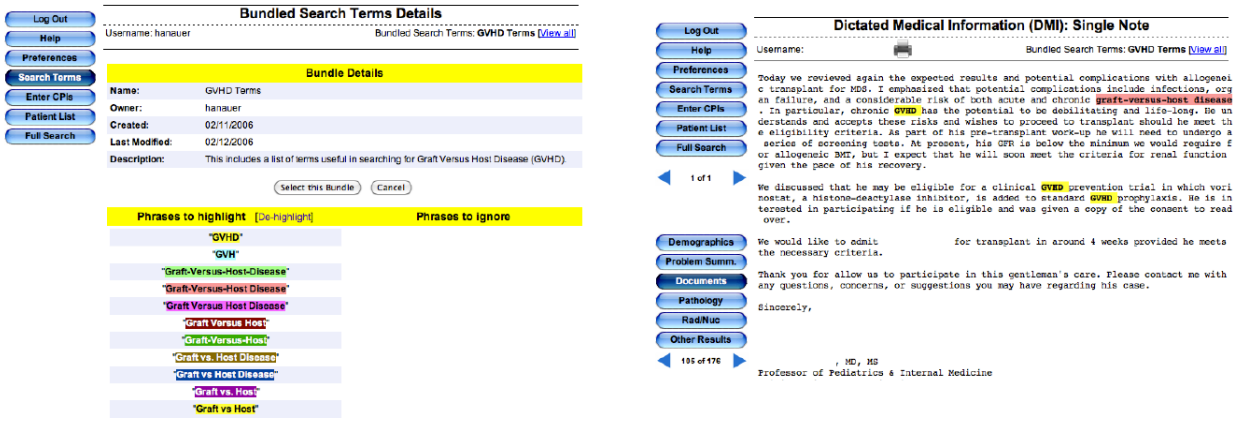


Figure 6.4: Capture d'écran du système de recommandation d'EMERSE. A gauche le dictionnaire *search term bundle* GVHD contient toutes les variations lexicales du concept "Graft Versus Host Disease". A droite, ce groupe de termes est utilisé pour la recherche d'information dans un document textuel. D'après Yang et al.[148]

L'utilisation du moteur de recherche était initialement réservée à la recherche clinique puis a été étendue aux cliniciens et au personnel en charge du codage médical à des fins de tarification de l'activité hospitalière [147].

Un papier de 2011 décrit la construction du module de recommandation de EMERSE [147]. Ce dernier a été conçu suite au constat que les utilisateurs perdaient du temps à rechercher un concept médical pouvant s'écrire de multiples façons. L'objectif du système de recommandation est d'afficher des termes alternatifs pertinents à l'utilisateur (figure 6.4). Ce système de recommandation s'appuie sur deux outils différents. Le premier outil est basé sur des dictionnaires médicaux personnalisés contenant des abréviations et des méthodes de correspondance de chaînes de caractères pour prendre en compte des variations lexicales et des fautes d'orthographe. Le deuxième outil repose sur une approche collaborative où les utilisateurs construisent et partagent leurs dictionnaires. Ces dictionnaires sont appelés *search term bundle*. Chacun possède un libellé et permet de regrouper un ensemble de termes associés. Par exemple, le dictionnaire *stade tumoral* regroupe 202 termes associés (Gleason, TNM...). D'après les auteurs, le système de recommandation a permis d'améliorer la satisfaction des utilisateurs.

Une étude a été menée pour analyser les termes recherchés dans EMERSE et l'utilisation du système de recommandation [148]. Sur une période de 4 ans, plus de 200 000 requêtes ont été analysées. La majorité des termes recherchés ont été réalisés par des groupes de termes ce qui confirme l'utilité du système de recommandation. Une grande proportion (20%)

de requêtes contenaient des abréviations. Dans cette étude, les auteurs ont aussi analysé la proportion des termes présents dans les terminologies médicales. Seulement 63,6% des termes recherchés étaient présents dans l'UMLS contenant environ 800 000 termes en anglais au moment de l'étude. Les auteurs ont catégorisé ces termes en les regroupant dans les groupes sémantiques de l'UMLS. Les observations cliniques sont la principale catégorie (28%), ils comprennent les signes cliniques, les symptômes et les maladies. Viennent ensuite les médicaments (12,2%) et les actes (11,7%).

Plusieurs papiers décrivent l'utilisation et les performances d'EMERSE pour une tâche donnée. En 2009, le système EMERSE a été utilisé pour la surveillance des complications post-opératoires [149]. Une recherche large a été réalisée pour identifier les patients suspects d'avoir une complication post-opératoire puis les dossiers ont été revus avec le logiciel EMERSE. Le système de recommandation a été paramétré avec un dictionnaire spécifique contenant des termes en lien avec ces complications. Comparé au circuit classique de signalement, l'utilisation d'EMERSE a permis d'identifier des faux négatifs, c'est-à-dire des patients non inclus dans le système de surveillance alors qu'une complication a été détectée avec EMERSE.

Une étude a été menée pour évaluer la rapidité à identifier des patients éligibles à une étude avec EMERSE [135]. Dans cette étude, les dossiers des patients contenaient en moyenne 102 notes cliniques. Sur 1383 patients screenés, seulement 13 étaient éligibles. Les chercheurs ont mesuré le temps mis pour revoir les dossiers et trouver les patients éligibles. Les performances pour identifier les patients éligibles étaient similaires entre l'utilisation de l'interface EMERSE et le logiciel métier CareWeb. Le temps passé à revoir les dossiers était significativement plus faible avec EMERSE. Un autre résultat de cette étude était de montrer que les utilisateurs expérimentés allaient plus vite que les utilisateurs novices d'EMERSE.

Dans un papier de 2015, Hanauer et al. expliquent que le succès d'EMERSE réside à la fois dans la simplicité et la grande utilité d'un moteur de recherche. Environ 10 ans après son lancement, plus de 750 études ont mentionné avoir utilisé le système EMERSE. Trouver des termes dans un document est une tâche assez simple à réaliser par une machine et pourtant très chronophage pour un humain. Un moteur de recherche ne fait que localiser un terme dans un dossier et laisse la complexité de l'interprétation de l'information à un humain. Un autre facteur de ce succès selon les auteurs est que les moteurs de recherche sont très utilisés sur le web et

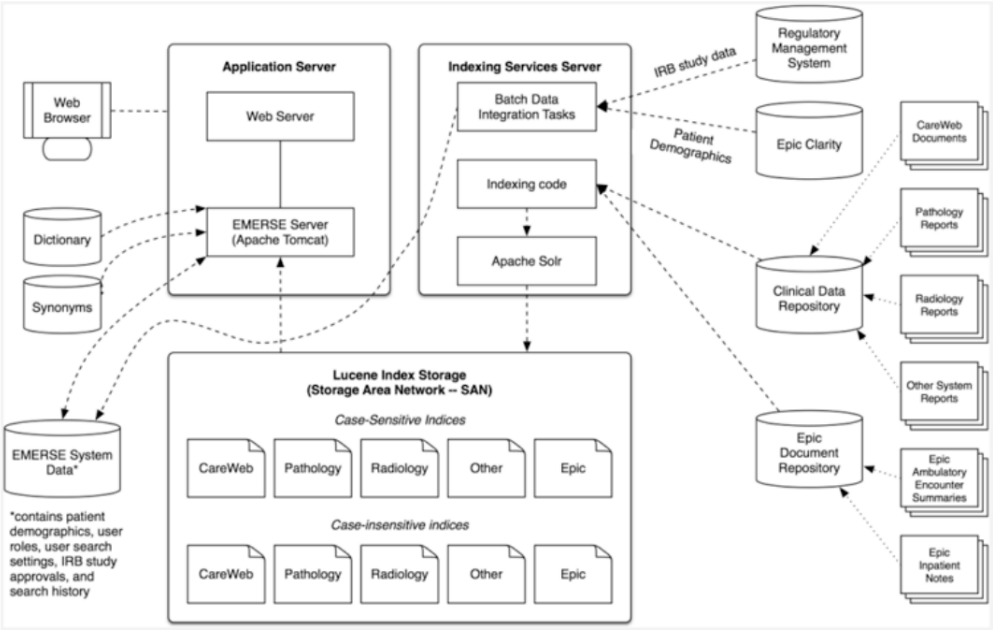


Figure 6.5: Architecture d'EMERSE. D'après Yang et al. [148].

que leur fonctionnement est donc intuitif.

D'après Hanauer, les méthodes avancées de TAL ne sont pas suffisamment matures pour comprendre une situation clinique. Pour des tâches complexes, les performances des machines ne sont pas suffisantes pour une extraction automatique. Elles ne peuvent donc pas remplacer la revue d'un dossier par un humain. Les auteurs décrivent EMERSE comme un *système d'intelligence augmentée, dans lequel le logiciel aide une personne à effectuer son travail plus efficacement* [150]. De nombreux projets similaires n'ont pas dépassé le stade de prototype tandis qu'EMERSE est utilisé depuis plusieurs années [150]. Le développement d'EMERSE a connu de nombreuses révisions depuis sa première implémentation selon les retours d'utilisateurs et l'évolution des technologies informatiques. En 2020, EMERSE était utilisé dans trois hôpitaux universitaires : Michigan, Caroline du Nord et Cincinnati. En 2021, un dépôt GitHub a été créé pour partager le code qui est accessible sur demande auprès de l'université du Michigan. L'utilisation du logiciel est gratuite pour un usage académique.

L'architecture d'EMERSE est présentée figure 6.5. Les documents en texte libre sont extraits de différents logiciels et centralisés dans un Clinical Data Repository. Seules les notes cliniques et leurs métadonnées sont indexées dans Lucene [150].

Dans l'interface, les documents sont affichés par source de données. Les résultats des

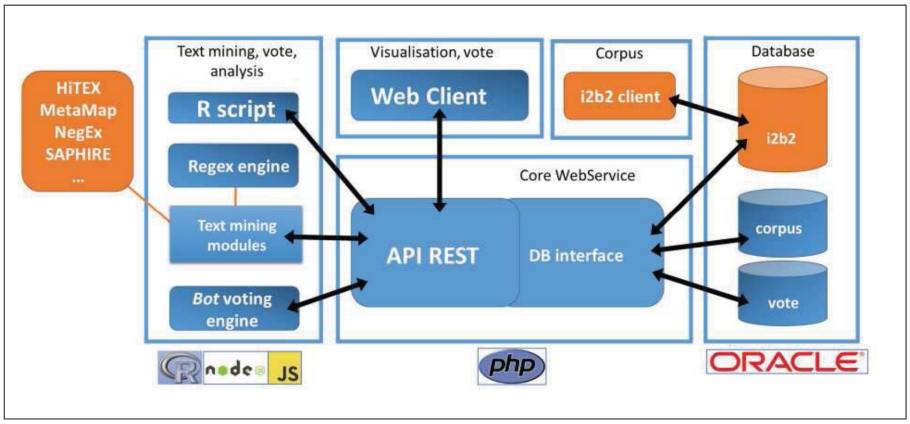


Figure 6.6: Architecture de FASTVISU d'après Escudie et al. [14]

recherches sont présentés du plus récent au plus ancien. Une zone de texte, appelée Snippet, affiche une petite portion de texte où le mot recherché apparaît. En cliquant sur celui-ci, le document apparaît et les mots recherchés apparaissent en sur-brillance. Tous les documents d'un patient sont visibles pour montrer à l'utilisateur les documents où le terme recherché apparaît. Pour la recherche clinique, une liste pré-définie de patients à revoir peut être chargée. Cette liste peut être générée ad-hoc ou provenir du résultat d'une requête i2b2 : les auteurs ont développé un module permettant de transférer les résultats d'une recherche de patients directement dans le système EMERSE. En 2015, les documents n'étaient pas dé-identifiés même pour les activités de recherche.

FASTVISU

A l'HEGP, Escudie et al. ont développé FASTVISU, un outil pour vérifier rapidement les critères d'éligibilité dans un dossier médical [14].

FASTVISU possède une architecture orientée services (figure 6.6). Les données d'un DPI sont extraites d'un entrepôt de données i2b2. Une interface web permet de visualiser les données en texte libre (figure 6.7). Dans l'exemple de la figure 6.7 l'utilisateur vote pour la présence d'un diabète de type 1 et d'une thyroïdite auto-immune. Des extraits sont présentés pour lui éviter de parcourir l'intégralité du dossier à la recherche d'information. FASTVISU utilise un module de TAL à base d'expressions régulières pour pré-annoter les concepts médicaux dans les documents. Plusieurs utilisateurs revoient les dossiers en parallèle puis les votes en désaccord sont revus pour trouver un consensus.

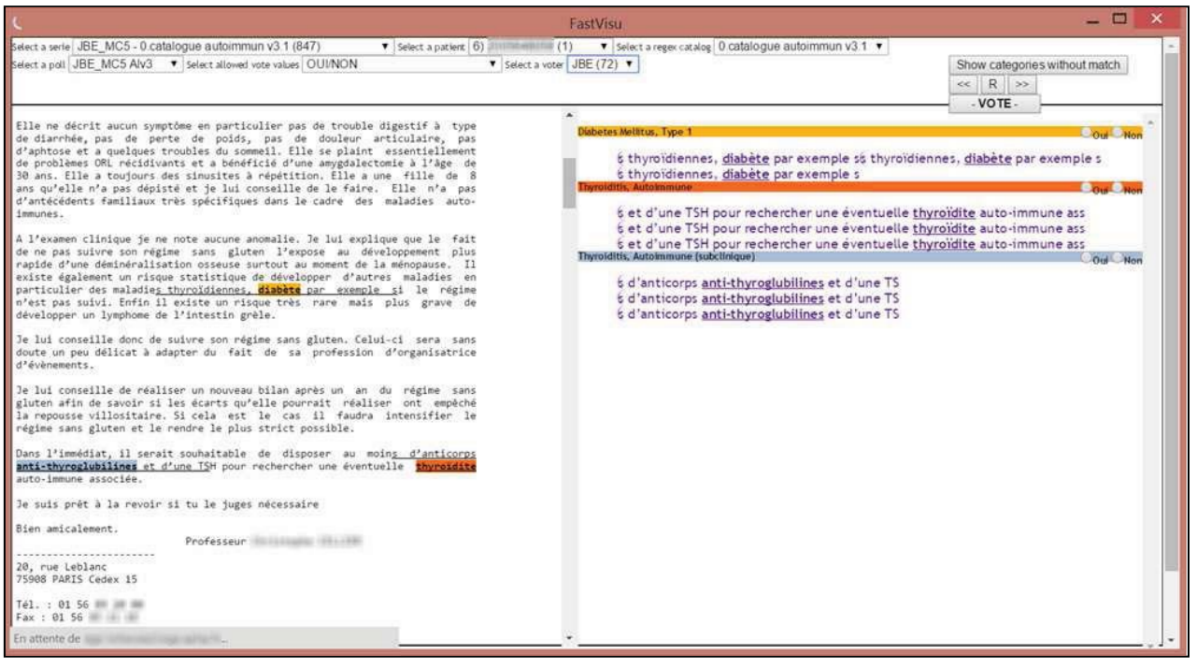


Figure 6.7: Interface de visualisation d'un dossier patient de FASTVISU d'après Escudie et al. [14].

Une évaluation a été réalisée pour mesurer le temps passé à revoir les dossiers. L'étude a été menée sur une cohorte de 741 patients atteints de maladie cœliaque. L'objectif était d'identifier dans cette cohorte les patients ayant une comorbidité auto-immune associée. Dans une première étape, le module de TAL a permis de filtrer les patients à revoir en identifiant les documents contenant la mention d'une maladie auto-immune associée. Au total, 276 patients ont été revus avec l'interface FASTVISU et l'identification d'une maladie auto-immune associée a pris seulement deux heures et demi. Le papier de 2015 indique que le système était dans une phase initiale de développement avec plusieurs cas d'usage en cours.

Docteur Warehouse

Situé à l'hôpital Necker, l'institut Imagine est spécialisé dans les maladies génétiques. Cet institut a développé en 2015 un EDS orienté documents appelé DrWarehouse [13]. Comme de nombreuses informations sur les maladies génétiques sont mentionnées dans les documents textuels, Dr Warehouse est un EDS spécialisé dans l'analyse du texte libre. L'importance du TAL se reflète dans son architecture : la table stockant les documents textuels (DWH_DOCUMENT) est au centre du modèle de données. Les données structurées sont liées à un document par une clef étrangère. Si aucun document n'est associé aux données structurées, un document est créé en

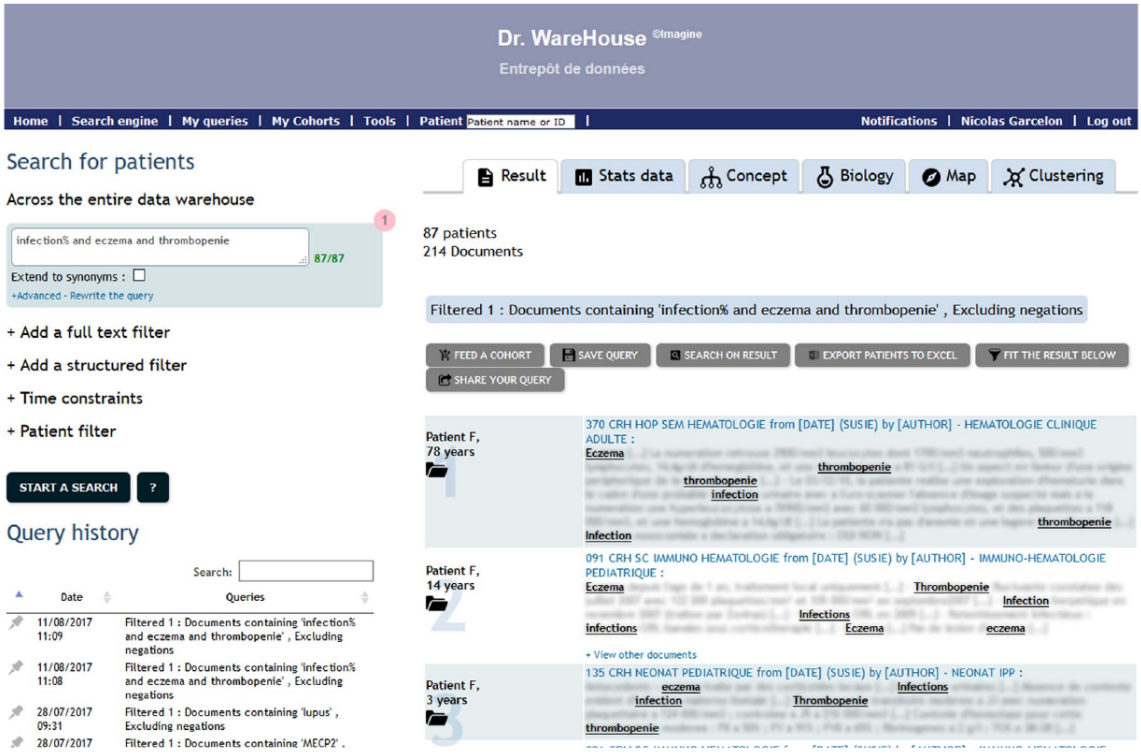


Figure 6.8: Interface de Dr Warehouse. L'utilisateur a recerché "infection% and eczema and thrombocytopenia". La requête retourne 87 patients et 214 documents. Les résultats sont affichés à droite, les termes trouvés apparaissent en sur-brillance. D'après Garcelon et al. [13]

concaténant les données structurées. Par exemple, les codes PMSI d'un séjour sont regroupés et stockés dans un document. Une étape de déidentification est réalisée en retirant noms, prénoms, date de naissance, numéros de téléphone et adresses. Les documents sont indexés avec les concepts et termes français de l'UMLS. Le contexte de chaque concept est aussi structuré : son appartenance au patient ou à un membre de sa famille, sa négation et son degré de certitude. Cette étape d'indexation des documents médicaux avec l'UMLS est l'étape la plus chronophage de la pipeline de traitement. Le moteur de recherche utilise les relations entre concepts de l'UMLS pour réaliser des recherches avancées. Dans l'interface (figure 6.8), le document est visualisé sans modification de sa forme (organisation des sections, paragraphes...).

Dr Warehouse fournit une visualisation centrée patient. L'interface comprend une ligne de vie, un moteur de recherche sémantique, un arbre généalogique, un parcours de soins, les concepts UMLS détectés dans le dossier, les cohortes dans lesquelles le patient a été inclus ou exclu et un module permettant de trouver des patients similaires.

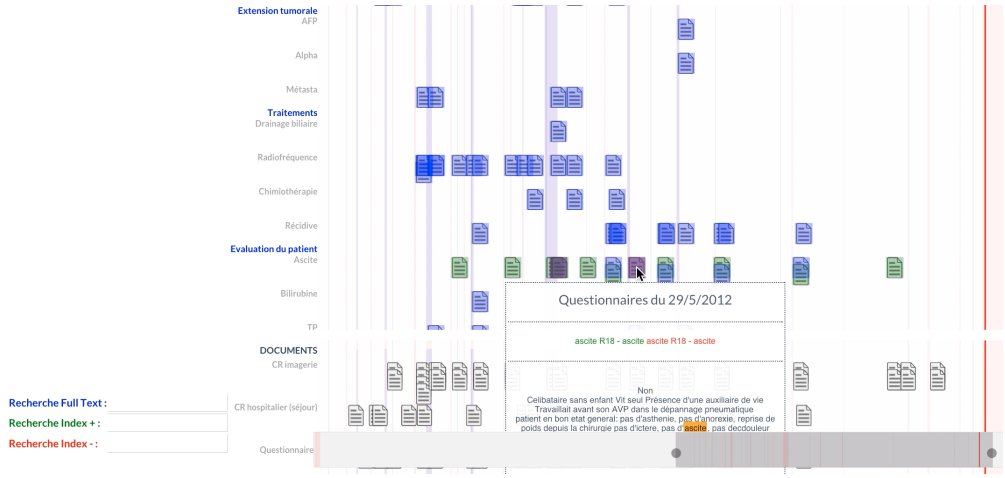


Figure 6.9: Interface du projet Ravel constituée d'une ligne de vie, d'un moteur de recherche et d'une visionneuse de documents.

Projet Ravel

Le projet Ravel est un projet ANR de 2011 réunissant deux industriels (Vidal et Medasys), deux hôpitaux (Rennes et Bordeaux) et quatre équipes de recherche françaises visait à développer un outil de visualisation d'un dossier patient [151, 152]. Le projet a aussi étudié les méthodes d'indexation pour requêter les données structurées et non structurées d'un dossier patient. En l'absence d'EDS à cette époque, une étape d'ETL (extraction, transformation et chargement des données) était nécessaire pour exploiter les données. La présence de Médasys, éditeur du DPI DxCare® utilisé à Rennes et Bordeaux, a permis de faciliter les extractions de données. Plusieurs terminologies médicales ont été utilisées pour l'annotation sémantique : MedDRA, CIM-10 et la SNOMED International notamment. Le contexte de chaque concept détecté comme la négation, la temporalité (évènement passé ou présent) et l'incertitude était aussi détecté. Une interface de visualisation comprenant une ligne de vie et un moteur de recherche a aussi été développé pour deux cas d'usage en cancérologie et en rhumatologie (figure 6.9). L'interface devait aussi afficher et résumer les informations cliniques les plus pertinentes.

VIP

VIP est une interface de visualisation de données centrée patient développée par IBM pour la prise en charge médicale en consultation externe et pour les réunions de concertation pluridisciplinaire en cancérologie [153]. L'interface est présentée figure 6.10. La ligne de vie affiche

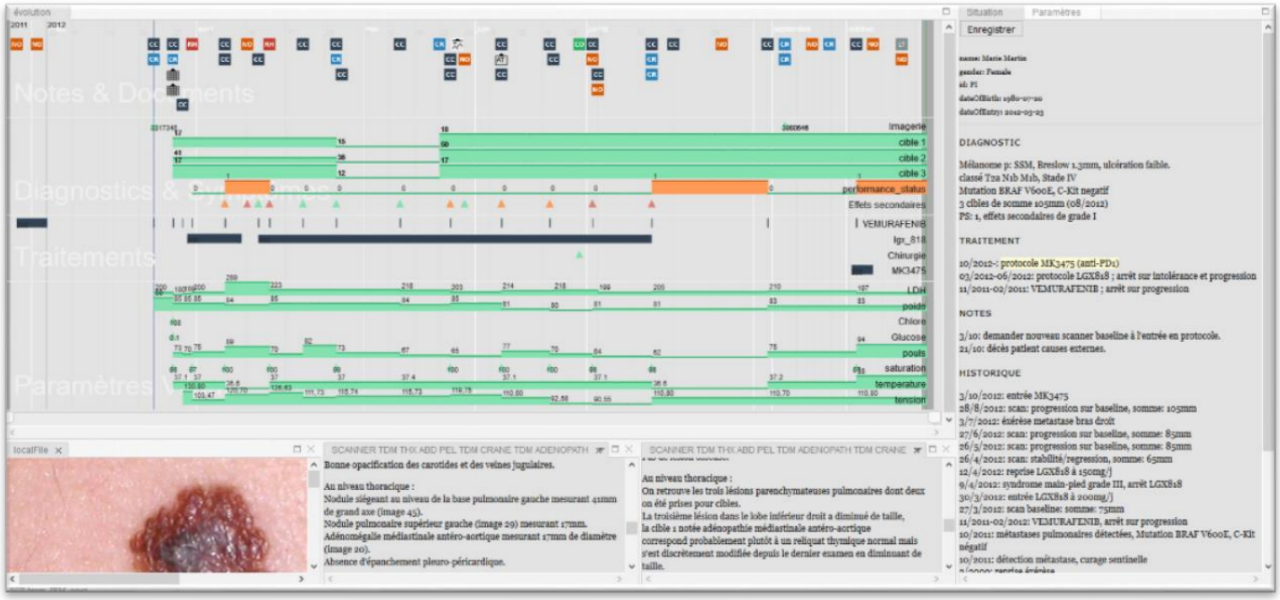


Figure 6.10: Interface de visualisation VIP, d'après Baudel et al.[153]

4 catégories : les documents textuels, les symptômes et les diagnostics, les traitements et les signaux vitaux. La ligne de vie affiche une vue complète des données du patient. Différentes icônes permettent de différencier les différents types de documents. Les évènements ponctuels comme les traitements sont représentés par un trait fin, les évènements se déroulant sur un intervalle de temps apparaissent sous forme de rectangle. Il est possible de zoomer et de naviguer sur la ligne de vie. En cliquant sur un item, le document complet apparaît. Il est aussi possible de visualiser l'historique des sections d'un document pour éviter de parcourir le même type de document un par un.

L'article ne mentionne pas la présence d'un moteur de recherche. Les auteurs expriment la difficulté de le déployer sur des données réelles pour des raisons de confidentialité.

6.3 Méthodes

Le maquettage de l'application a été réalisé dans le cadre d'un stage ingénieur[1] dans l'unité IAM, service d'information du CHU de Bordeaux, du 6 mai au 20 août 2019. Cet ingénieur a travaillé avec un médecin clinicien[2] pour identifier les besoins de visualisation et rédiger des spécifications fonctionnelles. Les logiciels Axure™ et Adobe XD™ ont été utilisés pour le

[1] Arnaud Godart, étudiant à l'école nationale supérieure de cognitique de Bordeaux

[2] Mathieu Lambert, PH dans le service de Médecine Interne et de Post-Urgence

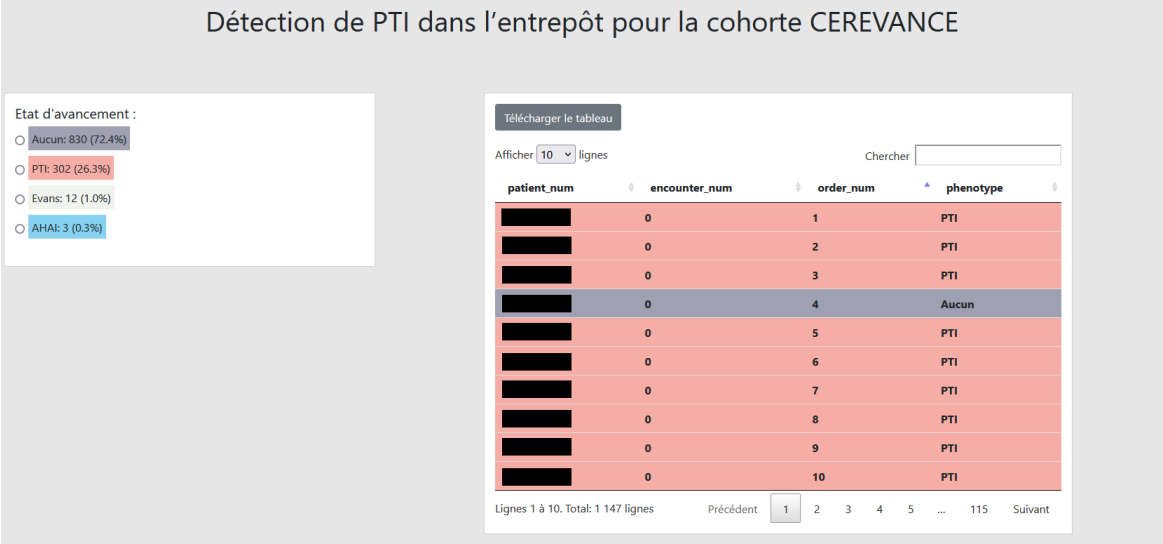


Figure 6.11: L'onglet avancement permet de visualiser une cohorte de patients. Dans cet exemple, 1 147 dossiers ont été revus par un ingénieur de recherche pour être inclus ou non dans une cohorte de maladie rare. Chaque dossier est classifié dans l'une des quatre catégories suivantes : "Aucun","PTI", "Evans" ou "AHAI". L'order_num correspond au numéro du patient de l'étude. L'encounter_num à 0 signifie que le dossier n'est pas filtré sur un séjour particulier.

maquettage. Un état de l'art de la visualisation d'un DPI a été réalisé par une revue de la littérature. Dans un procédé itératif, un total de sept maquettes a été réalisé pour atteindre un résultat satisfaisant. La maquette finale contenait une page d'accueil et trois modules : un module de ligne de vie pour visualiser l'ensemble des documents, un moteur de recherche et un module visionneuse pour afficher un document précis. L'interface SmartCRF, présentée ci-dessous, a été développée suite à ce travail.

6.3.1 Interface

L'interface web contient un onglet "projets" pour sélectionner un projet de recherche, un onglet "avancement" pour visualiser la liste des DPI d'une cohorte (figure 6.11) et un onglet "annotation" pour visualiser un DPI spécifique. Lors de la création d'un projet de recherche dans SmartCRF une liste de patients et de séjours est rattachée à celui-ci. Cette liste peut être issue du résultat d'une requête i2b2 ou fournie par l'investigateur. L'utilisateur autorisé se connecte au portail de l'EDS puis à l'application SmartCRF pour consulter les dossiers de sa cohorte. Chaque DPI peut être classifié par l'utilisateur (ex: patient éligible ou patient non éligible), les classes possibles sont paramétrées au début du projet.

La figure 6.12 montre l'interface développée pour visualiser un DPI.

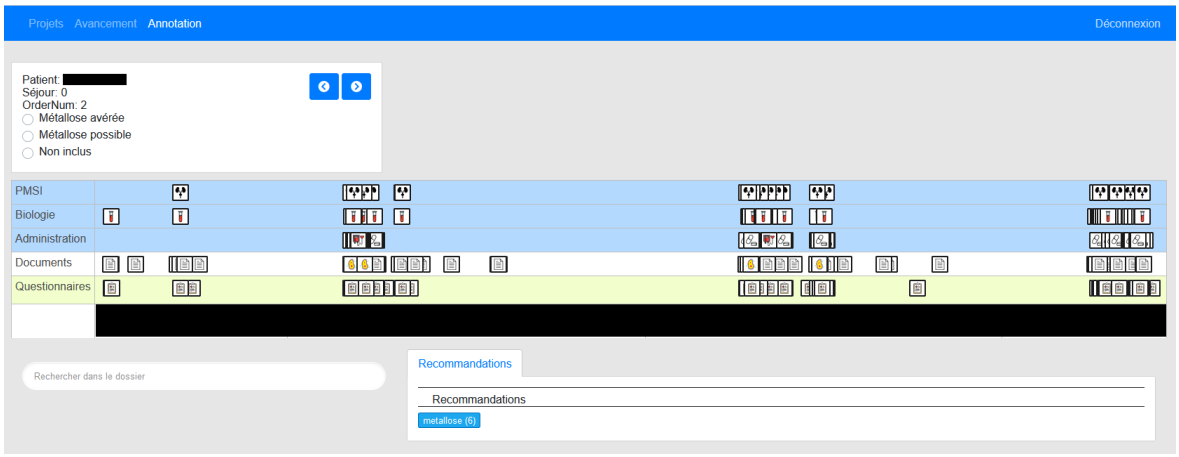


Figure 6.12: L'onglet annotation permet de visualiser un DPI de la cohorte. Dans cet exemple, il s'agit du 2ème patient (l'order_num est 2). Le patient peut être classifié dans trois catégories : métallose avérée, métallose possible ou non inclus. La ligne de vie affiche cinq catégories, chaque icône est cliquable pour visualiser le contenu. L'icône du PMSI représente un rein indiquant un code de diagnostic principal appartenant au chapitre XIV de la CIM-10 (maladies de l'appareil génito-urinaire). L'icône d'une poche de sang dans le groupe des administrations indique la réalisation d'une transfusion. L'icône jaune sur la ligne des documents indique un compte-rendu de bactériologie. Le système de recommandation, pré-paramétré pour cette étude, indique que le terme "metallose" apparaît six fois dans le dossier. En cliquant sur ce terme, la recherche est lancée par le moteur de recherche sur la gauche.

Ligne de vie

La libraire vis.js[3] a été utilisée pour réaliser la ligne de vie. La ligne de vie présente des similarités avec celle de VIP (figure 6.10) et LifeLines (figure 6.1). Elle permet d'afficher l'ensemble des éléments d'un dossier patient, de zoomer et de se déplacer sur l'axe horizontal. La ligne de vie contient cinq catégories correspondant à une source de données : PMSI, biologie, administration médicamenteuse (dont les transfusions), documents et formulaires DxCare®. Contrairement à LifeLines, ces catégories ne sont pas dépliables pour afficher des sous-catégories. Chaque élément de la ligne de vie, appelé item, correspond à un ou plusieurs éléments agrégés d'une source de données. Le grand volume de données de certains DPI a conduit à une ligne de vie surchargée, à un ralentissement de l'interface et à des chevauchements d'items. Pour résoudre ce problème, les données d'une seule journée issues du PMSI, de la biologie et des administrations médicamenteuses ont été agrégées dans un item unique. Certaines données issues de documents et des formulaires ont été arbitrairement espacées pour éviter des chevauchements d'items.

[3] `https://github.com/visjs/vis-timeline`

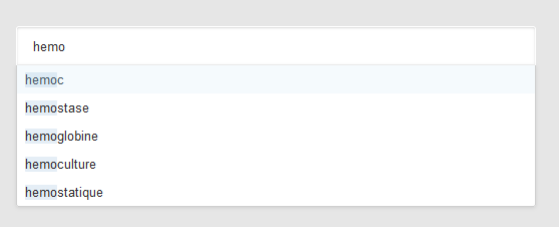


Figure 6.13: Système d'autocomplétion du moteur de recherche. Lorsque l'utilisateur saisit les deux premières lettres, des termes présents dans le dossier sont proposés.

Certains items ont une image pour représenter l'information qu'ils contiennent. Pour le PMSI, l'image d'un item représente le chapitre CIM-10 du diagnostic principal. Ces images permettent d'identifier rapidement les principaux diagnostics des hospitalisations (cardiologie, dermatologie, maladie infectieuse ...). Sur la ligne des administrations, une icône spéciale indique une transfusion. Sur la ligne des documents, les icônes permettent de distinguer les principaux documents : compte-rendu d'hospitalisation, de consultation, opératoire et d'imagerie.

Moteur de recherche

Le moteur de recherche permet de trouver un terme mentionné dans toutes les sources de données. Un système d'autocomplétion (figure 6.13) propose des termes présents dans le dossier lorsque l'utilisateur débute la saisie.

L'absence de suggestion d'un terme indique que celui-ci n'est pas présent dans le dossier. Lorsque l'utilisateur sélectionne un terme à rechercher dans le dossier, deux actions se produisent. Premièrement, les items de la ligne de vie contenant le terme recherché changent de couleur. La ligne de vie permet de visualiser quand le terme recherché est mentionné pour la première fois et quels séjours sont concernés. En cliquant sur un item de la ligne de vie, le document apparaît dans la visionneuse de document avec les mots recherchés en sur-brillance. Deuxièmement, les phrases de documents ou les libellés d'une terminologie contenant le terme recherché apparaissent sous le moteur de recherche. Comme dans EMERSE, ces résultats de recherche sont appelés snippet. En cliquant sur un snippet, le document apparaît dans la visionneuse avec les mots recherchés en sur-brillance. Lorsque le snippet correspond à une donnée structurée, l'historique des valeurs est affichée sous forme de tableau 6.15. Un snippet contient le plus souvent assez d'information pour comprendre le contexte du terme recherché ce qui évite d'ouvrir le document. Ces petites portions de texte sont affichées par ordre chronologique, du

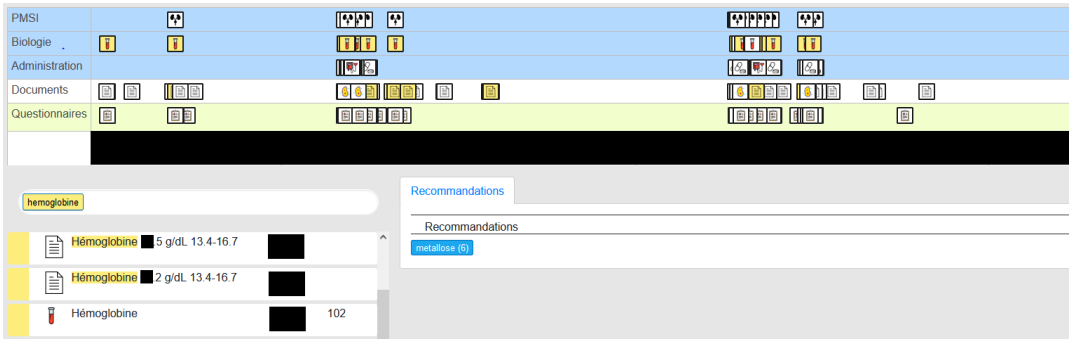


Figure 6.14: Résultats d'une recherche dans le dossier. Les items de la ligne de vie contenant le mot recherché changent de couleur. Sous le moteur de recherche sont affichés les snippets, petites portions de texte contenant le mot recherché. A gauche de chaque snippet, une icône affiche la source de données. L'icône d'un tube biologique indique que le terme "hémoglobine" est le libellé d'un code de biologie. Le nombre de 102 indique le nombre de résultats biologiques pour ce code. L'utilisateur accède aux détails en cliquant sur le résultat d'une recherche ou sur une icône de la ligne de vie.

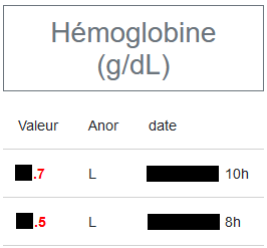


Figure 6.15: La visionneuse affiche l'historique des résultats biologiques sous forme de tableau ordonné par ordre chronologique. La lettre L signifie "low", c'est-à-dire que la valeur est en dessous de la limite basse de l'intervalle de valeurs normales. L'utilisateur peut aussi choisir de visualiser les résultats sous forme de graphique (non présenté).

plus récent au plus ancien.

Visionneuse de documents

La visionneuse de documents s'ouvre lorsque l'utilisateur clique sur un item de la ligne ou sur un snippet. La visionneuse positionne l'affichage sur le premier terme recherché trouvé (figure 6.16). Les termes recherchés dans le moteur de recherche apparaissent en sur-brillance avec des couleurs différentes dans le document affiché. Contrairement à DrWarehouse, le document original n'est pas affiché mais uniquement le texte extrait de ce dernier. La structure du document n'est donc pas conservé dans la visionneuse.

Figure 6.16: La visionneuse affiche un formulaire DxCare® contenant le mot recherché (VUP). L'utilisateur a annoté le texte "Diagnostic antenatal de VUP" pour signaler, où dans le dossier, l'information a été trouvée.

Système de recommandation

Le système de recommandation consiste à afficher la présence ou l'absence de termes pré-sélectionnés. Ces termes correspondent au *search term bundle* d'EMERSE. En cliquant sur un terme recommandé, le terme est placé dans le moteur de recherche et la recherche est réalisée. Le système de recommandation permet à un chercheur de visualiser rapidement la présence ou l'absence de termes d'intérêt pour son étude clinique. Il évite de réaliser une recherche manuelle via le moteur de recherche et permet un gain de temps en visualisant les termes présents et absents dans le DPI.

Système d'annotation

L'annotation consiste à sélectionner une portion d'un document et à lui assigner un libellé. Pour réaliser une annotation, l'utilisateur doit sélectionner une portion de texte et choisir un libellé d'annotation parmi une liste finie pré-établie au début du projet. Pour l'utilisateur, une annotation permet de tracer où l'information a été trouvée dans le dossier (figure 6.16). Cette action est similaire à celle de stabiloter un document papier pour mettre en avant des mots importants du dossier. En revenant au dossier, l'utilisateur peut visualiser les annotations réalisées ce qui lui permet de comprendre la classification d'un dossier. Une annotation génère une donnée structurée en base de données contenant des informations sur le dossier annoté, l'annotateur et l'annotation elle-même (date, texte, localisation dans l'EDS...). Dans certains cas usages, les annotations ont permis de générer des exemples d'entraînement pour développer des algo-

rithmes de classification automatique des dossiers ou d'extraction automatique d'information. Une annotation peut aussi être générée automatiquement par un algorithme pour pré-annoter des dossiers. Cette fonctionnalité est peu utilisée en pratique.

Initialement, l'annotation d'un terme était ajoutée automatiquement au système de recommandation mais cette fonctionnalité a été abandonnée car de nombreuses annotations conduisaient à surcharger le système de recommandation.

6.3.2 Architecture

L'architecture de l'application est présentée figure 6.17. Le serveur NodeJS est l'unique point d'entrée pour l'ensemble des requêtes utilisateur. Il assure la sécurité comme l'authentification des utilisateurs et est le seul à communiquer avec les autres services qui ne sont pas accessibles.

API données

L'API des données est un service chargé de récupérer les données, les indexer et les mettre en cache. Il interroge directement la base de données Oracle pour récupérer toutes les données d'un patient ou d'un séjour présentes dans i2b2. L'API récupère aussi des informations sur l'identité du patient pour réaliser une pseudonymisation par dictionnaire : nom, prénom, date de naissance et identifiants sont retirés automatiquement lorsqu'ils sont détectés dans un document textuel. Les métadonnées (libellés des examens biologiques, des médicaments etc…), stockées dans une base de données graphe, sont placées en cache et rattachées aux données cliniques pour fournir les libellés des examens biologiques, des codes de médicaments et PMSI etc.… L'API réalise toutes les transformations nécessaires pour fournir à l'interface les données nécessaires à l'affichage. Un module de cette API réalise en parallèle une indexation dans une base de données ElasticSearch. Les documents sont d'abord découpés en phrases par la libraire OpenNLP de Standford adaptée pour le français puis IAMsystem réalise une annotation sémantique.

ElasticSearch

Cette base de données orientée documents sert de cache pour stocker les données d'un DPI. Elle indexe les données textuelles, découpées par phrase, de chaque item de la ligne de vie et les termes détectés associés. ElasticSearch est utilisée pour l'autocomplétion et pour fournir les résultats d'un terme recherché (snippets).

Annotation sémantique

L'objectif de l'annotation sémantique est de détecter les termes à recommander et de fournir une liste de termes pour l'autocomplétion du moteur de recherche. Un dictionnaire a été spécialement créé pour cette tâche. Il combine une liste de syntagmes nominaux extraits des documents textuels de l'entrepôt (chapitre 5), des termes issus de différentes terminologies médicales et les termes à recommander stockés en base de données. Ce service est utilisé au moment de l'indexation des documents et au moment où un document est visionné. Lorsque l'utilisateur clique sur un document, l'API d'annotation est appelée pour annoter dynamiquement le document avec les termes recherchés. Cette détection dynamique évite d'indexer toutes les informations relatives à un terme détecté dans ElasticSearch. Les abréviations et les variations lexicales d'un terme sont prises en compte dynamiquement par IAMsystem, les fonctionnalités de recherche en texte libre d'ElasticSearch ne sont pas utilisées pour identifier un terme dans un document.

MariaDB

MariaDB est une base de données relationnelle open source. Elle stocke toutes les informations relatives à un projet de recherche : les utilisateurs rattachés à un projet, la liste des numéros de patients ou de séjours d'un projet, les classifications de chaque dossier, les termes à afficher par le système de recommandation et les annotations réalisées. Cette base de données ne contient aucune donnée textuelle sauf les annotations réalisées dans un document.

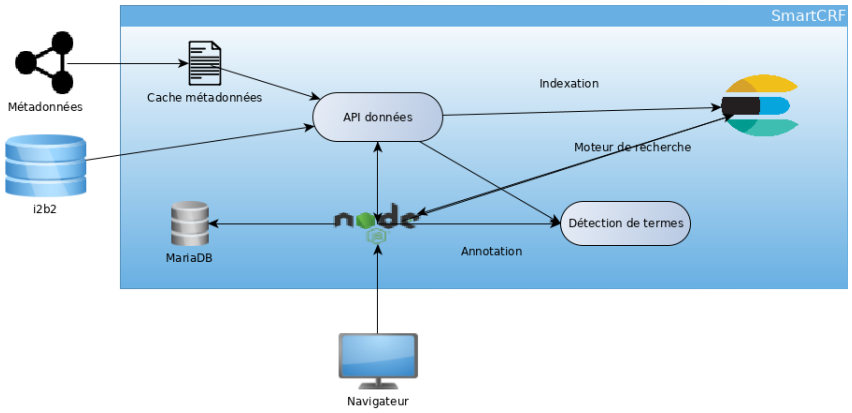


Figure 6.17: Architecture de SmartCRF

6.3.3 Evaluation

Une évaluation a été réalisée dans le cadre du projet "Appel à Manifestation d'Intérêt en Intelligence Artificielle" (AMI-IA) de la direction interministérielle du numérique (DINUM) et la direction interministérielle de la transformation publique (DITP). Ce projet visait à développer un moteur de recherche dans un DPI pour le soin et les activités d'utilisation secondaire. Cette étude avait pour but d'évaluer la recherche d'information en pratique clinique et en recherche clinique. L'évaluation a consisté à trouver sept informations fréquemment recherchées dans un dossier : antécédent d'hypertension artérielle, statut tabagique fumeur ou non, antécédent de transfusion, antécédent de diabète, antécédent d'antibiothérapie, antécédent d'allergie médicamenteuse et antécédent de dépression. Vingt-trois dossiers patients ont été sélectionnés pour comparer deux interfaces : DxCare® utilisée en routine et la nouvelle interface SmartCRF. Les réponses attendues étaient fermées (oui/non) pour faciliter l'analyse des résultats. Les 23 dossiers ont été revus par deux médecins qui ont alternativement utilisé le logiciel métier DxCare® et l'interface SmartCRF.

Pour l'évaluation en pratique clinique de DxCare®, la consigne était de rechercher une information en simulant une utilisation réelle. Les cliniciens lisent rarement l'intégralité du dossier pour trouver une information et utilisent une stratégie de recherche consistant à sélectionner les documents où l'information pourraient apparaître. Par exemple, les antécédents du patient sont souvent mentionnés dans un questionnaire DxCare® et dans les comptes-rendus d'hospitalisation. Si l'information n'était pas mentionnée dans les documents choisis,

le médecin pouvait considérer que l'information était absente et choisissait de répondre "non". Chaque médecin était libre d'utiliser sa stratégie de recherche habituelle.

Pour l'évaluation en recherche clinique, une ingénieure de recherche formée au recueil de données dans DxCare® a réalisé le même exercice uniquement avec DxCare®. Pour trouver les réponses aux questions, l'intégralité du dossier pouvait être lue. Les réponses fournies par l'ingénieure de recherche étaient considérées comme les bonnes réponses pour l'évaluation. Les réponses aux questions ont été saisies dans le logiciel REDCap, un eCRF.

6.4 Résultats

En trois ans, l'interface a été utilisée par une cinquantaine d'utilisateurs et dans plus de 100 projets de recherche. L'interface a été utilisée pour mener des études de faisabilité et des études rétrospectives, mais également pour constituer une cohorte de maladies rares et pour des activités de vigilance. L'outil a été adopté dans la routine de certaines activités comme l'addictovigilance pour confirmer plus rapidement les signaux détectés et dans la surveillance des infections du site opératoire où l'interface est utilisée pour confirmer la présence d'une infection et identifier la porte d'entrée afin de compléter les informations du formulaire de surveillance nationale. Dans une étude rétrospective, les ingénieurs de recherche ont choisi de combiner l'utilisation de DxCare® et de SmartCRF pour le remplissage d'eCRF. Le logiciel DxCare® restait la source d'information de référence pour remplir l'eCRF mais SmartCRF était utilisée pour localiser rapidement une information dans le dossier et vérifier qu'une information n'avait pas été manquée.

Une étude a classifié les comptes rendus d'imagerie de patients suspects d'une atteinte pulmonaire au Sars-Cov2 en trois catégories : atteinte typique, atteinte compatible ou absence d'atteinte pulmonaire. Un ETL a été réalisé pour alimenter l'EDS avec les classifications réalisées par cette étude et d'autres études ont ré-utilisées ces classifications. Ce résultat est une preuve de concept d'*entrepôt apprenant* consistant à alimenter l'EDS de données issues de l'interface SmartCRF.

6.4.1 Etude d'évaluation

En simulation d'une pratique clinique, le temps moyen pour collecter les informations était de 228 secondes (3 minutes 48 secondes) en moyenne avec DxCare® versus 155 secondes (2 minutes 35 secondes) avec SmartCRF. En pratique de recherche clinique, l'ingénieure de recherche a mis en moyenne 458 secondes (7 minutes et 38 secondes) par dossier. Sur les 161 réponses (23 patients multipliés par 7 questions), l'utilisation de DxCare® en pratique clinique a permis de trouver 135 bonnes réponses (84%) tandis que 159 bonnes réponses ont été trouvées avec SmartCRF (98,7%). Deux réponses contradictoires où une information positive était trouvée par SmartCRF tandis que l'ingénieure de recherche clinique avait renseigné une réponse négative ont été revues. Ces erreurs correspondaient à des informations contradictoires dans DxCare® : les documents récents mentionnaient une absence d'HTA et d'allergie médicamenteuse tandis qu'une information positive était mentionnée dans des documents anciens.

Cette étude a permis d'identifier des informations difficiles à trouver en pratique clinique où le temps de recherche d'information est limité. Les comorbidités (hta, diabète et dépression) étaient faciles à identifier dans DxCare®. Les antécédents de transfusion ont été difficiles à trouver : dans 40% des cas l'information n'a pas été trouvée alors que l'information était présente dans le dossier. Cette étude montre que SmartCRF permet de gagner, en moyenne, 77% de temps pour trouver les informations dans un dossier.

6.5 Discussion

L'interface SmartCRF a été développée pour faciliter la revue des dossiers dans le but d'identifier des patients éligibles à une étude et pour trouver plus rapidement des informations dans un DPI. Les retours positifs des utilisateurs et son utilisation au quotidien démontrent l'intérêt d'une telle interface. L'interface est composée d'un assemblage de modules déjà décrits dans la littérature. La ligne de vie est similaire à celle proposée par LifeLines en 1996 [140], le moteur de recherche et le système de recommandation possèdent des fonctionnalités similaires à celui de EMERSE [15], la visionneuse de documents mettant en sur-brillance les termes détectés ressemble à celle utilisée par DrWarehouse [13] et FASTVISU [14]. Ces outils similaires

présentent cependant des limites pour visualiser les données d'un DPI du CHU de Bordeaux. Le logiciel EMERSE se limite aux documents textuels, il ne permet pas d'intégrer et de visualiser des données structurées comme SmartCRF. De même FASTVISU ne décrit pas comment les données structurées pourraient être intégrées. Sylvestre et al. ont montré la nécessité de combiner à la fois les notes cliniques et les données structurées pour identifier les critères d'inclusion d'une étude [154]. DrWarehouse traite les données structurées comme des données textuelles ce qui limite les possibilités de visualisation de ces données. A contrario, ces autres interfaces implémentent des fonctionnalités et visualisation qui sont manquantes dans SmartCRF.

L'absence d'une interface unique pour la visualisation d'un DPI peut s'expliquer par les difficultés à prendre en compte l'hétérogénéité des différents environnements et les besoins spécifiques de certains établissements. La revue de Rind et al. [2], présentée plus haut, a montré l'hétérogénéité des interfaces et des finalités. De même, SmartCRF est particulièrement adapté aux spécificités des données du CHU de Bordeaux mais serait difficilement déployable et adaptable à un autre établissement. Le format des données reçu par l'interface de SmartCRF a été conçu ad-hoc pour être consommé directement par le navigateur sans nécessiter de transformation supplémentaire. Ce format est optimal pour les performances car il minimise le nombre d'opérations à réaliser avant affichage dans le navigateur mais n'est pas standard pour faciliter son installation dans un autre établissement.

SmartCRF présente quelques spécificités qui le démarquent des autres outils. Premièrement, l'interface ne dépend pas d'une solution particulière d'EDS. L'interface de DrWarehouse dépend d'une solution d'EDS spécifique qui n'est pas utilisée au CHU de Bordeaux et EMERSE indexe au préalable tous les documents d'un serveur de documents. Dans SmartCRF les données d'un DPI sont chargées et indexées à la volée dans une base de données ElasticSearch temporaire. Cette base de données est utilisée comme cache et vidée tous les jours pour prendre en compte le rafraîchissement quotidien des données de l'EDS du CHU de Bordeaux. L'application SmartCRF consomme peu de ressources car le nombre de patients indexés chaque jour est très petit comparé au nombre de patients présents dans l'EDS. Le chargement et l'indexation à la volée est rendue possible par l'efficace indexation des données dans i2b2, par la rapidité d'IAMsystem à annoter les documents et à ElasticSearch pour les indexer. Le temps de chargement d'un dossier en quelques secondes est nécessaire et satisfaisant pour les utilisateurs.

Certains dossiers très volumineux nécessitent toutefois plus de 10 secondes de chargement mais ces derniers sont rares.

Une autre particularité de SmartCRF est l'utilisation de syntagmes nominaux pour réaliser l'autocomplétion de la barre de recherche. Une étude d'EMERSE a montré que seulement 63,6% des termes recherchés étaient présents dans l'UMLS contenant environ 800 000 termes en anglais au moment de l'étude [148]. Comme le nombre de termes est beaucoup plus faible en français, l'UMLS serait donc une solution insuffisante pour l'autocomplétion. Les syntagmes nominaux sont des termes importants d'un document [125] et permettent donc d'offrir une bonne couverture des termes recherchés par les utilisateurs.

Cependant, l'application possède quelques limites. Comme les développeurs de DrWarehouse l'ont souligné, afficher le document original plutôt que le texte extrait facilite sa lecture et sa compréhension par l'utilisateur. Afficher le document original pose aujourd'hui un problème technique car ces données très volumineuses ne sont pas répliquées dans i2b2 qui conserve uniquement le texte extrait. Plus un document contient de mise en forme comme des en-têtes ou des tableaux, plus le document est difficile à comprendre dans la visionneuse.

Une autre limite est l'exhaustivité des données. Les processus d'ETL de l'EDS sélectionnent et filtrent les sources de données pertinentes à intégrer. Des erreurs peuvent être présentes dans l'ETL et des informations peuvent être manquantes dans l'interface. DxCare® contient l'exhaustivité des données patient mais son interface ne facilite pas la recherche d'information. Comme le temps imparti à la recherche d'information en pratique clinique et en recherche clinique est limité, l'intégralité du dossier ne peut pas toujours être lue faute de temps. SmartCRF permet de fouiller l'intégralité du dossier via son moteur de recherche mais certaines données peuvent être manquantes. Le bénéfice / risque de chaque outil doit être connu des utilisateurs avant utilisation. L'étude d'évaluation a montré que les informations étaient correctement identifiées par SmartCRF malgré des données possiblement manquantes.

En perspective, le moteur de recherche pourrait s'appuyer sur des terminologies pour offrir des recommandations. Dans DrWarehouse et dans le projet RAVEL, des terminologies médicales ont été utilisées pour réaliser une expansion sémantique d'un terme recherché, c'est-à-dire recommander des concepts liés par une relation dans une terminologie. La détection du contexte comme la négation ou la temporalité d'un terme détecté sont aussi des fonctionnalités absentes

de l'interface actuelle. Ces fonctionnalités sont présentes dans DrWarehouse et dans le projet RAVEL mais n'ont pas été implémentées dans EMERSE. D'après les auteurs d'EMERSE, le succès de leur système repose sur une interface simple avec peu de fonctionnalités. L'ajout d'une fonctionnalité nécessite de réaliser une évaluation spécifique pour démontrer son intérêt.

Les interfaces développées pour l'utilisation secondaire de données n'abordent pas les défis de réutilisation des données produites par les ingénieurs de recherche. La mise en qualité des données et l'annotation des dossiers est une activité très chronophage et coûteuse. Une interface comme SmartCRF permet de capturer et de réutiliser ces informations précieuses. SmartCRF est une preuve de concept d'*entrepôt apprenant* où les résultats d'une étude ont été chargés dans l'EDS puis utilisés par d'autres études. Ces données pourraient aussi permettre de développer des algorithmes de classification automatique pouvant aider les ingénieurs à revoir plus rapidement les dossiers. Les données produites en routine par les activités de recherche clinique et de vigilance ouvrent la voie à d'autres travaux de recherche pour les exploiter.

6.6 Conclusion

SmartCRF est une interface conçue pour revoir rapidement un DPI. Elle pallie les défauts des algorithmes qui n'obtiennent pas de performance satisfaisante pour classifier un dossier ou extraire une information lorsque la tâche est trop complexe. Les interactions homme-machine permettent à un expert humain de gagner du temps et les données générées par ces interactions peuvent servir à d'autres études et au développement d'algorithmes d'extraction d'information. Cette interface est une brique d'un *entrepôt apprenant* où chaque interaction avec un utilisateur est susceptible de structurer une information importante pour la recherche clinique.

Chapitre 7

Conclusion

Ce chapitre est une conclusion générale à la thèse. Il présente les principales contributions et les perspectives futures. Ce travail de recherche appliquée a été réalisé au sein de l'unité IAM en charge du déploiement de l'entrepôt de données (EDS) du CHU de Bordeaux. Il visait à trouver des solutions à des problèmes concrets d'extraction et de structuration d'information en prenant en compte les spécificités des données hospitalières et les contraintes des utilisateurs. La section suivante résume les contributions de cette thèse et leurs spécificités aux données hospitalières. La dernière section présente les perspectives de recherche faisant suite à ce travail.

7.1 Principales contributions

Le chapitre 2 a présenté une stratégie de record linkage adaptée aux données hospitalières. Nous avons montré que l'utilisation d'un moteur de recherche permet de diminuer le nombre de comparaisons à réaliser, étape indispensable pour un hôpital, sans perte importante de rappel. Aussi, les décès intra-hospitaliers permettent la création automatique d'un gold standard pour recourir à des méthodes d'apprentissage supervisé. Bien que le statut vital soit une information cruciale pour les études de faisabilité et les études observationnelles, seulement un quart des décès était enregistré dans notre système d'information hospitalier. Lorsque la probabilité d'appariements est au-dessus du seuil supérieur, la précision est suffisante pour réaliser des liens automatiques tandis qu'entre les deux seuils une revue manuelle doit être réalisée pour ne pas introduire d'erreur dans une étude.

Le chapitre 3 a montré qu'une approche par dictionnaire offrait des résultats satisfaisants pour détecter et identifier les médicaments mentionnés dans les formulaires DxCare®. Comme les professionnels de santé mentionnent des médicaments commercialisés en France, un dictionnaire suffisamment exhaustif comme Romedi est suffisant pour les détecter et les identifier. Son graphe de connaissance intègre les médicaments issus de données structurées (référentiel interne) et non structurées (documents textuels) afin de faciliter leur recherche dans un EDS. Ce graphe de connaissance permet de regrouper les médicaments détectés selon leur classification ATC pour rechercher, par exemple, les patients prenant des antidiabétiques oraux. L'extraction d'information sur les médicaments est importante pour la réalisation d'études pharmacologiques. C'est par exemple la première étape pour détecter les interactions médicamenteuses sur les ordonnances de sortie dans le cadre du projet PREPS PROSIT.

Le chapitre 4 présente IAMsystem, un algorithme d'annotation sémantique pour rechercher rapidement des concepts médicaux dans les documents textuels. Cet annotateur tire parti de l'inventaire des abréviations médicales créé au chapitre 5. Comme discuté dans le chapitre introductif, la majorité des informations cliniques est présente dans le texte libre [4]. L'hétérogénéité des documents d'un EDS présentée au chapitre 1 nécessite l'utilisation d'algorithmes adaptés à la difficulté de la tâche d'extraction d'information. En règle générale, plus un document est long, plus les éléments contextuels sont nombreux, plus l'algorithme doit être complexe pour extraire une information. Pour structurer une information sur un médicament, il est parfois nécessaire de prédire si le médicament identifié appartient au patient ou non, si le médicament est mentionné pour discuter une éventualité thérapeutique ou pour informer que le patient est allergique à ce médicament. Dans d'autres documents textuels, le contexte est évident ; certaines réponses aux formulaires DxCare® décrivent les traitements habituels d'un patient (figure 1.1). De même, il existe peu d'ambiguïté dans les ordonnances de sortie : sauf exception, le médicament mentionné est pris par le patient. Les approches par dictionnaire fonctionnent bien dans ces dernières situations et une solution simple comme IAMsystem est suffisante pour la détection. Sa rapidité permet son intégration dans l'interface SmartCRF pour détecter, rechercher et indexer à la volée de nombreux concepts médicaux dans un DPI afin de proposer une autocomplétion pertinente et un système de recommandation.

Le chapitre 5 a présenté des algorithmes de détection et d'identification du sens des abrévi-

ations rencontrées dans les documents médicaux. La majorité des algorithmes de détection des abréviations sont basés sur la cooccurrence des formes courtes et longues, ils ne fonctionnent qu'avec des abréviations dites locales. Dans les DPI, les abréviations sont largement utilisées sans mention de leurs sens, les abréviations présentes dans les données hospitalières sont globales. Pour limiter l'espace de recherche, les abréviations ont été recherchées dans les syntagmes nominaux des documents extraits par des méthodes linguistiques. Ce chapitre a présenté deux algorithmes adaptés à un EDS, le premier basé sur la détection automatique des troncations, le deuxième sur l'hypothèse distributionnelle qui stipule que les mots qui apparaissent dans les mêmes contextes ont des significations similaires ou apparentées [126, 127]. L'inventaire des abréviations médicales a permis d'améliorer la détection des concepts médicaux lors de la recherche d'information et de structurer les informations sur les antécédents médicaux

Le chapitre 6 présente une interface de visualisation d'un dossier patient spécifique aux données hospitalières. Lorsque la tâche de structuration de l'information est trop complexe, elle est réalisée par un expert humain via cette interface dédiée. Le temps d'extraction de l'information par un ingénieur informatique de l'unité IAM, via un algorithme, ne doit pas être supérieur au temps d'extraction manuelle de l'information. Développer, déployer et maintenir un algorithme pour chaque besoin spécifique des utilisateurs n'est pas envisageable. Demander à des utilisateurs de produire des annotations pour entraîner des modèles à l'état de l'art peut s'avérer contre-productif car les bénéfices en termes de performance ne justifient pas le temps investi à annoter, entraîner et déployer un algorithme. Une interface comme SmartCRF combine la compréhension du langage naturel d'un humain et la puissance de calcul d'une machine pour structurer manuellement une information. Cette interface permet aux utilisateurs de gagner du temps mais aussi de capturer les informations structurées par leur revue manuelle des dossiers. L'idée d'un *entrepôt apprenant* est de profiter des interactions utilisateurs avec SmartCRF pour capturer ces annotations et les réutiliser. Une preuve de concept a été réalisée en intégrant dans l'EDS des données produites par une étude qui ont par la suite été réutilisées dans une autre étude.

7.2 Perspectives

Les méthodes présentées dans ce travail de thèse se limitent à des tâches d'extraction d'information très spécifiques avec un outil simple d'annotation sémantique, IAMsystem. Ce travail ne s'est pas attaqué à la structuration automatique d'informations complexes d'un DPI comme la prédiction de l'éligibilité d'un patient à une étude. La structuration d'informations complexes nécessite de rechercher et regrouper de multiples éléments dans plusieurs sources de données d'un DPI. La solution proposée pour structurer ces informations est l'interface SmartCRF qui permet de faciliter la recherche d'information dans un dossier.

Nous faisons l'hypothèse que la structuration d'information dans les situations simples est une étape importante pour structurer par la suite des informations plus complexes d'un DPI. Comme les informations saisies dans un DPI sont redondantes et dupliquées [155], structurer les informations dans les documents courts devrait faciliter leur structuration dans les documents longs. Par exemple, les comptes-rendus d'hospitalisation (CRH) rédigés à la fin d'hospitalisation reprennent différents éléments du dossier dont certains sont structurés comme les résultats d'examens biologiques. Les antécédents médicaux mentionnés dans les CRH sont souvent issus des formulaires DxCare (texte court). Plus les éléments contextuels d'un CRH sont structurés, plus la structuration d'information d'un CRH devrait être simplifiée. En plus de structurer une information, l'annotation sémantique permet de tirer parti du contenu d'un graphe de connaissance en navigant dans les relations du graphe ou en utilisant une représentation contextualisée d'un noeud. Une des perspectives de recherche est d'identifier les coréférences entre les informations d'un CRH et les éléments structurés contextuels afin de faciliter l'extraction d'information dans ces documents longs.

Une autre perspective de recherche est de favoriser la participation des utilisateurs de l'EDS dans l'annotation et la structuration des données. Les annotations d'un DPI sont nécessaires pour le développement et l'évaluation d'algorithmes mais chronophages et coûteuses à produire. Les projets de recherche menés avec un EDS sont l'occasion de capturer les informations extraites des DPI pour les besoins d'une étude. La participation des utilisateurs nécessite de leur apporter des bénéfices immédiats en termes de gain de temps ou d'aide à la structuration des

futurs dossiers. Des interactions plus complexes avec les utilisateurs où la machine sélectionnerait les exemples à annoter pourraient être proposées [156]. Même si l'interface SmartCRF est une preuve de concept d'*entrepôt apprenant* où les interactions avec les utilisateurs permettent d'enrichir un EDS, l'hétérogénéité sémantique des informations extraites des dossiers par les chercheurs pose des difficultés à réutiliser ces annotations. Des travaux de recherche pourront être menés pour essayer de faciliter leur intégration et leur réutilisation dans un EDS.

Publications

- Cossin S, Jouhet V, Mougin F, Diallo G and Thiessard F. IAM at CLEF eHealth 2018: Concept Annotation and Coding in French Death Certificates. CEUR-WS. September 2018.
- Cossin S, Lebrun L, Niamkey A, Mougin F, Lambert M, Diallo G, Thiessard F, Jouhet V. SmartCRF: a prototype to visualize, search and annotate an electronic health record from an i2b2 clinical data warehouse. MedInfo. Lyon. Août 2019.
- Cossin S, Loustau R, Jouhet V, Mougin F, Evrard G, Giljardine C, et al. ROMEDI, une terminologie médicale française pour la détection des médicaments en texte libre. PFIA; 2018.
- Cossin S, Lebrun L, Lobre G, Loustau R, Jouhet V, Griffier R, Mougin F, Diallo G, Thiessard F. Romedi: an open data source about French drugs on the semantic web. MedInfo. Lyon. Août 2019.
- Cossin S, Godart A, Lambert M, Lebrun L, Jouhet V. SmartEHR : un moteur de recherche dans un dossier patient informatisé. Revue d'Épidémiologie et de Santé Publique. 1 mars 2020;68:S52.
- Cossin S, Jouhet V. IAM at CLEF eHealth 2020: Concept Annotation in Spanish Electronic Health Records. CEUR workshop proceedings. 2020;
- Cossin S, Diouf S, Griffier R, Le Barrois d'Orgeval P, Diallo G, Jouhet V. Linkage of Hospital Records and Death Certificates by a Search Engine and Machine Learning. JAMIA Open. janv 2021;4(1):ooab005.

- Cossin S, Jolly M, Larrouture I, Griffier R, Jouhet V, Giljardine C, et al. Semi-Automatic Extraction of Abbreviations and their Senses from Electronic Health Records. PFIA; Bordeaux; 2021.
- Cossin S, Diallo G., Jouhet V. IAM at IberLEF 2022: NER of Species Mentions. CEUR workshop proceedings. sept 2022; (accepté)

Bibliographie

[1] R. GRIFFIER, "Secondary use of health data: Identification of locks and contributions from the Bordeaux University Hospital's data warehouse," Master's thesis, Sep. 2020.

[2] A. Rind, T. D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman, "Interactive Information Visualization to Explore and Query Electronic Health Records," *Foundations and Trends in Human-Computer Interaction*, vol. 5, no. 3, pp. 207–298, Feb. 2013.

[3] C. Safran, "Update on Data Reuse in Health Care," *Yearbook of Medical Informatics*, vol. 26, no. 1, pp. 24–27, Aug. 2017.

[4] S. M. Meystre, C. Lovis, T. Bürkle, G. Tognola, A. Budrionis, and C. U. Lehmann, "Clinical Data Reuse or Secondary Use: Current Status and Potential Future Progress," *Yearbook of Medical Informatics*, vol. 26, no. 1, pp. 38–52, Aug. 2017.

[5] C. Safran, M. Bloomrosen, W. E. Hammond, S. Labkoff, S. Markel-Fox, P. C. Tang, and D. E. Detmer, "Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper," *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, no. 1, pp. 1–9, 2007.

[6] B. Kankova and G. Duftschmid, "Reusing Data of an EU-Wide EHR System for Clinical Trials: A Capabilities Analysis," *Studies in Health Technology and Informatics*, vol. 271, pp. 17–22, Jun. 2020.

[7] P. Coorevits, M. Sundgren, G. O. Klein, A. Bahr, B. Claerhout, C. Daniel, M. Dugas, D. Dupont, A. Schmidt, P. Singleton, G. De Moor, and D. Kalra, "Electronic health records: New opportunities for clinical research," *Journal of Internal Medicine*, vol. 274, no. 6, pp. 547–560, Dec. 2013.

[8] A.-S. Jannot, E. Zapletal, P. Avillach, M.-F. Mamzer, A. Burgun, and P. Degoulet, "The Georges Pompidou University Hospital Clinical Data Warehouse: A 8-years follow-up experience," *International Journal of Medical Informatics*, vol. 102, pp. 21–28, Jun. 2017.

[9] S. N. Murphy, M. E. Mendis, D. A. Berkowitz, I. Kohane, and H. C. Chueh, "Integration of clinical and genetic data in the i2b2 architecture," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, p. 1040, 2006.

[10] S. Murphy and A. Wilcox, "Mission and Sustainability of Informatics for Integrating Biology and the Bedside (i2b2)," *EGEMS (Washington, DC)*, vol. 2, no. 2, p. 1074, 2014.

[11] N. Paris, M. Mendis, C. Daniel, S. Murphy, X. Tannier, and P. Zweigenbaum, "I2b2 implemented over SMART-on-FHIR," *AMIA Summits on Translational Science Proceedings*, vol. 2018, pp. 369–378, May 2018.

[12] T. Boudemaghe and I. Belhadj, "Data Resource Profile: The French National Uniform Hospital Discharge Data Set Database (PMSI)," *International Journal of Epidemiology*, vol. 46, no. 2, pp. 392–392d, Apr. 2017.

[13] N. Garcelon, A. Neuraz, R. Salomon, H. Faour, V. Benoit, A. Delapalme, A. Munnich, A. Burgun, and B. Rance, "A clinician friendly data warehouse oriented toward narrative reports: Dr. Warehouse," *Journal of Biomedical Informatics*, vol. 80, pp. 52–63, Apr. 2018.

[14] J.-B. Escudié, A.-S. Jannot, E. Zapletal, S. Cohen, G. Malamut, A. Burgun, and B. Rance, "Reviewing 741 patients records in two hours with FASTVISU," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2015, pp. 553–559, 2015.

[15] D. A. Hanauer, Q. Mei, J. Law, R. Khanna, and K. Zheng, "Supporting information retrieval from electronic health records: A report of University of Michigan's nine-year experience in developing and using the Electronic Medical Record Search Engine (EMERSE)," *Journal of Biomedical Informatics*, vol. 55, pp. 290–300, Jun. 2015.

[16] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, and H. Liu, "Clinical information extraction applications: A literature review," *Journal of Biomedical Informatics*, vol. 77, pp. 34–49, Jan. 2018.

[17] J. D. Curb, C. E. Ford, S. Pressel, M. Palmer, C. Babcock, and C. M. Hawkins, "Ascertainment of vital status through the National Death Index and the Social Security Administration," *American Journal of Epidemiology*, vol. 121, no. 5, pp. 754–766, May 1985.

[18] C. Safran, "Reuse of clinical data," *Yearbook of Medical Informatics*, vol. 9, pp. 52–54, Aug. 2014.

[19] B. Jones and D. K. Vawdrey, "Measuring Mortality Information in Clinical Data Warehouses," *AMIA Summits on Translational Science Proceedings*, vol. 2015, pp. 450–455, Mar. 2015.

[20] OCDE, *Strengthening Health Information Infrastructure for Health Care Quality Governance*. 2013, Type: doi:https://doi.org/10.1787/9789264193505-en.

[21] K. Brameld, K. Spilsbury, L. Rosenwax, K. Murray, and J. Semmens, "Issues using linkage of hospital records and death certificate data to determine the size of a potential palliative care population," *Palliative Medicine*, vol. 31, no. 6, pp. 537–543, 2017.

[22] J. C. Doidge and K. Harron, "Demystifying probabilistic linkage," *International Journal of Population Data Science*, vol. 3, no. 1, Jan. 2018, Number: 1.

[23] K. Harron, C. Dibben, J. Boyd, A. Hjern, M. Azimaee, M. L. Barreto, and H. Goldstein, "Challenges in administrative data linkage for research:" *Big Data & Society*, Dec. 2017, Publisher: SAGE PublicationsSage UK: London, England.

[24] S. J. Grannis, J. M. Overhage, S. Hui, and C. J. McDonald, "Analysis of a Probabilistic Record Linkage Technique without Human Review," *AMIA Annual Symposium Proceedings*, vol. 2003, pp. 259–263, 2003.

[25] L. Gu, R. Baxter, D. Vickers, and C. Rainsford, "Record Linkage: Current Practice and Future Directions," CSIRO Mathematical and Information Sciences, Tech. Rep., 2003.

[26] H. Goldstein, K. Harron, and M. Cortina-Borja, "A scaling approach to record linkage," *Statistics in Medicine*, vol. 36, no. 16, pp. 2514–2521, Jul. 2017.

[27] B. P. Hejblum, G. M. Weber, K. P. Liao, N. P. Palmer, S. Churchill, N. A. Shadick, P. Szolovits, S. N. Murphy, I. S. Kohane, and T. Cai, "Probabilistic record linkage of de-identified research datasets with discrepancies using diagnosis codes," *Scientific Data*, vol. 6, Jan. 2019.

[28] R. Pita, E. Mendonça, S. Reis, M. Barreto, and S. Denaxas, "A Machine Learning Trainable Model to Assess the Accuracy of Probabilistic Record Linkage," in *Big Data Analytics and Knowledge Discovery*, L. Bellatreche and S. Chakravarthy, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 214–227, ISBN: 978-3-319-64283-3.

[29] M. A. Levin, H.-M. Lin, G. Prabhakar, P. J. McCormick, and N. N. Egorova, "Alive or dead: Validity of the Social Security Administration Death Master File after 2011," *Health Services Research*, vol. 54, no. 1, pp. 24–33, 2019.

[30] T. B. Newman and A. N. Brown, "Use of commercial record linkage software and vital statistics to identify patient deaths," *Journal of the American Medical Informatics Association: JAMIA*, vol. 4, no. 3, pp. 233–237, Jun. 1997.

[31] M. A. Jaro, "Probabilistic linkage of large public health data files," *Statistics in Medicine*, vol. 14, no. 5-7, pp. 491–498, Apr. 1995.

[32] I. P. Fellegi and A. B. Sunter, "A Theory for Record Linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, Dec. 1969, Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1969.10501049.

[33] L. Capuani, A. L. Bierrenbach, F. Abreu, P. L. Takecian, J. E. Ferreira, and E. C. Sabino, "Accuracy of a probabilistic record-linkage methodology used to track blood donors in the Mortality Information System database," *Cadernos de saude publica*, vol. 30, no. 8, pp. 1623–1632, Aug. 2014.

[34] D.-C. Li, C.-W. Liu, and S. C. Hu, "A learning method for the class imbalance problem with medical data sets," *Computers in Biology and Medicine*, vol. 40, no. 5, pp. 509–518, May 2010.

[35] M. P. J. v. d. Loo, "The stringdist package for approximate string matching," *The R Journal*, vol. 6, no. 1, pp. 111–122, 2014.

[36] D. Almeida, D. Gorender, M. Y. Ichihara, S. Sena, L. Menezes, G. C. G. Barbosa, R. L. Fiaccone, E. S. Paixão, R. Pita, and M. L. Barreto, "Examining the quality of record linkage process using nationwide Brazilian administrative databases to build a large birth cohort," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, p. 173, Jul. 2020.

[37] K. Harron, R. Gilbert, D. Cromwell, and J. van der Meulen, "Linking Data for Mothers and Babies in De-Identified Electronic Health Data," *PloS One*, vol. 11, no. 10, e0164667, 2016.

[38] D. R. Wilson, "Beyond probabilistic record linkage: Using neural networks and complex features to improve genealogical record linkage," in *The 2011 International Joint Conference on Neural Networks*, ISSN: 2161-4407, Jul. 2011, pp. 9–14.

[39] M. Guesdon, E. Benzenine, K. Gadouche, and C. Quantin, "Securizing data linkage in french public statistics," *BMC Medical Informatics and Decision Making*, vol. 16, Oct. 2016.

[40] R. Kingston, K. Sioris, J. Gualtieri, A. Brutlag, W. Droege, and T. G. Osimitz, "Post-market surveillance of consumer products: Framework for adverse event management," *Regulatory toxicology and pharmacology: RTP*, vol. 126, p. 105 028, Nov. 2021.

[41] C. A. Huber, T. D. Szucs, R. Rapold, and O. Reich, "Identifying patients with chronic conditions using pharmacy data in Switzerland: An updated mapping approach to the classification of medications," *BMC public health*, vol. 13, p. 1030, Oct. 2013.

[42] N. Hong, A. Wen, F. Shen, S. Sohn, S. Liu, H. Liu, and G. Jiang, "Integrating Structured and Unstructured EHR Data Using an FHIR-based Type System: A Case Study with Medication Data," *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, vol. 2017, pp. 74–83, 2018.

[43] OMEDIT, "OMEDIT Normandie - Bulletin d'information Avril 2016 - N°1," Apr. 2016.

[44] S. Cossin, L. Lebrun, G. Lobre, R. Loustau, V. Jouhet, R. Griffier, F. Mougin, G. Diallo, and F. Thiessard, "Romedi: An Open Data Source About French Drugs on the Semantic Web," *Studies in Health Technology and Informatics*, vol. 264, pp. 79–82, Aug. 2019.

[45] J. Rivierre and N. Morel, "Les médicaments génériques en France," p. 246, Jun. 2018.

[46] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant, "PubChem Substance and Compound databases," *Nucleic Acids Research*, vol. 44, no. D1, pp. D1202–1213, Jan. 2016.

[47] P. de Matos, R. Alcántara, A. Dekker, M. Ennis, J. Hastings, K. Haug, I. Spiteri, S. Turner, and C. Steinbeck, "Chemical Entities of Biological Interest: An update," *Nucleic Acids Research*, vol. 38, no. Database issue, pp. D249–254, Jan. 2010.

[48] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "DrugBank: A knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, no. Database issue, pp. D901–906, Jan. 2008.

[49] S. J. Nelson, K. Zeng, J. Kilbourne, T. Powell, and R. Moore, "Normalized names for clinical drugs: RxNorm at 6 years," *Journal of the American Medical Informatics Association: JAMIA*, vol. 18, no. 4, pp. 441–448, Aug. 2011.

[50] O. Bodenreider, R. Cornet, and D. J. Vreeman, "Recent Developments in Clinical Terminologies — SNOMED CT, LOINC, and RxNorm," *Yearbook of Medical Informatics*, vol. 27, no. 1, pp. 129–139, Aug. 2018.

[51] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014.

[52] O. Bodenreider and J. James, "The New SNOMED CT International Medicinal Product Model," in *ICBO*, 2018.

[53] R. Vander Stichele and D. Kalra, "Aggregations of Substance in Virtual Drug Models Based on ISO/CEN Standards for Identification of Medicinal Products (IDMP)," *Studies in Health Technology and Informatics*, vol. 294, pp. 377–381, May 2022.

[54] J. Nikiema and O. Bodenreider, "Comparing the Representation of Medicinal Products in RxNorm and SNOMED CT - Consequences on interoperability," in *ICBO*, 2019.

[55] F. Baader, S. Brandt, and C. Lutz, "Pushing the EL envelope," in *Proceedings of the 19th international joint conference on Artificial intelligence*, ser. IJCAI'05, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jul. 2005, pp. 364–369.

[56] L. Wang, Y. Zhang, M. Jiang, J. Wang, J. Dong, Y. Liu, C. Tao, G. Jiang, Y. Zhou, and H. Xu, "Toward a normalized clinical drug knowledge base in China-applying the RxNorm model to Chinese clinical drugs," *Journal of the American Medical Informatics Association: JAMIA*, vol. 25, no. 7, pp. 809–818, Jul. 2018.

[57] J. Grosjean, C Letord, J. Charlet, X Aimé, L Danès, J. Rio, I Zana, S. J. Darmoni, and C Duclos, "Un modèle sémantique d'identification du médicament en France," in *Atelier IA & Santé 2019*, Toulouse, France: Fleur Mougin and Brigitte Séroussi, Jul. 2019.

[58] C. L. Ventola, "Big Data and Pharmacovigilance: Data Mining for Adverse Drug Events and Interactions," *Pharmacy and Therapeutics*, vol. 43, no. 6, pp. 340–351, Jun. 2018.

[59] S. Sohn, C. Clark, S. R. Halgrim, S. P. Murphy, C. G. Chute, and H. Liu, "MedXN: An open source medication extraction and normalization tool for clinical text," *Journal of the American Medical Informatics Association : JAMIA*, vol. 21, no. 5, pp. 858–865, Sep. 2014.

[60] J. Jovanović and E. Bagheri, "Semantic annotation in biomedicine: The current landscape," *Journal of Biomedical Semantics*, vol. 8, Sep. 2017.

[61] J. Jouffroy, S. F. Feldman, I. Lerner, B. Rance, A. Burgun, and A. Neuraz, "Hybrid Deep Learning for Medication-Related Information Extraction From Clinical Texts in French: MedExt Algorithm Development Study," *JMIR medical informatics*, vol. 9, no. 3, e17934, Mar. 2021.

[62] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette, "Bio2RDF: Towards a mashup to build bioinformatics knowledge systems," *Journal of Biomedical Informatics*, vol. 41, no. 5, pp. 706–716, Oct. 2008.

[63] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web (Berners-Lee et. al 2001)," May 2001.

[64] I. Robu, V. Robu, and B. Thirion, "An introduction to the Semantic Web for health sciences librarians," *Journal of the Medical Library Association*, vol. 94, no. 2, pp. 198–205, Apr. 2006.

[65] S. S. Epp, *Discrete Mathematics with Applications*, Fifth. Cengage Learning, Jan. 2019, ISBN: 978-1-337-69419-3.

[66] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2020.

[67] A. Sheth, S. Padhee, and A. Gyrard, *Knowledge Graphs and Knowledge Networks: The Story in Brief*, Number: arXiv:2003.03623 arXiv:2003.03623 [cs], Mar. 2020.

[68] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "BRAT: A Web-based Tool for NLP-assisted Text Annotation," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, ser. EACL '12, Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 102–107.

[69] D. Mouromtsev, D. Pavlov, Y. Emelyanov, A. Morozov, D. Razdyakonov, and M. Galkin, "The Simple Web-based Tool for Visualization and Sharing of Semantic Data and Ontologies," in *International Semantic Web Conference (Posters & Demos)*, Oct. 2015.

[70] S. B. Johnson, S. Bakken, D. Dine, S. Hyun, E. Mendonça, F. Morrison, T. Bright, T. Van Vleck, J. Wrenn, and P. Stetson, "An electronic health record based on structured narrative," *Journal of the American Medical Informatics Association: JAMIA*, vol. 15, no. 1, pp. 54–64, Feb. 2008.

[71] E. Tseytlin, K. Mitchell, E. Legowski, J. Corrigan, G. Chavan, and R. S. Jacobson, "NOBLE - Flexible concept recognition for large-scale biomedical natural language processing," *BMC bioinformatics*, vol. 17, p. 32, Jan. 2016.

[72] F. Pech, A. Martinez, H. Estrada, and Y. Hernandez, *Semantic Annotation of Unstructured Documents Using Concepts Similarity*, Research Article, ISSN: 1058-9244 Pages: e7831897 Publisher: Hindawi Volume: 2017, Dec. 2017.

[73] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program," *Proceedings. AMIA Symposium*, pp. 17–21, 2001.

[74] O. Bodenreider, "The Unified Medical Language System (UMLS): Integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. Database issue, pp. D267–270, Jan. 2004.

[75] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: Historical perspective and recent advances," *Journal of the American Medical Informatics Association: JAMIA*, vol. 17, no. 3, pp. 229–236, Jun. 2010.

[76] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications," *Journal of the American Medical Informatics Association: JAMIA*, vol. 17, no. 5, pp. 507–513, Oct. 2010.

[77] L. Soldaini, *QuickUMLS: A Fast, Unsupervised Approach for Medical Concept Extraction*, 2016.

[78] N. Okazaki and J. Tsujii, "Simple and Efficient Algorithm for Approximate Dictionary Matching," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 851–859.

[79] C. A. Deisseroth, J. Birgmeier, E. E. Bodle, J. N. Kohler, D. R. Matalon, Y. Nazarenko, C. A. Genetti, C. A. Brownstein, K. Schmitz-Abe, K. Schoch, H. Cope, R. Signer, J. A. Martinez-Agosto, V. Shashi, A. H. Beggs, M. T. Wheeler, J. A. Bernstein, and G. Bejerano, "ClinPhen extracts and prioritizes patient phenotypes directly from medical records to expedite genetic disease diagnosis," *Genetics in medicine : official journal of the American College of Medical Genetics*, vol. 21, no. 7, pp. 1585–1593, Jul. 2019.

[80] C. Jonquet, N. H. Shah, and M. A. Musen, "The Open Biomedical Annotator," *Summit on Translational Bioinformatics*, vol. 2009, pp. 56–60, Mar. 2009.

[81] P. Gooch, "A modular, open-source information extraction framework for identifying clinical concepts and processes of care in clinical narratives," Accepted: 2012, Ph.D. City University, 2012.

[82] D. R. Morrison, "PATRICIA—Practical Algorithm To Retrieve Information Coded in Alphanumeric," *Journal of the ACM*, vol. 15, no. 4, pp. 514–534, Oct. 1968.

[83] V. Singh, *Replace or Retrieve Keywords In Documents at Scale*, Number: arXiv:1711.00046 arXiv:1711.00046 [cs], Nov. 2017.

[84] A. V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, Jun. 1975.

[85] S. Stewart, M. Von Maltzahn, and S. Abidi, *Comparing metamap to mgrep as a tool for mapping free text to formal medical lexicons*. Jan. 2012, vol. 895.

[86] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 1979.

[87] J. Clément, P. Flajolet, and B. Vallée, "The Analysis of Hybrid Trie Structures," report, INRIA, 1997.

[88] C.-Y. R. Huang, C.-Y. Lai, and K.-T. T. Cheng, "CHAPTER 4 - Fundamentals of algorithms," in *Electronic Design Automation*, L.-T. Wang, Y.-W. Chang, and K.-T. T. Cheng, Eds., Boston: Morgan Kaufmann, Jan. 2009, pp. 173–234, ISBN: 978-0-12-374364-0.

[89] K. Schulz and S. Mihov, "Fast String Correction with Levenshtein-Automata," *International Journal of Document Analysis and Recognition*, vol. 5, pp. 67–85, 2002.

[90] A. Névéol, A. Robert, F. Grippo, C. Morgand, C. Orsi, L. Pelikan, L. Ramadier, G. Rey, and P. Zweigenbaum, "CLEF eHealth 2018 Multilingual Information Extraction Task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian," in *CLEF*, 2018.

[91] A. Atutxa, A. Casillas, N. Ezeiza, V. Fresno-Fernández, I. Goenaga, K. Gojenola, R. Martínez, M. O. Anchordoqui, and O. Perez-de Viñaspre, "IxaMed at CLEF eHealth 2018 Task 1: ICD10 Coding with a Sequence-to-Sequence Approach," in *CLEF*, 2018.

[92] A. Miranda-Escalada, A. Gonzalez-Agirre, J. Armengol-Estapé, and M. Krallinger, "Overview of automatic clinical coding: Annotations, guidelines, and solutions for non-English clinical cases at CodiEsp track of eHealth CLEF 2020," *CEUR-WS*, 2020.

[93] A. Blanco, A. Pérez, and A. Casillas, "IXA-AAA at CLEF eHealth 2020 CodiEsp Automatic classification of medical records with Multi-label classifiers and Similarity Match Coders". In: Working Notes of Conference and Labs of the Evaluation (CLEF) Forum," *CEUR-WS*, 2020.

[94] N. Garcia-Santa and C. Kendrick, "FLE at CLEF eHealth 2020: Text Mining and Semantic Knowledge for Automated Clinical Encoding". In: Working Notes of Conference and Labs of the Evaluation (CLEF) Forum," *CEUR-WS*, 2020.

[95] Antonio Miranda-Escalada, E. Farré-Maduell, S. Lima-López, D. Estrada, L. Gascó, and M. Krallinger, "Mention detection, normalization & classification of species, pathogens, humans and food in clinical documents: Overview of LivingNER shared task and resources," *Procesamiento del Lenguaje Natural*, 2022.

[96] E. Zotova, A. Garcia-Pablos, N. Perez, P. Turon, and M. Cuadros, "Vicomtech at livingner 2022," *CEUR workshop proceedings*, Sep. 2022.

[97] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Number: arXiv:1810.04805 arXiv:1810.04805 [cs], May 2019.

[98] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, "Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *npj Digital Medicine*, vol. 4, no. 1, pp. 1–13, May 2021, Number: 1 Publisher: Nature Publishing Group.

[99] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel, *Extracting Training Data from Large Language Models*, Number: arXiv:2012.07805 arXiv:2012.07805 [cs], Jun. 2021.

[100] L. Luo, S. Yan, P.-T. Lai, D. Veltri, A. Oler, S. Xirasagar, R. Ghosh, M. Similuk, P. N. Robinson, and Z. Lu, "PhenoTagger: A Hybrid Method for Phenotype Concept Recognition using Human Phenotype Ontology," *Bioinformatics (Oxford, England)*, btab019, Jan. 2021.

[101] H. Liu, Y. A. Lussier, and C. Friedman, "A study of abbreviations in the UMLS," *Proceedings. AMIA Symposium*, pp. 393–397, 2001.

[102] S. Pakhomov, T. Pedersen, and C. G. Chute, "Abbreviation and acronym disambiguation in clinical discourse," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, pp. 589–593, 2005.

[103] H. Xu, P. D. Stetson, and C. Friedman, "Methods for Building Sense Inventories of Abbreviations in Clinical Notes," *Journal of the American Medical Informatics Association : JAMIA*, vol. 16, no. 1, pp. 103–108, 2009.

[104] P. D. Stetson, S. B. Johnson, M. Scotch, and G. Hripcsak, "The sublanguage of cross-coverage.," *Proceedings of the AMIA Symposium*, pp. 742–746, 2002.

[105] V. Joopudi, B. Dandala, and M. Devarakonda, "A convolutional route to abbreviation disambiguation in clinical text," *Journal of Biomedical Informatics*, vol. 86, pp. 71–78, Oct. 2018.

[106] X. Du, R. Zhu, Y. Li, and A. Anjum, "Language model-based automatic prefix abbreviation expansion method for biomedical big data analysis," *Future Generation Computer Systems*, vol. 98, pp. 238–251, Sep. 2019.

[107] M. Oleynik, M. Kreuzthaler, and S. Schulz, "Unsupervised Abbreviation Expansion in Clinical Narratives," *Studies in Health Technology and Informatics*, vol. 245, pp. 539–543, 2017.

[108] D. A. Bloom, "Acronyms, abbreviations and initialisms," *BJU international*, vol. 86, no. 1, pp. 1–6, Jul. 2000.

[109] D. L. Mowery, B. R. South, L. Christensen, J. Leng, L.-M. Peltonen, S. Salanterä, H. Suominen, D. Martinez, S. Velupillai, N. Elhadad, G. Savova, S. Pradhan, and W. W. Chapman, "Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: ShARe/CLEF eHealth Challenge 2013, Task 2," *Journal of Biomedical Semantics*, vol. 7, p. 43, Jul. 2016.

[110] J. T. Chang, H. Schütze, and R. B. Altman, "Creating an Online Dictionary of Abbreviations from MEDLINE," *Journal of the American Medical Informatics Association*, vol. 9, no. 6, pp. 612–620, Nov. 2002.

[111] W. J. Long, "Parsing Free Text Nursing Notes," *AMIA Annual Symposium Proceedings*, vol. 2003, p. 917, 2003.

[112] S. Gaudan, H. Kirsch, and D. Rebholz-Schuhmann, "Resolving abbreviations to their senses in Medline," *Bioinformatics*, vol. 21, no. 18, pp. 3658–3664, Sep. 2005.

[113] S. Moon, S. Pakhomov, N. Liu, J. O. Ryan, and G. B. Melton, "A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources," *Journal of the American Medical Informatics Association : JAMIA*, vol. 21, no. 2, pp. 299–307, Mar. 2014.

[114] N. Okazaki, S. Ananiadou, and J. Tsujii, "Building a high-quality sense inventory for improved abbreviation disambiguation," *Bioinformatics*, vol. 26, no. 9, pp. 1246–1253, May 2010.

[115] H. Xu, P. D. Stetson, and C. Friedman, "A Study of Abbreviations in Clinical Notes," *AMIA Annual Symposium Proceedings*, vol. 2007, pp. 821–825, 2007.

[116] N. Okazaki and S. Ananiadou, "Building an abbreviation dictionary using a term recognition approach," *Bioinformatics (Oxford, England)*, vol. 22, no. 24, pp. 3089–3095, Dec. 2006.

[117] M. Ciosici, T. Sommer, and I. Assent, "Unsupervised Abbreviation Disambiguation Contextual disambiguation using word embeddings," *arXiv:1904.00929 [cs]*, May 2019, arXiv: 1904.00929.

[118] A. Terada, T. Tokunaga, and H. Tanaka, *Automatic Expansion of Abbreviations by using Context and Character Information Abstract*. 2004.

[119] J. Toole, "A Hybrid Approach to the Identification and Expansion of Abbreviations," in *In RIAO 2000 6th Conference on Content-Based Multimedia Information Access*, 2000, pp. 725–736.

[120] C. Vo, T. Cao, and B. Ho, "Abbreviation Detection in Vietnamese Clinical Texts," *VNU Journal of Science: Computer Science and Communication Engineering*, vol. 34, no. 2, Dec. 2018, Number: 2.

[121] A. S. Schwartz and M. A. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical text," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 451–462, 2003.

[122] W. Zhou, V. I. Torvik, and N. R. Smalheiser, "ADAM: Another database of abbreviations in MEDLINE," *Bioinformatics (Oxford, England)*, vol. 22, no. 22, pp. 2813–2818, Nov. 2006.

[123] Y. Wu, J. C. Denny, S. Trent Rosenbloom, R. A. Miller, D. A. Giuse, L. Wang, C. Blanquicett, E. Soysal, J. Xu, and H. Xu, "A long journey to short abbreviations: Developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD)," *Journal of the American Medical Informatics Association: JAMIA*, vol. 24, no. e1, e79–e86, Apr. 2017.

[124] H. Schmid, "Improvements in Part-of-Speech Tagging with an Application to German.," *Proceedings of the ACL SIGDAT-Workshop*, 1995.

[125] J. A. Lossio-Ventura, C. Jonquet, M. Roche, and M. Teisseire, "BIOTEX: A system for Biomedical Terminology Extraction, Ranking, and Validation," in *ISWC: International Semantic Web Conference*, ser. CEUR Workshop Proceedings, n° 1272, Issue: 1272, vol. CEUR Workshop Proceedings, Riva del Garda, Italy, Oct. 2014, pp. 157–160.

[126] Z. Harris, "Mathematical structures of language," in *Interscience tracts in pure and applied mathematics*, 1968.

[127] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz, "A survey of word embeddings for clinical text," *Journal of Biomedical Informatics: X*, vol. 4, p. 100 057, Dec. 2019.

[128] S.-H. Cha, *Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions*. 2007.

[129] L. V. Grossman, E. G. Mitchell, G. Hripcsak, C. Weng, and D. K. Vawdrey, "A method for harmonization of clinical abbreviation and acronym sense inventories," *Journal of Biomedical Informatics*, vol. 88, pp. 62–69, Dec. 2018.

[130] J. Pustejovsky, Casta&#241, J. O, B. Cochran, M. Kotecki, and M. Morrell, "Automatic Extraction of Acronym-meaning Pairs from MEDLINE Databases," *MEDINFO 2001*, pp. 371–375, 2001, Publisher: IOS Press.

[131] H. Xu, Y. Wu, N. Elhadad, P. D. Stetson, and C. Friedman, "A new clustering method for detecting rare senses of abbreviations in clinical notes," *Journal of Biomedical Informatics*, vol. 45, no. 6, pp. 1075–1083, Dec. 2012.

[132] L. Heidemann, J. Law, and R. J. Fontana, "A Text Searching Tool to Identify Patients with Idiosyncratic Drug-Induced Liver Injury," *Digestive Diseases and Sciences*, vol. 62, no. 3, pp. 615–625, Mar. 2017.

[133] P. Raghavan, J. L. Chen, E. Fosler-Lussier, and A. M. Lai, "How essential are unstructured clinical narratives and information fusion to clinical trial recruitment?" *AMIA Summits on Translational Science Proceedings*, vol. 2014, pp. 218–223, Apr. 2014.

[134] K. E. Walsh, K. A. Marsolo, C. Davis, T. Todd, B. Martineau, C. Arbaugh, F. Verly, C. Samson, and P. Margolis, "Accuracy of the medication list in the electronic health record-implications for care, research, and improvement," *Journal of the American Medical Informatics Association: JAMIA*, vol. 25, no. 7, pp. 909–912, Jul. 2018.

[135] L. Seyfried, D. A. Hanauer, D. Nease, R. Albeiruti, J. Kavanagh, and H. C. Kales, "Enhanced identification of eligibility for depression research using an electronic medical record search engine," *International Journal of Medical Informatics*, vol. 78, no. 12, e13–18, Dec. 2009.

[136] L. M. Etheredge, "A rapid-learning health system," *Health Affairs (Project Hope)*, vol. 26, no. 2, w107–118, Apr. 2007.

[137] A. H. Nordo, H. P. Levaux, L. B. Becnel, J. Galvez, P. Rao, K. Stem, E. Prakash, and R. D. Kush, "Use of EHRs data for clinical research: Historical progress and current applications," *Learning Health Systems*, vol. 3, no. 1, e10076, Jan. 2019.

[138] D. L. Alonso, A. Rose, C. Plaisant, and K. L. Norman, "Viewing personal history records: A comparison of tabular format and graphical presentation using lifelines," *Behaviour & Information Technology*, vol. 17, no. 5, pp. 249–262, 1998, Place: United Kingdom Publisher: Taylor & Francis.

[139] V. L. West, D. Borland, and W. E. Hammond, "Innovative information visualization of electronic health record data: A systematic review," *Journal of the American Medical Informatics Association: JAMIA*, vol. 22, no. 2, pp. 330–339, Mar. 2015.

[140] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, "LifeLines: Using visualization to enhance navigation and analysis of patient records.," *Proceedings of the AMIA Symposium*, pp. 76–80, 1998.

[141] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, ISSN: 1049-2615, Sep. 1996, pp. 336–343.

[142] R. Bade, S. Schlechtweg, and S. Miksch, "Connecting time-oriented data and information to a coherent interactive visualization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '04, New York, NY, USA: Association for Computing Machinery, Apr. 2004, pp. 105–112, ISBN: 978-1-58113-702-6.

[143] D. S. Pieczkiewicz, S. M. Finkelstein, and M. I. Hertz, "Design and Evaluation of a Web-Based Interactive Visualization System for Lung Transplant Home Monitoring Data," *AMIA Annual Symposium Proceedings*, vol. 2007, pp. 598–602, 2007.

[144] W. Horn, C. Popow, and L. Unterasinger, "Support for fast comprehension of ICU data: Visualization using metaphor graphics," *Methods of Information in Medicine*, vol. 40, no. 5, pp. 421–424, 2001.

[145] M. Pohl, S. Wiltner, A. Rind, W. Aigner, S. Miksch, T. Turic, and F. Drexler, "Patient Development at a Glance: An Evaluation of a Medical Data Visualization," in *Human-Computer Interaction – INTERACT 2011*, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 292–299, ISBN: 978-3-642-23768-3.

[146] D. A. Hanauer, "EMERSE: The Electronic Medical Record Search Engine," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, p. 941, 2006.

[147] K. Zheng, Q. Mei, and D. A. Hanauer, "Collaborative search in electronic health records," *Journal of the American Medical Informatics Association : JAMIA*, vol. 18, no. 3, pp. 282–291, 2011.

[148] L. Yang, Q. Mei, K. Zheng, and D. A. Hanauer, "Query Log Analysis of an Electronic Health Record Search Engine," *AMIA Annual Symposium Proceedings*, vol. 2011, pp. 915–924, 2011.

[149] D. A. Hanauer, M. J. Englesbe, J. A. Cowan, and D. A. Campbell, "Informatics and the American College of Surgeons National Surgical Quality Improvement Program: Automated processes could replace manual record review," *Journal of the American College of Surgeons*, vol. 208, no. 1, pp. 37–41, Jan. 2009.

[150] D. A. Hanauer, J. S. Barnholtz-Sloan, M. F. Beno, G. Del Fiol, E. B. Durbin, O. Gologorskaya, D. Harris, B. Harnett, K. Kawamoto, B. May, E. Meeks, E. Pfaff, J. Weiss, and K. Zheng, "Electronic Medical Record Search Engine (EMERSE): An Information Retrieval Tool for Supporting Cancer Research," *JCO clinical cancer informatics*, vol. 4, pp. 454–463, May 2020.

[151] F. Thiessard, F. Mougin, G. Diallo, V. Jouhet, S. Cossin, N. Garcelon, B. Campillo, W. Jouini, J. Grosjean, P. Massari, N. Griffon, M. Dupuch, F. Tayalati, E. Dugas, A. Balvet, N. Grabar, S. Pereira, B. Frandji, S. Darmoni, and M. Cuggia, "RAVEL: Retrieval and visualization in ELectronic health records," *Studies in Health Technology and Informatics*, vol. 180, pp. 194–198, 2012.

[152] G. Diallo, N. Grabar, F. Thiessard, N. Garcelon, J. Grosjean, M. Dupuch, S. Bento-Pereira, B. Frandji, S. J. Darmoni, and M. Cuggia, "Towards complex queries on data from complex patients.," in *AMIA*, 2012.

[153] T. Baudel and G. Brochard, "VIP : Système de visualisation de données patient," Palaiseau, France, 2018.

[154] E. Sylvestre, G. Bouzillé, E. Chazard, C. His-Mahier, C. Riou, and M. Cuggia, "Combining information from a clinical data warehouse and a pharmaceutical database to generate a framework to detect comorbidities in electronic health records," *BMC medical informatics and decision making*, vol. 18, no. 1, p. 9, 2018.

[155] J. O. Wrenn, D. M. Stein, S. Bakken, and P. D. Stetson, "Quantifying clinical narrative redundancy in an electronic health record," *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, no. 1, pp. 49–53, 2010.

[156] Y. Chen, R. J. Carroll, E. R. M. Hinz, A. Shah, A. E. Eyler, J. C. Denny, and H. Xu, "Applying active learning to high-throughput phenotyping algorithms for electronic health records data," *Journal of the American Medical Informatics Association : JAMIA*, vol. 20, no. e2, e253–e259, Dec. 2013.