Méthodes structurelles et sémantiques pour la mise en correspondance de cas textuels de dysmorphies fœtales

Yves Jean Vincent Parès

Université Pierre et Marie Curie

École doctorale Pierre-Louis de Santé Publique à Paris : Épidémiologie et Sciences de l'Information Biomédicale

Inserm UMR_S 1142 / Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en Santé

Méthodes structurelles et sémantiques pour la mise en correspondance de cas textuels de dysmorphies fœtales

———

Par Yves Parès

Thèse de doctorat d'informatique médicale
dirigée par Marie-Christine Jaulent et Xavier Aimé

Présentée et soutenue publiquement le 1er décembre 2016

Devant un jury composé de :

Isabelle Bichindaritz	Professeur associé, CS Department, State Univ. of New York Oswego	Rapporteur
Chantal Reynaud	Professeur Univ. Paris 11, LRI (UMR CNRS 8623)	Rapporteur
Pierre Zweigenbaum	DR CNRS, LIMSI (UPR CNRS 3251)	Examinateur
Jean-Marie Jouannic	PU-PH, Hôpital Trousseau (APHP) & Univ. Paris 6	Examinateur
Xavier Aimé	Chercheur INSERM, LIMICS (UMRS 1142)	Co-directeur
Marie-Christine Jaulent	DR INSERM, LIMICS (UMRS 1142)	Directrice

Table des matières

Remerciements

Je remercie les membres du projet Accordys et en particulier Xavier Aimé et Jean Charlet du LIMICS pour m'avoir accompagné pendant cette thèse et m'avoir de nombreuses fois aidé dans mon travail.

Je remercie le LIMICS (à l'époque ICS, Ingénierie des Connaissances en Santé) pour m'avoir accepté en thèse et l'école doctorale Pierre-Louis de Santé Publique de Paris 6 pour m'avoir accordé ce contrat doctoral.

Je remercie Clément Dalloux, que j'ai encadré en stage de Master 2 de traitement automatique des langues, pour son travail sur Accordys et sa bonne humeur.

Je remercie Stefan Darmoni du CISMeF pour m'avoir permis d'utiliser l'outil d'annotation ECMT et Chloé Cabot pour m'avoir aidé toute une journée dans cette phase de l'étude.

Je remercie mes co-thésards Alexandre Galopin, Meriem Maaroufi, Romain Ng et Marion Richard ainsi que Damien Leprovost, post-doctorant, pour tous les bons moments partagés au laboratoire et en conférence. Tout ceci n'aurait pas été pareil sans vous.

Je remercie mes parents qui m'ont toujours prodigué soutien moral et logistique dans mon parcours, et ma chère Annie qui a du vivre avec moi pendant tout ce temps. Figurez-vous qu'il paraît que j'étais pénible sur la fin.

Et bien entendu, je remercie Marie-Christine Jaulent pour son soutien, son suivi et sa gentillesse sans failles durant ces années de thèse. 👍

Et je vous remercie, vous, lecteur de ce mémoire, pour l'attention que vous portez à ce travail. On ne vous le dit pas assez mais vous êtes quelqu'un de bien. Maintenant commençons.

Chapitre 1

Introduction

L'ingénierie des connaissances se place dans le cadre de l'intelligence artificielle. Située au croisement de l'héritage des systèmes experts, du traitement du langage naturel et des sciences cognitives, l'ingénierie des connaissances cherche à répondre à l'un des plus gros défis de l'IA : présenter les informations, les procédés de raisonnement et les connaissances spécifiques à un domaine métier dans des représentations dénuées d'ambigüités mais permettant de capturer les subtilités et nuances des langages de ces domaines pour permettre à un programme informatique d'automatiser une partie des traitements et raisonnements effectués habituellement par les spécialistes de ces domaines. Le but n'est jamais de remplacer l'expert, toujours de l'accompagner en facilitant l'accès aux connaissances et en accélérant les procédés cognitifs qui peuvent l'être. Dans cette thèse, nous nous intéressons plus spécifiquement à l'ingénierie des connaissances médicales dans le domaine de la fœtopathologie, qui est l'étude des anomalies et dysmorphies du fœtus. Les sources d'information utilisées par les ingénieurs de la connaissances sont multiples, mais elles sont principalement de deux types : les documents écrits par les spécialistes du domaine considéré et les entretiens effectués avec ces mêmes spécialistes. Les missions de l'ingénierie des connaissances sont donc de comprendre et décrire des domaines de la connaissance et rendre réutilisable et réutiliser les connaissances de ces domaines.

Plusieurs approches peuvent êtres utilisées, soit séparément soit conjointement, pour essayer de mener à bien ces missions. On pourrait citer deux grandes familles d'approches : celles basées sur les exemples et celles basées sur les formalisations. La première utilise l'inférence à partir de données enregistrées, l'apprentissage automatique (*machine learning*) ou le raisonnement par analogie. L'un de ses principaux représentant est le raisonnement à partir de cas (RàPC), où le but est de rapprocher des problèmes actuels à des problèmes déjà résolus par le passé, le présupposé étant que deux problèmes similaires devraient également avoir des solutions similaires. Une notion clef de cette famille est donc celle de *similarité* entre cas ou exemples, afin de pouvoir effectuer des regroupements.

La seconde famille est celles des méthodes passant par une compilation préalable des connaissances du domaine. À l'heure actuelle, sous l'impulsion du développement du web sémantique, son principal représentant est la construction d'ontologies. Une ontologie vise à représenter les propriétés de ce qui

existe (aussi bien les entités concrètes que les concepts abstraits). Une ontologie d'un domaine constitue donc un modèle de données représentatif de l'ensemble des concepts manipulés par ce domaine, avec notamment les liens qui existent entre ces concepts. Tout cela ne veut pas dire que ce type d'approche se passe de données du domaine, mais que ces données ne sont pas utilisées telles quelles *in fine*. Si elles existent, elles sont compilées avec les connaissances, soit extraites de textes décrivant le domaine, soit obtenues lors d'entretiens avec les spécialistes.

Le lecteur a pu déjà remarquer l'utilisation faite des mots *information*, *connaissance* et *donnée*. Ils ne sont pas utilisés comme synonymes. Une *donnée* concerne un élément d'information concernant un cas ou un individu en particulier. On appelle *donnée brute* une donnée relevée – par exemple lors d'un examen – par l'observation ou l'utilisation d'instruments de mesure ou de capteurs. Une fois relevée, une donnée brute a donc un caractère factuel et intemporel. Une *connaissance* renseigne sur un concept ou un principe valide dans un domaine donné à une époque donnée. Selon l'avancée de la recherche dans ce domaine, les *connaissances* peuvent donc être amenées à varier. L'application des *connaissances* aux *données brutes* permet d'obtenir des *données inférées*, qui seront remises en cause si bien entendu les connaissances sont remises en cause. Dans notre cas, un traité de fœtopathologie contient donc surtout des *connaissances*, et un compte rendu d'examen des *données*. Voici un exemple :

— le fait de savoir que l'enfant de Mme A présente des malformations crâniennes et notamment une distance interorbitaire de 40mm est une *donnée brute*,
— le fait de savoir qu'à l'examen cytogénétique on a observé une mutation du gène FGFR2 est également une donnée brute,
— le fait de savoir que des malformations crâniennes comprenant un hypertélorisme, donc une distance interorbitaire trop grande, peuvent être liées à la maladie de Crouzon est une *connaissance*,
— le tableau génétique de la maladie de Crouzon est également une connaissance,
— le fait de savoir (peut-être après examens supplémentaires) que l'enfant de Mme A est atteint de la maladie de Crouzon est une *donnée inférée*.

Venant du domaine du développement et de l'architecture logicielle et avec un intérêt pour l'intelligence artificielle et le traitement du langage, c'est assez naturellement que je me suis orienté vers un travail de thèse me permettant de parfaire mes connaissances de ces domaines et de les appliquer à l'aide à la décision en médecine. Le reste de ce chapitre présente tout d'abord l'intérêt que représente le texte pour les ingénieurs de la connaissance médicale. Viennent ensuite l'objectif applicatif, les verrous scientifiques ciblés et l'approche choisie par cette thèse et plus généralement par le projet ANR Accordys dans lequel elle se replace.

1.1 Valeur du texte en ingénierie des connaissances médicales

La médecine partage avec bien d'autres domaines l'omniprésence de la représentation textuelle d'informations. Que cela soit dans les manuels de référence des spécialités, dans les comptes rendus d'examens ou dans les prescriptions et traitements, on retrouve la nécessité d'exprimer l'information

sous des formats textuels ayant divers niveaux de structuration.

En France notamment, comme l'explique Charlet (2002), les tentatives d'informatisation des dossiers médicaux des patients ont montré la difficulté résidant dans la conservation du contexte de l'information lorsque des dossiers initialement textuels sont représentés en base de données. De plus, même en présence d'une aide apportée par un système de codage de l'information (via des formulaires associés à un thésaurus, par exemple), les médecins montrent une propension à adjoindre des notes en texte libre aux parcours cliniques qui, eux, sont représentés avec un codage particulier.

Il est également à noter que les efforts de formalisation des ingénieurs de la connaissance arriveront toujours après les besoins des médecins en termes de représentation de l'information. Autrement dit, quel que soit le niveau de formalisation atteint, l'avancée perpétuelle de la médecine fait qu'il existera toujours une période pendant laquelle les besoins de représentation des médecins n'auront pas encore reçu de réponse de la part des ingénieurs de la connaissance, période pendant laquelle la seule option pour les médecins sera d'utiliser le texte libre.

Tout ceci tend à indiquer que le texte est un acteur indissociable du processus de sérialisation de l'information médicale, qu'il aura toujours une valeur aux yeux des spécialistes et qu'il restera donc nécessaire de composer avec lui.

1.1.1 Espace des représentations et continuum de structuration

L'extraction d'information d'un texte vise à construire un modèle de données et à l'instancier pour chaque document à partir des éléments trouvés, manuellement ou non, dans un texte libre. Ces deux états, texte libre et données représentées dans un modèle particulier, constituent deux points d'un espace unidimensionnel de représentations que nous appelons *continuum de structuration*, une notion qui rejoint le spectre caractérisé par Richter et R. Weber (2013b). Il s'agit d'une continuité, c'est à dire d'un ensemble de représentations concevables sur lequel on peut progresser, c'est à dire augmenter la quantité d'information exploitable de manière automatisée par un programme (en structurant et/ou formalisant de plus en plus la donnée). Si de surcroît on le fait sans changer la quantité d'information compréhensible par un humain, nous dirons que ce continuum est *bidirectionnel*. À l'inverse, s'il y a perte d'éléments d'information ou bien introduction d'ambiguïtés lorsque l'on passe d'une représentation moins structurée à une autre plus structurée, alors nous dirons qu'il est *unidirectionnel*, puisqu'il sera impossible de reconstituer la source d'information originale après s'être déplacé le long du continuum.

Plus formellement, on définit un continuum de structuration comme un triplet $(R, <, P)$, où :

— R est un ensemble de représentations ;
— $<$ est une relation d'ordre totale sur R ;
— P est un ensemble de fonctions de $R \to R$ strictement croissantes selon $<$, appelées *progressions*.

Si pour tout $p \in P$ il existe son inverse p^{-1}, alors le continuum de structuration est bidirectionnel.

La figure 1.1 est une proposition de présentation de l'espace des représentations, en plaçant différents

modèles usuels de représentation de données sur deux axes :

1. la capacité à exploiter la structure de cette représentation au sein d'un programme. Cette capacité augmente au fur et à mesure que cette structure est rendue de plus en plus explicite ;
2. la quantité d'informations non ambigues que cette représentation peut conserver, autrement dit sa capacité à rendre la donnée plus formelle.

On ajoute une troisième caractérisation, mais qui n'est pas indépendante des deux précédentes :

3. la quantité de travail nécessaire à l'opérationalisation de cette représentation et à la création de données l'utilisant.

Dans cet espace des représentations, une représentation précise va donc être un *point* et un modèle de représentation une *surface*. Les modèles présentés ici sont généralement assez génériques et flexibles. Ainsi, selon la façon dont il est utilisé, un même modèle peut permettre d'obtenir des représentations plus ou moins structurées ou plus ou moins formelles, et cette liberté d'utilisation d'un modèle ou langage de représentation rend floues les limites de l'utilisabilité de ce dernier. Cependant, il existe quand même pour un modèle donné un cadre de cas d'usages en dehors duquel l'utilisation de ce modèle peut être considérée comme inappropriée car abusive ou délétère [1].

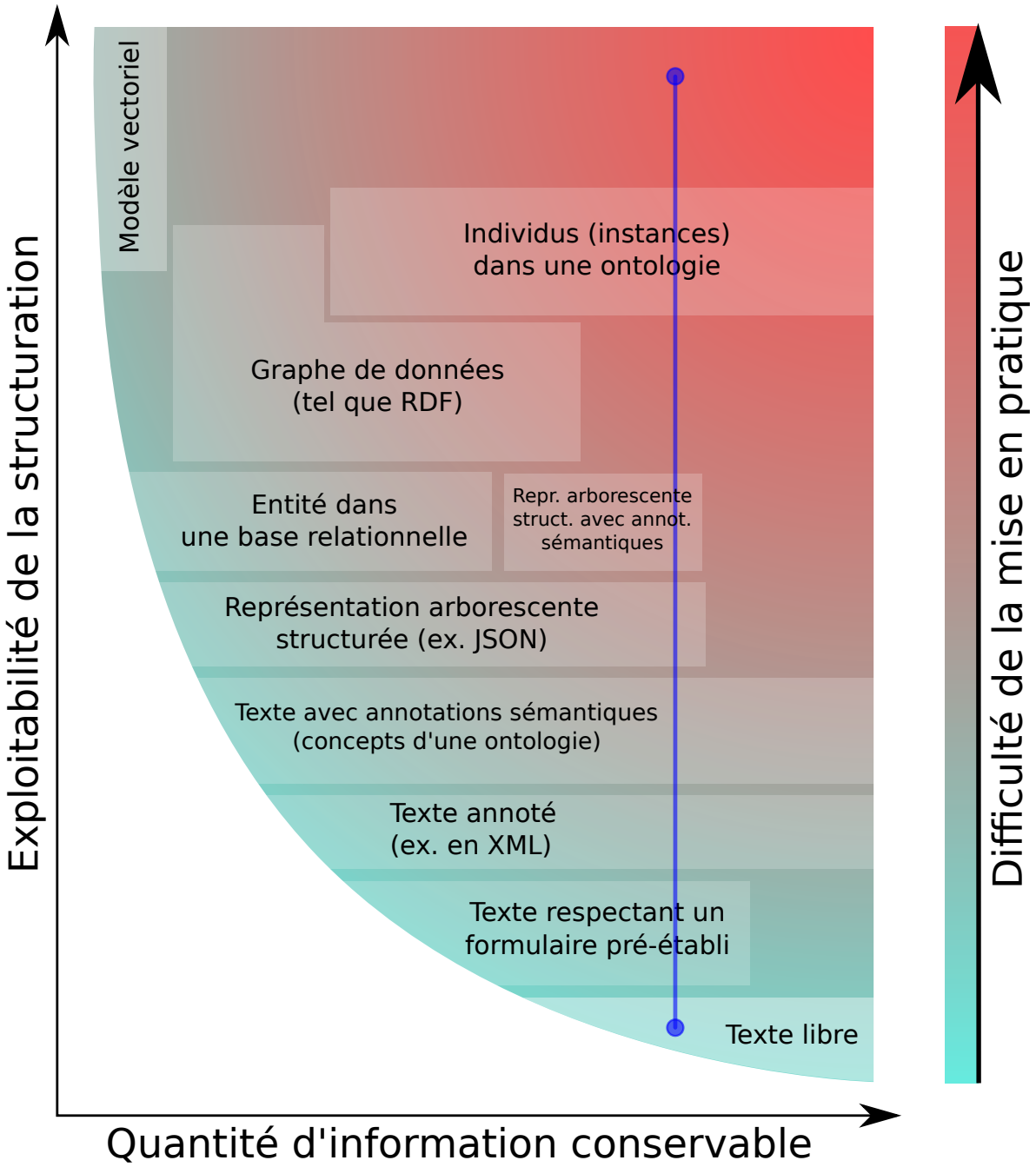
Sur cette même figure 1.1, le segment bleu est un exemple de continuum de structuration bidirectionnel, de la même manière que tout segment de droite vertical traçable sur cette figure. L'intégralité d'un continuum n'est pas forcément réalisable en pratique. En effet, l'extrême structuration nécessiterait l'incorporation au modèle de données de la moindre variation qui puisse exister, ou la prise en compte de la moindre nuance permise par le texte. C'est pourquoi toute représentation *réalisable en pratique* comprendra toujours une part de texte que la machine ne peut traiter que de manière très basique, par exemple en comparant la ressemblance de deux extraits d'un point de vue lexical ou syntaxique. Progresser le long d'un continuum bidirectionnel consiste donc à réduire autant que possible cette part de texte.

1.1.2 Texte libre contre texte semi-structuré

Nous parlons ici de texte *semi-structuré* par opposition à texte *libre*. Les praticiens hospitaliers suivant des protocoles d'examen, les comptes rendus produits sont souvent du texte pré-organisé, parfois obtenu en remplissant des grilles, formulaires ou modèles de documents. Il y a donc un certain niveau d'attente et de prédictabilité quant à la représentation des données trouvées dans ces comptes rendus. Cependant, la normalisation n'est que surfacique et, dans le détail, chaque praticien va avoir sa propre façon de noter ses observations et interprétations. Ainsi, on distinguera dans ces comptes rendus, selon leur granularité, plusieurs niveaux de structuration :

1. XML est par exemple fréquemment utilisé pour représenter des documents très structurés sous forme de paires attributs/valeurs composites. Il en ressort des documents très verbeux qui seraient rendus beaucoup plus compacts dans un format plus spécialisé et fait pour la représentation de ces associations d'attributs/valeurs. Le format POM de description de paquetages logiciels utilisé par Maven, un moteur de production pour Java, en est un exemple typique.

FIGURE 1.1: Espace des représentations et continuums de structuration.



On représente ici certains des modèles de représentations les plus communs sous formes de surfaces dans un espace en deux dimensions. Chaque point de cet espace est une représentation précise. Les dimensions sont la quantité d'information qu'une représentation peut exprimer (telle que jugée par un humain) et l'exploitabilité de cette représentation par un programme informatique. Une troisième caractérisation est la difficulté de la mise en pratique d'une représentation donnée. C'est une fonction définie sur l'espace des représentations, associant ici informellement à chaque point un code couleur allant de vert (très facile) à rouge (irréalisable).

Le segment bleu vertical est un exemple de continuum de structuration bidirectionnel.

— le *niveau de granularité grossier*, composé des grandes sections (et éventuellement sous-sections) qui composent un document ;
— le *niveau de granularité intermédiaire*, composé généralement des intitulés des observations faites (selon la longueur des documents, ce niveau peut être englobé dans le précédent) ;
— le *niveau de granularité fin*, composé des observations et interprétations sur le cas qui se réduisent à quelques mots (la plupart du temps spécifique au domaine) qui se suffisent pour les praticiens à eux-mêmes, et non à des phrases complètes.
— le *niveau de granularité détaillé*, qui localement est souvent ce qui s'apparentera le plus à du texte libre. Il contient toutes les informations que l'auteur du compte rendu a jugé utile de détailler sous forme de phrases complètes.

Les informations présentes à chaque niveau seront caractérisées selon :

— leur *présence*, certaines informations pouvant purement et simplement être éludées lorsque le praticien estime qu'elles ne sont pas pertinentes pour la description du cas médical actuel, et ce même si la collecte de ces informations fait partie du protocole établi ;
— leur *nature* ;
— *l'ordre* dans lequel ces informations sont énoncées.

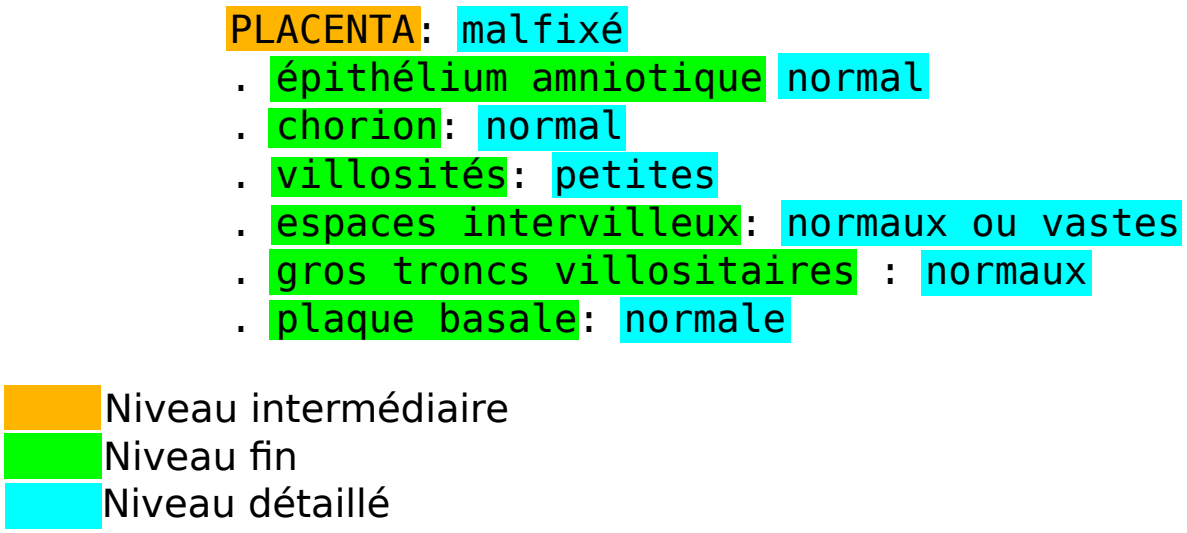
Quand bien même un formulaire ou protocole est utilisé pour produire des comptes rendus médicaux, chacun de ces niveaux de granularité va présenter différents types de variations d'un compte rendu à un autre :

— au *niveau de granularité grossier* la *nature* variera en principe peu, mais si elle le fait cela sera en fonction du temps. En effet, les changements à ce niveau auront lieu lorsque les formulaires seront mis à jour pour refléter l'avancée des connaissances de la spécialité médicale en question. *L'ordre* en revanche pourra changer d'un praticien à l'autre ;
— au *niveau de granularité intermédiaire* la *nature* variera également selon le temps mais aussi selon les praticiens ;
— les niveaux de granularité *fin* et *détaillé* sont bien entendu ceux qui varieront le plus puisque c'est ici que l'information spécifique au cas rencontré sera renseignée. Toutefois, au niveau *fin* ce sont plus souvent des termes normalisés qui seront utilisés, sa variabilité en *nature* et en *ordre* sera donc plus faible qu'au niveau *détaillé* où des phrases différentes pourraient être utilisées pour représenter la même information.

La figure 1.2 montre un exemple de ce que nous entendons par ces niveaux de granularité sur une portion de compte rendu.

Par *données structurées* nous entendons ici tout type de données représentées d'une manière qui soit réductible sans perte d'information à une séquence de paires *(attribut, valeur atomique)*. Ceci comprend toute hiérarchie (ou arborescence) de collections de données usuellement utilisées en programmation : listes, ensembles, tableaux associatifs (ou dictionnaires). Une donnée est dite *totalement* structurée s'il n'est pas possible de concevoir un continuum de structuration où la représentation de cette donnée ne soit pas à l'extrémité haute de ce continuum. Un formulaire où chaque case (identifiée par un attribut)

FIGURE 1.2: Extrait d'un compte rendu d'examen fœtoplacentaire



Dans cet extrait de compte rendu, on a surligné les niveaux de granularité présents

sera renseignée par un nombre ou un symbole pouvant être pris comme un atome (c'est à dire qu'il n'est pas nécessaire ou concevable que ce symbole soit discrétisé davantage) sera donc considéré comme une donnée totalement structurée.

Dans la suite de cette thèse, nous nommerons *analyse structurelle* la tâche qui consiste, dans un document contenant divers niveaux de granularité et donc une *structure latente*, à rendre explicite cette structure, et donc à hiérarchiser toute l'information contenue. À partir d'un texte, on obtiendra donc une arborescence. Chaque nœud de cette arborescence contiendra du texte (plus ou moins long en fonction du niveau de granularité duquel ce nœud est issu), caractérisant le contenu attendu dans les nœuds sous-jacents.

1.1.3 Mots, termes et concepts

Indépendamment de l'analyse structurelle d'un document, il est important d'associer de la manière la plus précise possible un sens à chaque mot qu'il contient. Très souvent, chaque mot pris seul ne véhiculera que peu de sens, car il ne sera qu'une partie d'une expression. Il est en effet impossible sans contexte de différencier localement le sens voulu par la suite de lettres *couvent* dans la phrase

« Les poules du couvent couvent. »

et ce à cause de l'homographie présente. De même, dans la phrase :

« Il lavait au jet d'eau son jet privé. »

on trouve deux homonymes de *jet*. *Jet d'eau* étant une expression suffisamment utilisée telle quelle,

on peut considérer qu'elle forme un terme indivisible, et repérer ce terme dans le texte plutôt que le considérer comme trois mots séparés lève l'ambigüité sur le sens de la première occurence du mot *jet*. Le raisonnement est exactement le même pour *jet privé*.

Il est donc important d'analyser les mots en contexte. Nous pouvons identifier deux contextes pour chaque mot : *local* et *global*. Le contexte *local* d'un mot M est composé des mots qui l'environnent directement (qui sont donc dans la même phrase ou sur la même ligne) et qui impactent le sens voulu de M à cet endroit précis du texte. Le contexte *global* de M sera composé de toute l'information présente ailleurs (la plupart du temps précédemment) dans le document et qui impacte également le sens que l'auteur a voulu donner à M. Le contexte *global* comprend donc toute l'information qui est laissée *implicite* dans le contexte *local*.

La figure 1.2 présentée précédemment donne un exemple d'un extrait de l'observation faite d'un placenta post-accouchement. La quatrième ligne mentionne la taille des villosités (de fines franges qui forment le relief de certaines surfaces de l'organisme). Le contexte *local* ne permet pas de savoir que l'on parle des villosités du placenta en particulier. Cet extrait n'étant qu'une partie d'un compte rendu plus large, cette ligne prise seule ne permet pas de savoir à quel organe cette information est à rattacher. On pourrait avoir tendance à penser qu'il faut remonter jusqu'à la mention `PLACENTA` pour être sûr de savoir de quoi on parle. Ici, ce n'est même pas nécessaire, les lignes environnantes mentionnent l'épithélium *amniotique* (partie du tissu de l'amnios où les cellules sont très resserrées) et les espaces intervilleux (espaces entre les deux plaques du placenta). Les espaces intervilleux sont par exemple dans la SNOMED (*Systematized Nomenclature of Medicine*) classés dans les *arrière-faix*, à savoir les produits de la gestation qui restent dans la matrice après la sortie du fœtus (donc le placenta, le cordon ombilical et les membranes enveloppant le fœtus). Le simple fait de visiter les lignes environnantes permet donc de se placer plus précisément et de donner la priorité au sens *villosités choriales* – qui est le terme préféré dans le thésaurus MeSH (Medical Subject Headlines) pour désigner les villosités placentaires – par opposition à, par exemple, *villosités intestinales*.

Donner un sens à un groupe de mots consiste à trouver sa fermeture sémantique : ramener vers ce groupe de mots toute l'information environnante qui est nécessaire pour pouvoir le comprendre sans ambiguïté. Dans la suite de cette thèse, nous nommerons *analyse sémantique* la tâche qui consiste, dans un document qui contient initialement uniquement des mots indifférenciés, à associer autant que possible à groupe de mots sa fermeture sémantique, rendant donc le plus précisément possible compte du sens de ces mots *dans le contexte dans lequel ils se trouvent*.

Lorsque les documents sont des comptes rendus médicaux, ils utilisent en grande partie un vocabulaire normalisé. Ce vocabulaire est généralement compilé dans des *terminologies* médicales (ou thésauri), qui regroupent et catégorisent tous les termes appartenant à ce vocabulaire. De plus, ces *termes* dénotent toujours dans un texte des *concepts médicaux* précis, qui peuvent être les objets de diverses *relations*, telles que :

— une relation de subsomption (*« Cœur »* est un cas particulier de *« Organe »*),
— une inclusion physique du type *« est contenu dans »* (comme par exemple la relation qui existe entre le concept de *« Cœur »* et celui de *« Cage thoracique »*),

— ou encore n'importe quelle autre relation pertinente pour le domaine.

Ces relations peuvent elles aussi connaître la subsomption : par exemple *« est un os de »* peut être définie comme un cas particulier de la relation *« est une partie de »*. Ces concepts et les relations qui les relient une fois identifiés et formalisés forment ce que l'on appelle des *ontologies* de domaines médicaux. Un *terme* peut être un mot ou une expression, et plusieurs termes différents pourront correspondre à un même *concept*. Le terme général pour désigner les terminologies et les ontologies est celui de *ressource termino-ontologique*, abrégé en RTO.

Tout ceci peut nous aider lors de l'établissement du contexte *local* d'un mot. Comme nous l'avons dit, beaucoup de mots ne peuvent être interprétés seuls, et les thésaurus et ontologies proposent déjà des termes contenant ces mots dans autant de contextes que l'avancée d'un domaine médical donné a pu identifier. Nous pouvons donc tenter de repérer les *termes* du domaine qui sont utilisés tels quels dans un compte rendu médical, et donc de par la même une première partie des *concepts* médicaux impliqués dans ce compte rendu. Ceci s'appelle une *annotation sémantique*, et permet d'obtenir une première ébauche des fermetures sémantiques, qui seront ultimement chacune représentée par un concept.

Nous parlons d'ébauche car une bonne partie des groupes de mots ne pourront en effet pas avoir de fermeture sémantique définitive associée, car il manque la prise en compte du contexte *global*. Le fait que certaines informations soient, au niveau *local*, laissées implicites fait que plusieurs termes potientiels pourront être associés à un groupe de mots donné : les termes qui ne coïncident pas exactement mais sont *lexicalement* les plus similaires à ce groupe de mots.

1.1.4 Imbrication des tâches d'analyses structurelle et sémantique

Pour tenir compte maintenant du contexte *global* des groupes de mots, nous allons utiliser le fait qu'une ontologie de domaine relie entre eux les différents concepts. C'est ici que nous faisons intervenir les notions de *proximité* et *similarité* sémantique (Harispe et al. 2013). Ces deux notions sont toutes deux des métriques qui rendent compte du lien qui existe entre deux concepts. Par exemple, un concept (celui de « véhicule ») qui est légèrement plus global qu'un autre (celui de « voiture ») aura une *similarité* assez forte avec ce dernier, et un concept (« café », en tant que boisson) qui partage une relation sémantique avec un autre (« tasse ») aura une *proximité* assez forte avec lui.

Une fois les deux tâches réalisées, nous avons d'un côté la structure arborescente du document, de l'autre les fermetures sémantiques potentielles pour chaque groupe de mots. Pour affiner ces fermetures sémantiques et réduire l'espace des possibles, nous allons retrouver le contexte *global* dans la structure arborescente précédemment obtenue. Ainsi, certaines informations présentes uniquement au *niveau de granularité grossier* ou niveau *intermédiaire* pourront être répercutées au niveau *fin* ou niveau *détaillé* en comparant la *proximité* des concepts potentiels relevés et en retenant les plus proches.

De cette manière, dans l'exemple donné en figure 1.2, le fait que le concept de *Placenta* (identifiant D010920 dans le thésaurus MeSH), présent au *niveau de granularité intermédiaire*, partage une proximité sémantique plus grande avec *Villosité choriale* (D002824 dans le MeSH) qu'avec *Villosité intestinale*

(15072 dans la FMA – Foundational Model of Anatomy) augmente la probabilité que l'observation `villosités: petites`, présente au niveau *fin*, soit annotée correctement par *Villosités choriales* : `petites`. À moins que le concept plus précis de *« Petite villosité choriale »* puisse être trouvé dans une autre terminologie, la formalisation de cette partie de l'information devra s'arrêter là, et une petite partie de l'observation sera donc laissée en texte libre.

1.2 Un système de raisonnement à partir de cas textuels de fœtopathologie

Comme expliqué en section 1.1.2, nous sommes en présence de documents qui présentent à la fois une structure latente et du texte libre. La suite de ce chapitre détaillera cet état de fait au travers du projet de recherche Accordys dans lequel cette thèse se place.

1.2.1 Contexte d'application : le projet Accordys

Le projet Accordys (Charlet 2013) a démarré en 2012. Il rassemble des praticiens de la gynécologie et de la fœtopathologie (une branche de l'anatomopathologie) ainsi que des chercheurs en ingénierie de la connaissance, traitement automatique du langage naturel et recherche d'informations. Le but de ce projet est de permettre aux médecins de réutiliser le bloc de connaissances présent dans les comptes rendus sur papier d'examens fœtaux stockés dans leur hôpital. Le corpus que nous utilisons en est un sous-ensemble (environ 2000 cas de fœtopathologie), écrit sur environ 10 ans par différents fœtopathologistes. Chaque cas est celui d'une grossesse qui s'est terminée soit par une mort fœtale *in utero* (MFIU) soit par une interruption medicale de grossesse (IMG). Dans le cas d'Accordys, réutiliser cette connaissance signifie donner la capacité aux médecins de retrouver facilement des cas passés qui sont similaires à des cas actuels, afin d'aider au diagnostic et à l'assistance et au suivi des familles.

Ce processus de réutilisation de cas passés pour obtenir de l'information sur les cas présents et donc améliorer les capacités de décision est très similaire à la problématique du domaine de recherche du raisonnement à partir de cas (RàPC, Plaza (1995) Watson et Marir (1994)). Cependant, les besoins des médecins font que le projet Accordys ne cible que deux étapes du cycle habituel de RàPC (Aamodt et Plaza 1994) : la remémoration d'anciens cas et l'enregistrement de nouveaux cas. Les étapes de réutilisation et d'adaptation sont donc laissées à la charge des fœtopathologistes. La raison est que l'étape de remémoration, faite manuellement dans des archives papier, est celle qui est cognitivement lourde et coûteuse en temps, d'où le besoin des médecins d'une nouvelle manière de stocker, retrouver et comparer des cas.

1.2.2 Problèmes posés par le corpus

Ici, nous revenons sur les notions présentées en 1.1.2, et nous les replaçons dans le contexte d'Accordys.

La principale spécificité d'Accordys est son corpus de fœtopathologie. Il s'agit d'un corpus extrait depuis des archives, et qui devrait une fois son extraction terminée comporter le détail d'environ 2000 cas de fœtopathologie, donc 2000 fœtus morts-nés ou bien dont la grossesse a été interrompue médicalement. Le détail de ce corpus sera présenté en 3.1 mais nous pouvons déjà établir ici certaines des problématiques que son traitement pose. Par traitement, nous entendons « comparaison de deux comptes rendus afin d'établir une similarité ».

Le problème principal lié au fond de chaque compte rendu est que le corpus mélange des données observées (donc brutes) et des données inférées par les médecins. Mais rien dans le texte ne permet immédiatement de différencier ces deux types de données, et les connaissances qui ont permis au médecin d'effectuer ces inférences à partir des données brutes ne sont pas mentionnées. Il y a donc ici un problème de pérénité de la validité des données inférées qui peut se poser lors de la réutilisation de dossiers âgés de plus de 20 ans ;

Les problèmes liés à la forme de chaque compte rendu sont les suivants :

— Les informations sont toujours présentées de manière très succinctes. Les documents sont en principe clairs et non ambigus pour un spécialiste du domaine, mais chaque ligne ou phrase peut être ambigüe *si considérée en isolation*. Une structure latente est présente, les informations sont très hiérarchisées et la prise en compte du contexte environnant est donc toujours nécessaire pour pouvoir comparer des comptes rendus ;
— Les comptes rendus ont initialement été rédigés à partir de modèles (grilles, ou templates) où les praticiens devaient en principe remplir les espaces laissés blancs. Cependant ces grilles ne sont pas intégralement suivies à la rédaction de nouveaux comptes rendus. Un certain nombre d'informations sont par exemple présentées avec des marqueurs différents et/ou dans un ordre différent ;
— Chaque document est issu d'un dossier papier numérisé, et traité par une passe de reconnaissance des caractères (OCR). En fonction de l'âge d'un document, cette passe pourra plus ou moins bien fonctionner, et l'on retrouvera un nombre non négligeable de documents contenant des mots impossibles à corriger si on les considère indépendamment les uns des autres (ce que font classiquement les correcteurs orthographiques tels que GNU Aspell [2]) mais qui deviennent relativement clairs dès lors que l'on observe leur contexte local et éventuellement global.

Les problématiques de fond sont en dehors de la portée de ce travail de thèse, mais celles liées à sa forme seront adressées dans la suite de ce document.

2. http://aspell.net

1.2.3 Hypothèse de recherche : intérêt d'une approche hybride structure/contenu

Nous avons établi que dans notre corpus la prise en compte de la structure était nécessaire pour qu'un système puisse tirer parti des données contenues dans deux comptes rendus afin de les comparer. Cette forte structure latente, résultant en plusieurs niveaux de granularité, et cette non-localité textuelle de la donnée (autrement dit ce besoin de regarder au delà du contexte le plus local de chaque élément de donnée pour pouvoir le comparer à d'autres) ne sont pas présentes dans n'importe quel corpus, mais cependant nous défendons l'idée que la prise en compte de la structure est toujours quelque chose de profitable à partir du moment où l'on peut, dans un corpus de documents, observer au moins deux niveaux de granularité structurels.

Concrètement, pour notre corpus nous devrions observer une plus grande pertinence dans les rapprochements effectués si la méthode calculant les similarités inter-documentaires tient compte de la structure latente de ces documents. Plus précisément, le travail présenté dans cette thèse vise à évaluer la viabilité de l'hypothèse suivante :

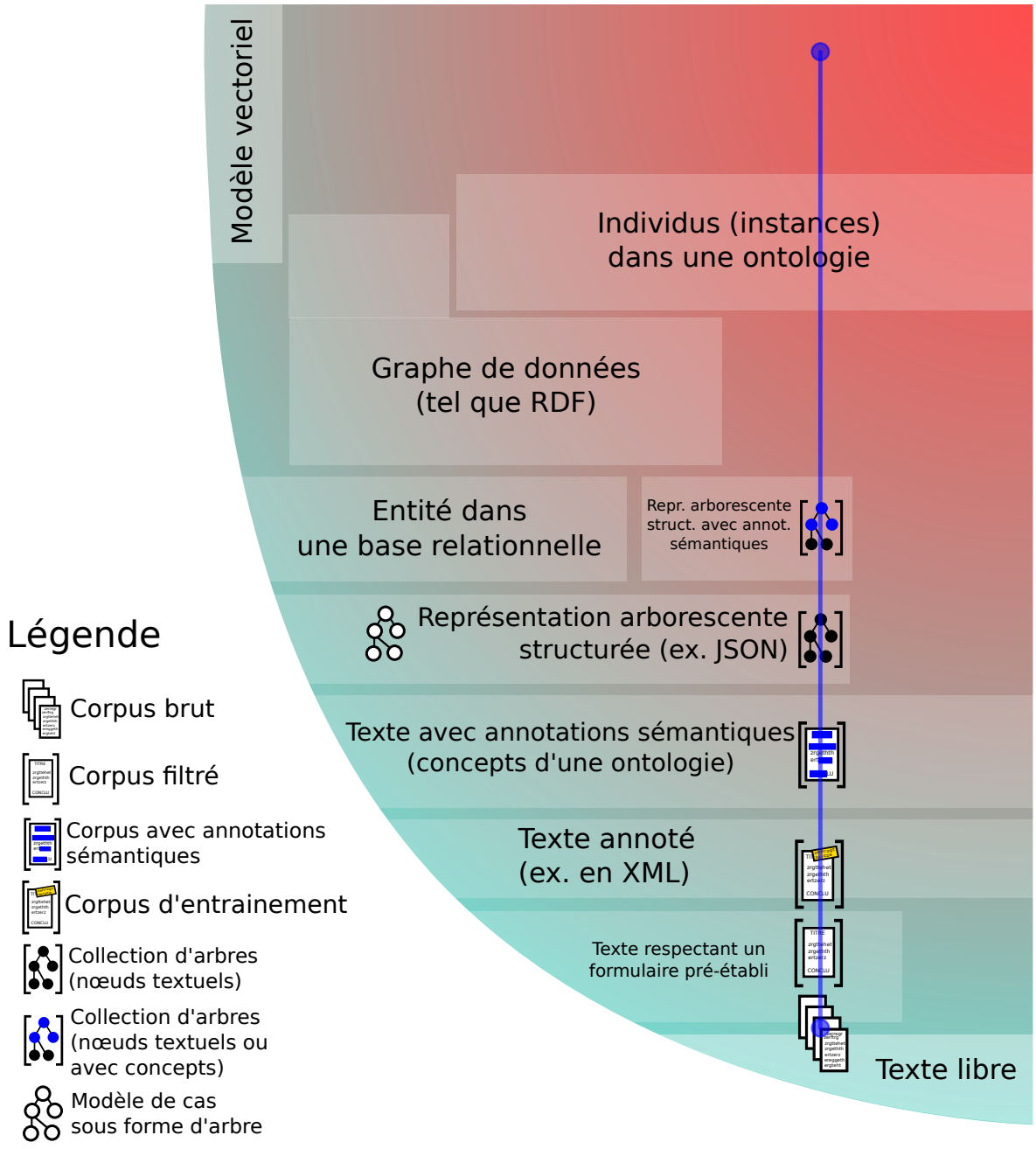
> Dans un corpus documentaire présentant avec un certain marquage, plus ou moins consistant, une hierarchie de divers niveaux de granularité, une approche sémantique hybride opérant (1) sur la structure globale latente du document, (2) sur les éléments sémantiques qui peuvent être identifiés et (3) sur le contenu qui ne peut pas être élevé au delà du texte brut devrait permettre d'obtenir des similarités entre cas plus précises et donc des rapprochements de cas plus pertinents qu'une méthode ne tenant pas compte de la structure globale.

Afin de tester cette hypothèse, nous allons faire subir à notre corpus une série de transformations. La figure 1.3 place les différents états du corpus qui en résulteront sur un continuum de structuration, ainsi que la légende pour désigner ces états que nous utilserons tout au long de ce mémoire.

Ce travail de thèse ne sera en mesure de tester cette hypothèse que sur un seul corpus. Cependant, nous pensons que ce travail est généralisable et présente toujours un intérêt pour des corpus où la structure latente est moins présente, autrement dit où l'on n'observe que deux ou trois niveaux de granularité, mais où cette structure a quand même tendance à se répéter d'un document à un autre. Ceci peut par exemple prendre la forme de sections au niveau *grossier* (sections que l'on peut retrouver d'un document à un autre) et de leur contenu sous forme de paragraphes au niveau *intermédiaire*, sans que ces paragraphes ne présentent de structure latente exploitable qui permettrait de descendre au niveau *fin*. Les raisons à cela sont les suivantes :

— L'indentification des similarités au niveau *grossier* peut servir de première étape pour guider l'indentification des similarités au niveau *intermédiaire*. La segmentation des informations du niveau *intermédiaire*, autrement dit leur replacement dans un contexte précis du niveau *grossier*, permet d'être plus précis dans les comparaisons qui seront effectuées par la suite, afin d'élaguer des comparaisons qui seraient non pertinentes. Cela permet aussi pour des méthodes fonctionnant par apprentissage de constituer non pas une base d'apprentissage unique englobant le contenu de chaque document, mais par exemple une base par section se trouvant au niveau *grossier*. Dans notre cas, au lieu de comparer l'intégralité des textes entre eux, nous pourrions segmenter et

FIGURE 1.3: États résultant de transformations successives du corpus



construire un comparateur dédié à chaque partie des comptes rendus. Ainsi, nous pourrions ne comparer le texte provenant de la partie « Examen macroscopique » qu'avec d'autres examens macroscopiques, la partie « Examen du placenta » qu'avec d'autres examens du placenta, et ainsi de suite ;
— Si les informations contenues au niveau *intermédiaire* sont sujettes à ambigüité, il peut arriver, comme dans notre cas, que la récupération des informations au niveau *grossier* puisse aider à lever ces ambigüités.

1.2.4 Objectif de la thèse dans ce contexte

Ce travail de thèse au sein du projet Accordys est un travail de recherche et développement. La partie recherche s'intéresse à deux choses :

1. appliquer à un domaine médical des technologies classiques de recherche d'information (modèles vectoriels) et du web sémantique (repérage de concepts médicaux généraux) qui ne lui ont pas encore été appliquées ;
2. étant donné la nature du corpus textuel disponible, évaluer l'applicabilité d'une méthode prenant en compte sa mise en forme matérielle et sa structure latente, en la comparant aux méthodes évoquées au point 1.

La partie développement quant à elle a pour but de fournir des outils logiciels supportant le système final d'Accordys d'harmonisation, stockage et mise en correspondance de cas. Ce système doit stocker les cas passés disponibles après les avoir représentés dans un modèle commun (soit vectoriel, soit arborescent, soit un hybride des deux), et répondre à la question « étant donné ce nouveau compte rendu d'examen fœtoplacentaire, quels sont les 10 comptes rendus passés qui lui ressemblent le plus, de par leurs dysmorphies et leur diagnostic ? », une question typique du raisonnement à partir de cas. Plus concrètement, il s'agit d'une boîte à outils avec les fonctionnalités suivantes :

— transformation d'un cas textuel en un arbre ;
— mise en correspondance d'arbres ;
— instanciation d'un modèle de cas prédéfini ;
— application de diverses mesures de similarités sur des arbres pouvant être paramétrées selon la teneur des nœuds : par distance d'édition si les nœuds sont des chaînes de caractères, par comparaison de vecteurs si beaucoup de nœuds contiennent des paragraphes suffisamment longs, par mesures de similarité sémantiques si des concepts ou termes provenant d'une ressource termino-ontologique adéquate ont pu être détectés dans les nœuds.

La fonctionnalité d'annotation sémantique en elle-même n'est pas du ressort de cette thèse, mais sera développée conjointement par le projet Accordys.

Il ne s'agit donc pas ici de produire une application finale, mais des éléments réutilisables qui pourront être combinés selon les besoins applicatifs, l'idée étant de permettre la comparaison de textes contenant une structure, que l'on dispose ou non de ressources terminologiques du domaine duquel les

textes proviennent. Bien entendu, une grande partie des fonctionnalités exposées se baseront sur des technologies déjà existantes sous forme de librairies logicielles, notamment dans l'écosystème Java (pour les technologies liées au web sémantique) mais également dans ceux des langages R et Python (traitement du langage naturel et recherche d'informations). Une grande partie du travail de développement étant lié à la manipulation de structures de données, le plus gros du développement sera réalisé en Clojure, un langage de type Lisp pour la JVM et donc interopérable de façon transparente [3] avec les libraries Java existantes, et qui excelle dans le cadre de la transformation de structures de données (de par sa conception et via des concepts logiciels tels que les *zippers* – qui seront évoqués en section 4.6 – et les références fonctionnelles, ou *lenses*, tous deux facilitant la navigation dans des structures de données et l'application de diverses transformations).

Ces outils pourront donc à terme être réutilisés lors de l'application des résultats d'Accordys à d'autres corpus (en fœtopathologie ou non), ou bien pourront être utilisés pour produire des représentations arborescentes intermédiaires lors de tâches d'extraction de connaissances depuis le corpus d'Accordys.

1.3 Organisation du mémoire

La suite de ce mémoire est composé de quatre parties :

— l'état de l'art qui s'intéressera aux domaines de la recherche d'information (RI), du web sémantique, des mesures de similarités textuelles et sémantiques et de la mise en correspondances de structures de données séquentielles et arborescentes ;
— le matériel qui présentera plus en détail le corpus utilisé pour ce travail de thèse ;
— la méthode proposée et les travaux réalisés ;
— les résultats obtenus à l'issue de ces travaux ;
— les perspectives de ce travail de thèse, pour le projet Accordys comme l'application et l'extension des méthodes envisagées à d'autres domaines.

3. Le terme consacré est *seamlessly*, qui se traduit par « sans couture apparente » mais l'expression française est rarement utilisée.

Chapitre 2

État de l'art

Comme l'explique Charlet (2002), il existe deux options pour qui veut informatiser des documents médicaux (les structurer afin de les stocker dans une base de données et/ou les rendre exploitables par un algorithme) tels qu'un dossier patient tout en conservant les caractéristiques contextuelles de l'information contenue :

— la définition *a priori* des faits médicaux pertinents via une conceptualisation du domaine et la création d'un schéma conceptuel de données ;
— l'utilisation de la forme linguistique et documentaire du dossier consacrée par le praticien ou l'hôpital comme un vecteur de l'information médicale, au même titre que les faits médicaux à proprement parlé que ce dossier relate. Ceci consiste donc à considérer les faits conjointement avec leur contexte (au moins *local*) et avec le formatage avec lequel ils sont présentés.

Par exemple, dans nos comptes rendus, la ligne

```
- mensurations : W 35 CM VC 24 CM PC 22 CM PIED 5,3 CM
```

laisse beaucoup d'informations ambiguës :

— que mesure-t-on ? (information déductible de ce qui se trouve précédemment dans le texte)
— à quoi correspondent les paramètres `W`, `VC` et `PC` mesurés ? (information déductible uniquement si l'on connaît le domaine, voire mêmes les habitudes précises des médecins ayant écrit le document)

Le but de cette thèse est de montrer que ces deux approches peuvent être utilisées conjointement, afin de bénéficier des apports d'une conceptualisation – quand bien même cette dernière ne soit que partielle – faite à priori tout en tirant profit de la forme donnée au texte. Ainsi, dans cet état de l'art nous présentons les travaux ayant trait à ces deux axes. Nous terminons par les méthodes existantes pour apparier et aligner des structures de données, un élément clef de la méthode que nous proposons par la suite.

2.1 Méthodes de mesure de similarité basées sur le texte

Dans cette partie, nous nous intéressons aux méthodes permettant d'utiliser uniquement les informations contenues dans des textes pour établir des mesures de similarité entre eux. Nous n'extrayons donc pas à proprement parler d'information du texte, l'interprétation finale étant laissée à l'utilisateur. Ainsi pour tout corpus C de documents on sait juste que pour tout $D_1 \in C$, $info(D_1)$ existe et est l'ensemble des informations contenues dans D_1 et qu'il existe un document $D_2 \in C$ tel que $D_2 = \arg\max_{D \in C \setminus D_1} |info(D_1) \cap info(D)|$, autrement dit que D_2 est le document de C qui maximise la quantité d'information en commun avec D_1. On dit alors que D_2 est le document le plus similaire à D_1 dans C.

L'intégralité de l'information expliquée et impliquée par le texte n'étant pas directement accessible à un programme informatique, il faut donc trouver des approximations de la fonction $info(D)$ qui se basent uniquement sur l'information explicite dans C, c'est à dire les mots que C contient.

2.1.1 Modèles vectoriels

Le but d'une modélisation vectorielle est de représenter chaque entité sur laquelle des comparaisons vont devoir êtres effectuées de manière géométrique. Il s'agit de représenter des unités lexicales sous forme de vecteur. L'application que l'on cherche à réaliser impactera le choix de l'unité (chaque document ? Chaque paragraphe ? Ou même chaque mot ?) qui deviendra un vecteur et de l'unité ou groupe d'unités lexicales desquelles on dérivera les dimensions de ces vecteurs. Dans le cas de notre corpus, une grande partie des mots se regroupent en termes consacrés spécifiques au domaine. Ces termes sont à prendre tels quels, de manière indivisible, c'est pourquoi il peut être intéressant de repérer les termes du domaine et de les utiliser comme dimensions, conjointement avec les mots qui ne sont pas regroupables. Par exemple dans la figure 2.1 les portions surlignées sont des termes fréquents dans les domaines de la fœtopathologie, de l'anatomopathologie ou de l'anatomie en général, qui devraient idéalement être pris tels quels.

FIGURE 2.1: Termes de fœtopathologie dans un compte rendu

2. RADIOGRAPHIES DU SQUELETTE COMPLET FACE ET PROFIL :
Les clichés montrent un squelette normal en dehors de la microcrânie déjà notée.
La présence des points d'ossification calcanéens, et la mesure des os longs sont compatibles avec un âge gestationnel de 26 semaines

Les techniques propres à la modélisation vectorielle restent les mêmes quels que soient les choix d'unités et de dimensions à modéliser et viennent de l'algèbre linéaire. Par exemple, la mesure TF-IDF pose que les documents deviennent des vecteurs, et les mots (termes) des dimensions (LANDAUER et DUMAIS 1997). L'entièreté du corpus est donc une matrice $N_D \times N_M$, N_D étant le nombre de documents du corpus et N_M le nombre total de mots différents utilisés dans ce corpus, et les vecteurs en question

sont les lignes de cette matrice. On a donc ainsi fait deux choses : on a caractérisé géométriquement les documents en fonction des mots qu'ils contiennent (si l'on regarde les lignes de la matrice) et, transposément, caractérisé les mots en fonction des documents dans lesquels ils apparaissent (si l'on regarde les colonnes de la matrice). Comme l'expliquent Landauer, McNamara et al. (2013), la mesure TF-IDF permet d'identifier les mots dans une collection de documents qui sont utiles pour déterminer le thème abordé par chaque document. En d'autres termes, si un mot apparaît dans un « faible » nombre de documents mais apparaît de nombreuses fois dans ceux-ci, alors ce mot aura un poids élevé, car on considère qu'il contribue au thème.

En modélisation vectorielle, $info(D)$ est donc un vecteur dans $(\mathbb{R}^+)^M$. La similarité est obtenue par l'angle qui sépare les deux vecteurs associés à deux documents. On utilise en général le cosinus de cet angle qui est plus direct à calculer et qui rend compte de la même information :

$$similarité(D_1, D_2) = \frac{info(D_1) \cdot info(D_2)}{||info(D_1)|| \times ||info(D_2)||}$$

2.1.2 Analyse distributionnelle

En analyse distributionnelle, on cherchera à représenter les termes par l'ensemble des autres termes qui *co-occurrent* avec eux. L'idée comme quoi chaque mot peut être caractérisé par ceux qui l'environnent a été popularisée par Firth (1957). Ainsi, deux mots qui apparaissent fréquemment dans le même contexte seront considérés comme proches sémantiquement (Harris 1954). Ceci s'appelle l'hypothèse distributionnelle, qui consiste à poser que les comportements distributionnels sont suffisants pour étudier l'intégralité des propriétés du langage et que les phénomènes linguistiques (tels que les variations de sens) peuvent être complètement expliqués par la méthodologie distributionnelle. En effet, Harris supposait qu'en tant que science, la linguistique ne devait pas se préoccuper de facteurs extra-linguistiques, et que donc le sens des mots devait être déductible de l'analyse distributionnelle. Les variations de sens seront donc provoquées par les variations dans la distribution. On sait depuis les travaux de Noam Chomsky que ceci est une approximation, mais de ce fait, la méthodologie distributionnelle permet de quantifier les différences de sens entre les entités linguistiques, donc d'établir la similarité sémantique entre les mots.

Fréquemment, les mots ne seront pas traités tels qu'ils apparaissent dans le corpus par l'analyseur distributionnel. Les prétraitements les plus couramment réalisés (conjointement ou non) sont les suivants :

- **le filtrage des mots vides,** qui va retirer du texte tous les mots (*« stopwords »*) appartenant à une liste noire (déterminants, prépositions, mots jugés trompeurs…) et dont on estime qu'ils n'apportent pas d'information dans le cas d'usage considéré. Ces listes sont souvent appelées *stoplists* ;
- **la lemmatisation,** qui va remplacer chaque mot par son lemme aide à réduire le nombre de variantes de chaque mot présentes dans le corpus, et donc de réduire la dispersion des données ;

l'annotation morphosyntaxique (*part-of-speech tagging*), qui va adjoindre à chaque mot sa catégorie morphosyntaxique. Ainsi, en fonction de son emplacement, « rire » serait par exemple remplacé par `rire_NOM` ou `rire_VERBE`. Ceci provoquera l'effet inverse de la lemmatisation, puisqu'on va artificiellement augmenter le nombre de variantes existantes, ce qui peut ainsi diminuer la performance de l'analyse distributionnelle dans les cas où l'on prend des contextes assez larges pour chaque mot (Sahlgren 2006). Cette annotation est cependant nécessaire si l'on veut filtrer le corpus pour ne garder que certaines catégories (uniquement les noms et les adjectifs, par exemple). L'annotation morphosyntaxique peut aussi être la première étape à l'établissement du contexte de chaque mot. En effet, ce contexte peut-être obtenu par fenêtre graphique (par exemple « tous les mots à une distance de 4 mots ou moins du mot considéré font partie de son contexte », un cas dans lequel on a une fenêtre graphique d'une largeur de 9 mots) ou bien de manière syntaxique, en construisant donc le graphe de dépendance de chaque phrase du texte. Ceci nécessite bien entendu que le texte soit grammaticalement correct dans son intégralité, un postulat mis en défaut dans bon nombre de corpus de spécialité médicale (tel que celui dont est extrait l'exemple en figure 1.2).

la racinisation (ou *stemming*) il s'agit de remplacer chaque mot par son radical. Plus simple que la lemmatisation, cela peut la remplacer lorsque les catégories morphosyntaxiques des mots ne sont pas déductibles, par exemple parce que le texte est réalisé avec des économies linguistiques (Dubois et al. 1994), comme c'est le cas pour Accordys ;

À partir des années 1990, l'analyse distributionnelle sera notamment appliquée au domaine médical afin de construire des thésaurus de termes de domaines médicaux particuliers. Grefenstette (1992) le fera pour un corpus en anglais d'actes thérapeutiques en oncologie et Bouaud et al. (2000) pour le français. Pour l'établissement des contextes des mots, les auteurs privilégient plutôt des fenêtres graphiques de taille assez restreinte (deux mots de chaque côté dans le cas de Généreux et al. (2013) par exemple).

L'analyse distributionnelle va donc rapprocher des termes (mots ou groupes de mots) en fonction de ce qui les environne dans le texte. Il est ainsi possible de représenter chaque mot par un vecteur, dont les dimensions seront souvent appelées *contextes*, puisqu'elles représentent les contextes dans lesquelles les termes apparaissent. Avec ce mode de représentation on utilisera une similarité cosinus pour évaluer la similarité du sens de ces mots. Une autre mesure de similarité utilisée fréquemment pour les textes de spécialité est l'indice de Jaccard, qui va comparer le nombre de mots communs aux contextes de deux mots w_1 et w_2 par rapport au total des mots différents présents dans leurs contextes :

$$S(w_1, w_2) = \frac{|contextes(w_1) \cap contextes(w_2)|}{|contextes(w_1) \cup contextes(w_2)|}$$

En général l'analyse distributionnelle est souvent appliquée à des gros corpus (1 milliard de mots voire plus), mais il est toutefois nécessaire de faire cette analyse avec des corpus beaucoup plus petits en taille (1 million de mots ou moins) et c'est souvent le cas de corpus de spécialités. Récemment, Périnet et Hamon (2014) se sont intéressés à l'utilisation de la méthodologie distributionnelle sur des corpus de ce type, grâce à la généralisation des contextes, ce qui permet de réduire la dispersion des données dans ce type de corpus.

2.1.3 *Topic modeling*

Le *topic modeling* a pour but de pallier la dispersion des données habituellement observée en analyse distributionnelle : une grande partie des mots n'apparaissant que dans peu de contextes, la matrice mots/contextes sera creuse. Le *topic modeling* est donc une méthode de réduction de dimensionalité : à partir d'un nombre D_1 de dimensions (*contextes*) dans la matrice originale, on va obtenir D_2 dimensions (*thèmes*) ($D_2 < D_1$) dans une nouvelle matrice, de telle sorte que chaque nouvelle dimension soit une combinaison linéaire des contextes originaux. Pour une unité lexicale donnée (mot ou document) représentée initialement avec un vecteur de contextes $\vec{c}$ on va donc avoir $\vec{d} = M \cdot \vec{c}$, où $\vec{d}$ est le vecteur des *thèmes* et M est une matrice de taille $D_2 \times D_1$. Le but du topic modeling est donc de trouver cette matrice M. Plus concrètement, les contextes qui se retrouvent fréquemment autour des mêmes mots vont avoir tendance à être « regroupés » dans un thème.

L'analyse sémantique latente (LSA ou LSI) (Landauer et Dumais 1997) (Landauer, McNamara et al. 2013) calcule cette matrice M en réalisant une décomposition en valeurs singulières (SVD) de la matrice entités/contextes. Le nombre de thèmes doit être fixé à l'avance. L'allocation latente de Dirichlet (LDA) (Blei et al. 2003) utilise un modèle graphique probabiliste génératif. Elle étend la LSA de manière probabiliste où chaque document est représenté comme un mélange aléatoire des thèmes latents et où chaque thème est caractérisé par une distribution de mots.

2.2 Structuration et extraction d'informations venant de corpus de texte

Nous nous intéresserons ici d'abord à deux choses : les structures énumératives et l'ambigüité dans un corpus de texte. Nous présenterons ensuite le domaine du raisonnement à partir de textes (une branche du RàPC), puis de l'extraction d'information depuis des comptes rendus médicaux.

2.2.1 Traitement des structures énumératives

Comme l'explique Virbel (1999) :

> Énumérer mobilise deux actes : un acte mental d'identification des éléments d'une réalité du monde dont on vise un recensement, et où on établit une relation d'égalité d'importance par rapport au motif de recensement ; et un acte textuel qui consiste à transposer textuellement la coénumérabilité des entités recensées, par la coénumérabilité des segments linguistiques qui les décrivent.

L'acte d'énumération consiste donc à regrouper des éléments indépendants sous un même critère d'homogénéité. Ainsi, dans l'exemple de la figure 1.2, le critère d'homogénéité des informations énumérées est le fait qu'ils présentent tous une observation faite sur une sous-partie d'un placenta.

Une structure énumérative (SE) se compose d'une énumération (succession) d'*items*, possiblement précédée d'une *amorce* et terminée par une *conclusion* (ou *clôture*) (Luc 2001). Ho-Dac et al. (2010) appelle *enumeraThème* le critère d'homogénéité qui relie les items qui se trouvent au sein d'une même énumération. L'énuméraThème se trouve donc un niveau hiérarchique au dessus des items. Ho-Dac et al. (2010) présente aussi l'approche descendante poursuivie lors du projet ANNODIS [1] (Afantenos et al. 2012), projet dont le but était la constitution d'un corpus de documents de genres, domaines et longueurs variés où les structures du discours ont été annotées. Cette approche part du principe que prendre en compte les structures de haut niveau (*niveau de granularité grossier*) aide à la compréhension locale du texte. Ce travail distingue quatre dimensions pour décrire chaque SE :

— sa taille (le nombre de mots qu'elle contient),
— sa cardinalité (le nombre d'items),
— sa composition (présence d'éléments et marqueurs optionnels),
— son niveau de granularité (son emplacement dans la structure du document).

Les SE seront regroupées en 4 catégories, de type 1 (niveau de granularité le plus grossier) jusqu'à 4 (le plus fin). Ces structures peuvent être réalisées soit d'une façon purement linguistique (en utilisant notamment des connecteurs tels que *et*, *ou*, *premièrement*, *d'autre part*, *ensuite*, etc.) ou bien également avec une *mise en forme matérielle* (MFM) (Virbel et al. 2005) autrement dit en utilisant des marqueurs typographiques (des puces ou des tirets par exemple) qui deviennent vecteurs de sémantique. Une SE est dite *parallèle* lorsqu'elle est *homogène* (les items énumérés sont visuellement structurés de la même façon), *paradigmatique* (les items n'ont pas de relations de dépendance syntaxique ou rhétorique les uns avec les autres) et *isolée* (son amorce et sa clôture ne contiennent pas d'autre structure énumérative) (Luc 2001). Ainsi, dans une SE parallèle, c'est la même relation sémantique qui relie chaque item à l'amorce de la SE.

Fauconnier et al. (2013) réalise l'analyse des structures énumératives parallèles avec MFM d'un corpus pour en extraire des connaissances qui seront utilisées pour construire et enrichir des ontologies. L'idée est donc ici de trouver les relations sémantiques qui sont exprimées par les caractères typo-dispositionnels et non par des formulations strictement linguistiques. L'apprentissage est réalisé ici avec un modèle statistique discriminatif (une régression logistique multinomiale, ou classifieur d'entropie maximale), entraîné sur un ensemble de triplets *(amorce, relation, item)* obtenu par une annotation manuelle du corpus. Les types (classes) de relations possibles sont par exemple « hyperonymie » ou « synonymie ». Cette étude montre l'intérêt de la prise en compte des SE et de la MFM par rapport à une simple étude par trigrammes. Et au final, les résultats suggèrent que la classification des SE est meilleure lorsqu'on se limite à l'identification de la relation entre l'amorce et le premier item. L'applicabilité des CRF à ce même problème (pour remplacer le classifieur d'entropie maximale) est également évoquée.

1. http://redac.univ-tlse2.fr/corpus/annodis

2.2.2 Segmentation et étiquetage de textes et séquences

La segmentation correspond à une première forme de structuration à un unique niveau de granularité (autrement dit en mettant à plat les quatre niveaux définis précédemment). On peut définir plusieurs types de segmentation :

— Segmentation en unités lexicales, telles que les mots, phrases, paragraphes ou certains types de structures énumératives (Ho-Dac et al. 2010),
— Segmentation en thèmes (*topics*).

La phase d'étiquetage (typage) des segments identifiés est parfois considérée séparément de la segmentation en elle-même. En effet, pour savoir où un segment de texte se termine et où un autre commence, il peut être important d'avoir défini la nature du segment courant, et peut-être des segments qui l'environnent. Le repérage des mots du texte (*tokenisation*) est par exemple fréquemment suivi par la détermination de leur nature grâce à un étiquetage morpho-syntaxique (voir section 2.1.2). Le regroupement des phrases ou paragraphe d'un texte en segments thématiquement liés n'est vraiment utile que si l'on peut caractériser d'une manière ou d'une autre le thème de chaque segment. Ainsi et au vu de nos besoins, nous considérerons segmentation et étiquetage comme un tout indissociable.

Hearst et Plaunt (1993) et Kozima (1993), à une époque où les documents complets – et plus seulement des abstracts ou résumés – commencent à être de plus en plus répandus sur le web, posent la question de comment dans un texte composé de multiples paragraphes il est possible de retrouver les thèmes et sous-thèmes évoqués. Kozima (1993) définit un indicateur appelé profil de cohésion lexical (LCP) qui localise les frontières des segments dans un texte. Un segment contient des mots qui sont liés ensemble par des relations de cohésion, évaluées en mesurant la similarité des mots dans un réseau sémantique. Le LCP évalue la similarité mutuelle des mots dans un même segment, et les auteurs montrent qu'il est fréquemment corrélé avec le jugement d'un humain qui aurait à effectuer le même découpage thématique.

Hearst et Plaunt (1993) quant à eux présentent une méthode de structuration dans le but de servir une recherche d'information où l'utilisateur pourrait effectuer une recherche non pas seulement sur les documents complets mais aussi sur des sous-parties ou des sous-thèmes. Hearst utilise la localité des mots dans le document comme un indicateur de la thématisation de ses sous-parties. Ainsi, un mot se répétant de nombreuses fois dans des paragraphes proches, mais quasiment absent dans le reste du texte, sera un indicateur assez fort qu'un thème est partagé par ces paragraphes et renseignera sur la nature de ce thème. L'intuition, à l'échelle des paragraphes, est donc assez similaire à celle à l'origine de TF-IDF et de l'analyse sémantique latente (Deerwester et al. 1990) (Landauer et Dumais 1997) qui, elle, est généralement appliquée à l'échelle de documents pour les comparer.

Salton et al. (1997) a étudié la structuration automatique de documents dans le cadre de la création automatique de résumés. Leur approche procède par création d'hyperliens *intra* documentaires. Selon le motif des liens tissés, on peut caractériser la structuration du texte, et ainsi extraire des passages pour produire un résumé.

Lafferty et al. (2001) détaillent les *Conditional Random Fields* (CRF). Il s'agit d'un modèle statistique graphique (au même titre que les modèles de Markov cachés, les réseaux bayésiens ou les réseaux neuronaux) où les différentes variables stochastiques modélisées sont reliées par des relations de dépendance non orientées. Les variables stochastiques sont deux deux types, les variables d'entrée (également appelées caractéristiques ou *features*) et les variables de sortie (également appelées étiquettes ou *labels*). X est le vecteur des valeurs des variables d'entrée, et Y celui des variables de sortie. Les CRF cherchent à modéliser la distribution de probabilité conditionnelle $P(Y|X)$. Ceci fait des CRF un modèle statistique *discriminant*, puisqu'il peut être utilisé, après entraînement, pour segmenter, étiqueter et classifier des séquences mais qu'il ne peut pas être utilisé pour en générer de nouvelles. Depuis Lafferty et al. (2001), les CRF font partie du paysage des outils classiques de la segmentation par apprentissage, notamment en traitement d'image ou en traitement automatique de la langue.

Nous sommes dans notre cas intéressés par une autre forme de segmentation : la segmentation en unités de structure. Si notre document est un arbre, nous voulons repérer et étiqueter ses noeuds.

2.2.3 Désambiguïsation d'un corpus

Désambigüiser une unité lexicale consiste à trouver l'ensemble des sens possibles et à choisir, en fonction du contexte, celui qui semble le plus probable. Un des premiers modèles de représentation des sens d'un mot, appelé analyse sémique ou componentielle, donne une bonne idée des notions manipulées lors de cette tâche. En analyse sémique une unité lexicale (mot par exemple) a plusieurs *sèmes*. Un sème (ou *trait sémantique*) est un composant distinctif de cette unité lexicale qui sera soit positif (l'unité a ce composant), soit négatif (l'unité n'a pas ce composant) soit sans objet (l'unité est orthogonale à ce composant). Ce dernier cas signifie que caractériser en fonction de ce composant n'aurait pas de sens, et serait une erreur catégorielle. Par exemple, la phrase *« La couleur bleue est plus lourde que la couleur rouge. »* n'a pas de sens dans le langage courant : les couleurs ne *possèdent* pas de masse, elles ne peuvent donc avoir ni la propriété d'être lourdes, ni sa négation, puisque que dire d'une couleur qu'elle est « légère » n'a pas plus de sens. Couleur et masse sont des notions orthogonales dans le langage courant, bien qu'un domaine artistique pourrait par exemple utiliser une notion de masse pour comparer des couleurs.

Prenons l'exemple de la phrase « Le canard était posé sur la table. », il y a ambigüité sur le sens de l'unité lexicale « canard ». Tellier (2010) montre un exemple de décomposition sémique de cette dernière (voir figure 2.2). La relation *« être posé sur »* liant le canard et la table dans la phrase implique qu'ici, le « canard » possède la propriété d'être matériel. On peut donc éliminer la branche « (-) matériel ». En revanche, nous ne pouvons pas préciser plus le sens simplement avec cette phrase, il faudrait qu'elle soit plongée dans un contexte plus global pour vérifier si d'autres sèmes peuvent être précisés. L'ambiguïté entre « canard, animal » [(+) animé] et « canard, journal » [(-) animé] persiste donc. Il serait toutefois possible d'augmenter la probabilité du sème « (-) animé » (et donc de diminuer celle de « (+) animé ») du fait de la présence du verbe « poser » qui est plus souvent utilisé pour des objets inanimés.

Toute la difficulté en désambiguïsation est donc de représenter et repérer les propriétés latentes dans

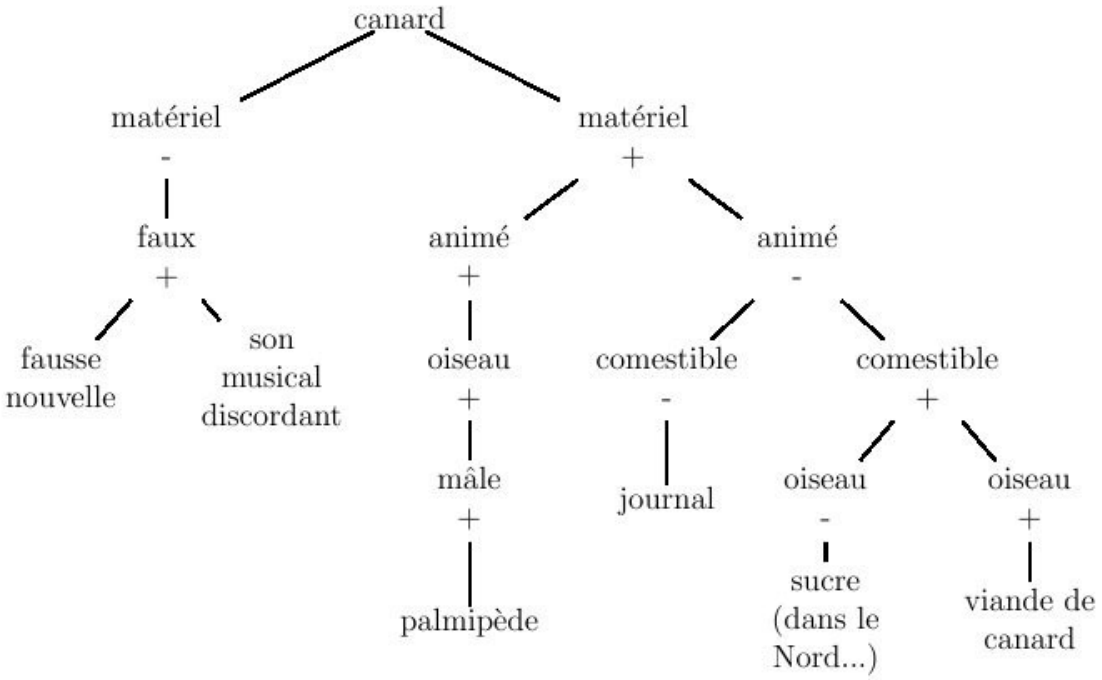


FIGURE 2.2: Arbre de désambigüisation contenant les sèmes associés au mot "canard"

le texte de chacune de ses unités lexicales ambigües. Des études récentes tirent parti des avancements en ingénierie des connaissances, notamment des bases de connaissances déjà construites (réseaux sémantiques et ontologies) qui représentent des concepts associés à leurs propriétés. Chaque concept ayant une signification unique et formalisée, trouver quel concept se rattache à un syntagme présent dans un corpus désambiguïse le syntagme en question. JIMENO YEPES et ARONSON (2012) observe que ce sont les méthodes à base d'apprentissage supervisé qui produisent les meilleures performances lors d'une tâche de désambiguïsation. Cependant, la réalisation d'un corpus annoté manuellement fournissant des données d'entrainement pour chaque sens de chaque mot ambigu du domaine considéré étant une lourde tâche, les auteurs proposent une méthode combinant un graphe de connaissance (le métathésaurus UMLS, pour Unified Medical Language System [2]) avec de l'apprentissage non-supervisé (un K-Means effectuant le clustering de vecteurs obtenus par analyse distributionnelle). Les auteurs expérimentent cette méthode sur un corpus de la communauté biomédicale extrait de Medline (JIMENO-YEPES et al. 2011). Pour chaque mot ambigu dans ce corpus, on retrouve dans l'UMLS les concepts candidats (autrement dit les sens potentiels) et leurs définitions associées. On représente chaque définition dans un espace vectoriel (cf. 2.1.1) où les dimensions sont les mots des définitions. Les auteurs nomment cette approche MRD (*machine readable dictionary*). Ensuite, on extrait pour chaque mot ambigu son contexte dans le texte que l'on représente dans le même espace vectoriel. Afin de ne pas être trop dépendant des mots utilisés dans la définition de chaque concept et de tenir compte du fait que les contextes des mots ambigus contiendront très souvent des mots qui ne sont pas dans ces définitions, le résultat final est obtenu après itération d'un algorithme de K-Means, où les valeurs initiales des

2. https://www.nlm.nih.gov/research/umls

barycentres sont les vecteurs correspondant aux concepts candidats et où la proximité entre individus (les vecteurs issus des mots ambigus) et centroïdes est évaluée avec la similarité cosinus. Jimeno Yepes et Aronson (2012) observent une amélioration des performances par rapport à l'approche MRD utilisée seule, sans toutefois atteindre celles d'une méthode totalement supervisée.

Toujours dans la même optique, Prokofyev et al. (2013) propose une méthode de désambiguïsation semi-supervisée basée sur un graphe de connaissances (*« background knowledge graph »*) conjointement à des statistiques extraites par apprentissage automatique d'un corpus, ceci dans le cadre du croisement de publications venant de différentes sous-communautés de chercheurs. Des expérimentations sont menées sur deux corpus : celui de Jimeno-Yepes et al. (2011) cité plus haut et un autre issu de la communauté des chercheurs en physique (venant du portail ScienceWISE). Les annotations réalisées sur le corpus sont réalisées à partir du graphe, il n'y a donc pas d'annotation manuelle à proprement parler.

Tchechmedjiev (2012) propose une revue de méthodes d'IA classiques (recuit simulé, approche par colonie de fourmis, algorithmes évolutionnistes...) combinées avec des mesures de similarités basées sur des connaissances pour la désambiguïsation.

2.2.4 Raisonnement à partir de textes (TCBR)

Le raisonnement à partir de cas opère généralement sur des bases de cas représentés de manière structurée, dans un modèle de données établi. Si l'on cherche maintenant à opérer sur du texte libre, on peut déjà dégager deux grandes stratégies : transformer le texte en un format lisible par la machine afin de se ramener à un cas où les techniques de RàPC classiques sont utilisables (tel que R. Weber et al. (1998), qui instancie un modèle de cas à partir de documents textuels) ou garder la représentation textuelle et adapter les techniques de RàPC pour qu'elles puissent le manipuler. Ces deux approches sont identifiées dans la littérature comme étant du *textual case-based reasoning* (TCBR) (R. O. Weber et al. 2005), mais c'est la deuxième stratégie qui nous intéressera ici. L'un des premiers systèmes de TCBR produits utilisant directement le texte comme cas est FAQFinder (Burke et al. 1997), qui l'applique à la recherche de documents en langue générale (autrement dit, le système ne vise pas un domaine particulier). Le but est de répondre à la question d'un utilisateur en retrouvant dans les foires aux questions de *news groups* USENET celles qui seraient susceptibles de contenir une ou plusieurs questions similaires. WordNet [3] est utilisé pour obtenir les similarités sémantiques entre les mots des questions, et le système procède en deux phases : d'abord cibler les *news groups* les plus approchants à l'aide d'une simple analyse des mots-clefs de la question, puis une recherche dans les FAQ de ce *news group* pour trouver la question la plus similaire.

Pour être plus précis sur les attendus d'un système de TCBR, Richter et R. Weber (2013b) définissent deux vues lorsqu'on manipule des documents textuels :

Vue orientée document Seulement les métadonnées des documents sont gérées par le système ;

3. http://www.cogsci.princeton.edu/~wn

Vue orientée contenu Tout le contenu des documents est accédé par le système de RàPC.

Ils précisent que le TCBR concerne uniquement la vue orientée contenu. Dans ce contexte, la similarité entre un nouveau problème et un cas textuel passé doit refléter à quel point le texte contient des connaissances intéressantes pour résoudre ce nouveau problème. Comme l'explique l'un des premiers articles dédiés au TCBR (Lenz 1998), ce domaine doit composer avec des problèmes qui sont plus traditionnellement ceux de la communauté de la recherche d'information (RI). Concernant la comparaison des cas, les techniques utilisées en TCBR donc vont être proches de celles présentées précédemment. On pourra utiliser donc des modèles vectoriels pour représenter les cas et une similarité cosinus comme mesure, comparer les mots que les cas contiennent en comparant leurs bi- ou trigrammes (les séquences de deux ou trois caractères contigus qu'ils contiennent), réaliser une analyse distributionnelle des textes ou du topic modelling, etc.

Lenz (1998) définit les sources de connaissances à utiliser par un système de TCBR dans le cadre de corpus de spécialité. La base de connaissances complète sera segmentée en couche (*layers*), et chaque couche correspondra à des procédures d'analyse spécifiques à appliquer. Le marqueur *(E)* signifie que l'auteur entend l'intervention manuelle d'un expert du domaine pour constituer une source de connaissance. Voici les couches de la plus basse à la plus haute :

Couche mot-clef : composée d'un dictionnaire de mots-clefs *(E)* généraux et spécifiques au domaine et d'un analyseur pour les reconnaître dans les textes. L'obtention de mots-clefs à mettre dans ce dictionnaire peut également se faire par analyse des fréquences des mots dans le corpus ;

Couche phrase : un autre analyseur reconnaît les phrases types spécifiques au domaine (également depuis un dictionnaire *(E)*). Il doit pouvoir reconnaître une phrase type séparée en deux dans un document par l'introduction d'une proposition ; Des sources de connaissances (documentations, manuels d'utilisation par exemple) doivent être utilisées pour repérer les phrases types à inclure au dictionnaire ;

Couche thésaurus : cette couche doit relier différents mots-clefs (généraux et spécifiques au domaine) les uns aux autres (relations de synonymie, d'hypéronymie, etc.) ; WordNet fournit ce type de thésaurus en langue générale. Cette couche établit donc les similarités linguistiques entre les mots ;

Couche glossaire : cette couche donne les similarités au sens du domaine entre les mots-clefs et les phrases types *(E)* ;

Couche des valeurs des caractéristiques (*features*) : cette couche établit les grandeurs manipulées par le domaine (longueurs, poids...), ainsi que les caractéristiques fréquemment utilisées pour décrire les objets du domaine avec les valeurs qualitatives qui peuvent être affectées à ces caractéristiques. Cette couche définit aussi les mesures de similarités *(E)* qui compareront ces valeurs de caractéristiques entre deux documents ;

Couche structure de domaine : en fonction des valeurs de caractéristiques repérées dans un document à la couche précédente, il est parfois possible de classifier le document. Cette couche scinde donc les différents documents d'un même domaine en différentes catégories, ce qui peut accélérer les rapprochements de cas faits par la suite ;

Couche extraction d'information : dans le cas où les documents contiennent une section spéciale où l'information apparaît déjà structurée (liste de paires *(attribut, valeur)* par exemple), cette couche a pour but de la repérer et d'en extraire les valeurs.

L'un des intérêts de cette séparation de la base de connaissances utilisées est sa modularité. Par exemple, la façon de concevoir les similarités peut changer entre deux applications même si le domaine reste identique, et dans ce cas la seule couche à modifier sera la couche glossaire. Ainsi la plupart des couches sont réutilisables d'une application à l'autre, voire même d'un domaine à l'autre. Les auteurs montrent donc divers exemples d'applications de TCBR avec pour chacune les spécificités ayant trait à la construction de chaque couche.

Dans le domaine clinique, Arnold et al. (2010) proposent une méthode de comparaison de dossiers patients basée sur la comparaison de leurs thèmes (*topics*) latents (notion évoquée en section 2.1.3), obtenus par allocation de Dirichlet latente.

Pour finir, Recio-Garcia et al. (2007) présentent l'utilisation du TCBR dans le cadre de l'outil jCOLIBRI, une plateforme servant à configurer et générer des systèmes de raisonnement à partir de cas.

2.2.5 Traitement spécifique des comptes rendus médicaux

Comme nous l'avons expliqué en 1.1.2, la reconstruction du contexte des informations passe notamment par le repérage de la structure du document. Diverses méthodes pour segmenter et repérer des éléments de structure dans un texte ont été évoquées en 2.2.2. Nous allons maintenant nous intéresser aux applications de méthodes de ce type spécifiquement à des corpus de spécialité médicale. Des efforts de structuration automatique de comptes rendus médicaux ont été réalisés. Taira et al. (2001) s'y sont intéressés dans le cas de comptes rendus de radiologie, cependant leur étude vise à structurer une partie du document, il ne s'agit donc pas d'une approche holistique. L'approche se base sur un processus de traitement du langage naturel qui repère l'information importante afin de la structurer. Les entrées de leur système sont des rapports en texte libre issus d'études radiologiques, et les auteurs précisent bien que leur système ne requiert pas de la part des médecins qu'ils changent leur style d'écriture, rendant le système utilisable avec des archives de comptes rendus existantes. Leur approche est statistique et repose sur la constitution d'une base d'entraînement annotée (via une interface graphique développée pour leurs besoins) sur laquelle de l'apprentissage automatique sera réalisé. Les auteurs évoquent la difficulté qu'il y a à travailler sur des documents utilisant un vocabulaire large et codé, avec des styles d'écritures agrammaticaux, ou abrégés, qui sont très communs dans ces corpus de spécialité (commme on a pu le voir dans les extraits de notre corpus), ainsi qu'à devoir gérer les variations de styles issues de divers praticiens. Leur approche utilise des *frames* qui sont similaires aux triplets RDF du web sémantique. Chaque frame a un sujet (une structure anatomique observée) et va contenir diverses propriétés (*hasLocation*, *hasSize*...) associées à une valeur. Ces frames sont prédéfinies à l'avance et vont devoir être instanciées. Il s'agit donc ici d'une structuration fine, mais comme on l'a dit, sélective. Le processus commence par utiliser un système basé sur des règles simples pour déterminer les débuts et fins de sections (donc les éléments appartenant au *niveau de granularité grossier*), dans lequel sont

codés en dur les titres les plus communs (*« Clinical history »*, *« Findings »*...). A l'intérieur de chaque section, la segmentation en phrases est réalisée par apprentissage grâce à un classifieur d'entropie maximale. Après construction du graphe de dépendance de chaque phrase grâce à un parseur statistique, les relations logiques qu'il contient sont inférées grâce à un ensemble de relations préétabli dépendant de la section dans laquelle on se trouve, et ces relations sont utilisées pour construire et remplir les frames.

Plus récemment et toujours en radiologie, Hassanpour et Langlotz (2015) montrent que 14 ans plus tard ce domaine doit toujours composer avec les comptes rendus en texte libre. La méthode est aussi ici d'extraire par apprentissage automatique le contenu pertinent à partir d'un modèle d'information établi au préalable. Les auteurs exposent la couverture limitée et les problèmes de généralisation inhérents aux méthodes basées sur des dictionnaires ou des règles. Le corpus utilisé vient de trois organisations différentes et leur modèle n'est pas spécifique à un type d'examen radiologique ou à un hôpital, il reste assez souple (assez bas sur le continuum de structuration, donc). Il s'agit d'un modèle à cinq classes qui seront utilisées pour annoter les comptes rendus d'une base d'apprentissage. Leur méthode est centrée sur une reconnaissance des entités nommées, pour laquelle ils proposent trois alternatives : par dictionnaire, par modèles de Markov d'entropie maximale (MEMM ou CRM) et par CRF.

Baneyx (2007) (chapitre 5) présente l'outil MedCKARe (ou plus tard MedOC), destiné à l'aide au codage médical de comptes rendus hospitaliers de pneumologie. Le codage des comptes rendus médicaux vise à faciliter leur indexation et la recherche de comptes rendus dans une base de données documentaire de l'hôpital. C'est un processus manuel fastidieux, et les systèmes automatisés d'aide au codage (Friedman et al. 2004) veulent faciliter cette étape rendue obligatoire par la législation française. L'outil MedOC se base sur des patrons lexico-syntaxiques (Malaisé et al. 2004) qui cherchent à reconnaître certaines séquences précises de texte, pour donner une représentation conceptuelle (donc en lien avec une ontologie de la pneumologie) des séquences extraites. Ces patrons sont représentés sous forme d'automates à états finis, par exemple `<Pathologie><DET>*<ObjetAnatomique>`, qui repérera tout nom de pathologie suivi d'un nombre quelconque de déterminants, eux-mêmes suivis du nom d'un objet anatomique. Cet automate reconnaîtra par exemple *« agénésie du corps calleux »*. Tout comme précédemment, le système fonctionne donc grâce à des règles d'extraction pré-établies, mais la différence principale est l'interactivité de l'outil : le codage se passe toujours sous la supervision du médecin.

2.3 Mesures de similarité basées sur des données utilisant une représentation formalisée

Dans cette partie nous verrons ce qu'est une ontologie et quels sont les langages logiques qui permettent d'en définir une, puis nous verrons les mesures de similarité entre entités définies dans une ontologie et enfin comment ces formalismes et mesures peuvent être utilisés en RàPC et en aide à la décision médicale.

2.3.1 Ontologies et langages formels de représentation des connaissances

Les ontologies, en informatique, sont un formalisme apparu il y a 20 ans en représentation des connaissances. C'est Gruber (1993) qui introduit ce terme en ingénierie des connaissances, et le définit comme étant *« une spécification explicite d'une conceptualisation d'un domaine »*. Guarino (1998) pose qu'une ontologie est une théorie logique prenant en compte le sens attendu d'un vocabulaire formel. Les ontologies sont la plupart du temps exprimées grâce aux langages des logiques de description, fréquemment appelées par leur nom anglais *description logics* ($\mathcal{DL}$) (Baader 2003). Il s'agit d'une restriction en terme d'expressivité de la logique du premier ordre, afin d'en fournir un fragment décidable. Le développement des ontologies partage avec celui des réseaux sémantiques (Lehmann 1992) qui l'a précédé la volonté de simplifier les choses (Marquis et al. 2014, chapitre 5) par rapport à la logique classique :

— l'acquisition, la formalisation et le partage des connaissances, ainsi que
— les processus de raisonnement nécéssaires pour inférer des connaissances (Calvanese et al. 2007).

Toutes ces propriétés ont fait des $\mathcal{DL}$ l'outil de préférence pour raisonner sur des ontologies dans le cadre du Web sémantique (Berners-Lee et al. 2001), et ce au travers des deux versions du standard OWL [4]. Nous ferons dans cette section une présentation rapide des logiques de description et de la manière dont le domaine médical a pu se les approprier, notamment au travers du Web sémantique.

2.3.1.1 Logiques de description

Les notions les plus importantes en $\mathcal{DL}$ sont les *concepts* (également appelées classes), les *individus* (ou instances de ces classes) et les *relations* (également appelées rôles) entre ces derniers. À titre d'example, on peut concevoir en $\mathcal{DL}$ la classe des humains ainsi :

$$Humain \sqsubseteq (Femme \sqcup Homme) \sqcap \neg(Femme \sqcap Homme) \tag{2.1}$$

Tout concept qui n'est pas dérivé d'autres est appelé concept *primitif* et les autres sont appelés concepts *définis*. Ainsi, dans une ontologie qui ne contiendrait que la définition (2.1) accompagnée de la déclaration de quelques individus, les concepts $Homme$ et $Femme$ seraient primitifs et $Humain$ serait un concept défini. Toute disjonction ($\sqcup$), conjonction ($\sqcap$) ou négation ($\neg$) de concepts étant un nouveau concept dérivé des précédents, on exprime ainsi que la classe des humains est composée de tous les individus qui sont soit des femmes, soit des hommes, mais pas les deux à la fois. Ceci est nécessaire car en $\mathcal{DL}$ nous faisons l'hypothèse du *monde ouvert* : on considère que la connaissance présente dans l'ontologie n'est jamais complète, et rien dans la définition (2.1) ne permet de déduire que les concepts $Femme$ et $Homme$ sont disjoints. Si l'on souhaite inclure ce nouvel axiome, on peut réécrire (2.1) en (2.2).

4. Web Ontology Language. Voir http://www.w3.org/2004/OWL

$$\begin{aligned} Homme &\sqsubseteq \neg Femme \\ Humain &\sqsubseteq Femme \sqcup Homme \end{aligned} \tag{2.2}$$

Les relations quant à elles peuvent être définies grâce à des quantificateurs, pour imposer des restrictions sur la manière dont les concepts peuvent ou doivent être reliés entre eux pour être corrects au regard de l'ontologie. Nous pouvons par exemple créer un nouveau concept :

$$\begin{aligned} HumainNormalementFormé \sqsubseteq & Humain \\ & \sqcap (\forall\ aPourMembre\ .\ (Bras \sqcup Jambe)) \\ & \sqcap (= 2\ aPourMembre\ .\ Bras) \\ & \sqcap (= 2\ aPourMembre\ .\ Jambe) \end{aligned} \tag{2.3}$$

exprimant qu'un humain normalement formé ne peut avoir pour membres que des bras ou des jambes et qu'il doit en avoir exactement deux de chaque. Ainsi, l'expression $(\forall aPourMembre.(Bras \sqcup Jambe))$, qui utilise une quantification universelle, est un concept et peut donc être utilisée dans une conjonction. C'est le concept de tous les individus qui, lorsqu'ils sont reliés à un ou plusieurs autres individus par la relation $aPourMembre$, ne sont reliés par cette relation qu'à des instances du concept $(Bras \sqcup Jambe)$ et jamais à autre chose.

On notera que dans les définitions (2.1) et (2.3) nous avons créé nos concepts définis non pas par égalité mais par *subsomption* ($\sqsubseteq$) entre la classe définie et la conjonction de concepts qui sert à la définir. Par exemple dans le cas de (2.3), être un humain normalement formé implique d'avoir deux bras et deux jambes, mais tout humain ayant deux bras et deux jambes n'est pas nécessairement normalement formé. Comme toute expression en $\mathcal{DL}$, la formule (2.1) peut être traduite en logique du premier ordre classique :

$$\forall X.Humain(X) \rightarrow (Femme(X) \vee Homme(X)) \wedge \neg(Femme(X) \wedge Homme(X))$$

La relation de subsomption est utilisée pour former des hiérarchies de concepts. Une relation de subsumption équivalente existe pour les relations et permet donc de former des hiérarchies de relations. Ainsi, $aPourJambeDroite \sqsubseteq aPourJambe \sqsubseteq aPourMembre$ permet à un raisonneur de $\mathcal{DL}$ d'inférer que si un individu a une jambe droite, alors il a forcément au moins un membre.

Dans ce modèle de représentation de la connaissance, la *signature* d'une ontologie $\mathcal{O}$ est un vocabulaire, un triplet contenant un ensemble de concepts primitifs, un ensemble d'individus et un ensemble de rôles primitifs. $\mathcal{O}$ adjoindra à ce vocabulaire formel un ensemble de formules logiques, telles que (2.3), construites pour restreindre l'ensemble des modèles acceptables du domaine considéré et donc donner du sens à ce vocabulaire. L'*interprétation* $\mathcal{I}$ de $\mathcal{O}$ donne la sémantique de $\mathcal{O}$ en des termes ensemblistes. Elle se définit sur un domaine d'interprétation $\Delta^{\mathcal{I}}$. Il s'agit d'un ensemble, tel que pour tout $x_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, $x_i^{\mathcal{I}}$ soit l'interprétation d'un individu x_i de $\mathcal{O}$. Si A est un concept de l'ontologie $\mathcal{O}$, alors son interprétation sera $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$. S'il existe dans $\mathcal{O}$ une relation de subsomption $A \sqsubseteq B$, son interprétation sera $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$. Toute ontologie admet un concept $\top$ dont l'interprétation est l'ensemble

$\Delta^{\mathcal{I}}$ lui-même, et un concept $\perp$ dont l'interprétation est l'ensemble vide. La négation $\neg A$ d'un concept A aura pour interprétation :

$$(\neg A)^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} | A^{\mathcal{I}} \cap (\neg A)^{\mathcal{I}} = \varnothing \wedge A^{\mathcal{I}} \cup (\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

Pour finir, chaque rôle établissant une relation entre deux individus, pour tout rôle R on aura son interprétation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Dans le cas plus spécifique du langage OWL, Patel-Schneider et al. (2004) exposent la sémantique de chaque constructeur du langage en théorie des modèles. Motik et al. (2009) font de même pour la sémantique directe de OWL 2.

On peut définir plusieurs fragments de langages de logiques de description, chacun contenant ses constructeurs ajoutant un niveau particulier d'expressivité et à combiner pour créer des expressions dans cette logique. Par exemple :

Langage	Constructeurs
$\mathcal{AL}$	Concepts primitifs, hiérarchies de concepts, rôles primitifs, conjonctions, quantification universelle pour les rôles, négations de concepts primitifs
$\mathcal{C}$	Négations de concepts arbitraires
$\mathcal{U}$	Disjonctions
$\mathcal{E}$	Quantification existentielle pour les rôles
$\mathcal{R}$	Conjonctions de rôles
$\mathcal{H}$	Hiérarchies de rôles
$\mathcal{R}^+$	Transitivité des rôles
$\mathcal{N}$	Restriction de cardinalité, ex. $(\geq 1\ aPourMembre)$
$\mathcal{Q}$	Restriction de cardinalité qualifiée, ex. $(\geq 1\ aPourMembre.Bras)$

Ainsi, les langages $\mathcal{ALUER}^+$ et $\mathcal{ALCR}^+$, qui sont équivalents en termes d'expressivité [5] et souvent abrégés en $\mathcal{S}$, sont une logique de description particulière. Le langage $\mathcal{SHI}$ est la base de toutes les versions et déclinaisons du standard OWL.

2.3.1.2 Alternatives aux logiques de description

Bien que les plus populaires à l'heure actuelle, les logiques de description ne sont pas les seuls langages utilisables pour décrire des ontologies. Les graphes conceptuels (Sowa 1976) (Mugnier et Chein 1996), issus des réseaux sémantiques, proposent un cadre de travail où les inférences sont réalisées via des opérations sur des diagrammes, et possèdent des capacités d'expressivité équivalentes aux $\mathcal{DL}$. Kifer et al. (1995) présentent la F-Logic, qui est un langage à cadres (ou *frames*, selon Minsky (1974)) avec des éléments de description de données reprises des langages de programmation orientée objet apparus depuis. On y retrouvera donc les notions d'héritage, de polymorphisme par sous-typage (une fonction acceptant une instance de la classe A acceptera également toute instance d'une sous-classe de A) et d'encapsulation. Le but est de fournir une syntaxe plus compacte que les $\mathcal{DL}$ tout en conservant

5. Puisque $A \sqcup B = \neg(\neg A \sqcap \neg B)$

une sémantique précise. De plus la F-Logic inclut des éléments de raisonnement non monotone qui n'existent pas en $\mathcal{DL}$. Une logique est dite monotone si il n'est pas possible d'invalider des inférences réalisables à partir d'une base de faits donnée en *ajoutant* de nouveaux faits, uniquement en retirant ou en modifiant les faits déjà présents. Dans une logique monotone, il est donc possible d'exprimer un fait du type « les oiseau peuvent voler » et ensuite d'ajouter le fait « une autruche est un oiseau *qui ne peut pas voler* ». Le premier fait est considéré comme une règle *par défaut*. Cela n'est pas possible dans une logique monotone car les deux faits sont considérés comme contradictoires.

Ait-Kaci (2007) décrit les *Order-Sorted Feature Logics* ($\mathcal{OSF}$) – décrites initialement par Carpenter (1992) – qui selon l'auteur trouvent comme les $\mathcal{DL}$ leur source dans les travaux de Brachman (1978). Les $\mathcal{OSF}$ sont des langages logiques représentant et permettant l'inférence sur les *feature terms* (Ontañón et Plaza 2013). Aït-Kaci compare les $\mathcal{OSF}$ avec les $\mathcal{DL}$ en termes de capacités d'expression et de raisonnement. En $\mathcal{OSF}$, un *feature term* ψ est défini comme ceci

$$\psi ::= X : s\ [f_1 \doteq \Psi_1, ..., f_n \doteq \Psi_n]$$

où X est la variable *racine* ($root(\psi)$) de *sorte* s, où les f_i sont les *features* (caractéristiques) de X et où les Ψ_i sont également des variables ou des ensembles de variables, elles-mêmes munies de leurs *features*. Les *sortes*, qui agissent comme des types pour les variables, sont reliées dans une hiérarchie $\mathcal{S}$, un ensemble muni d'une relation d'ordre partiel, tel que $s \leq s'$ signifie que s est une sorte plus générale que s'. Les *feature terms* ont déjà été appliqués au domaine médical par Armengol et Plaza (2003) pour mesurer des similarités entre des cas d'empoisonnements de rats pour prédire l'activité cancérigère de composés chimiques.

A partir de ces propriétés, il existe pour les termes une relation de subsomption ($\sqsubseteq$).[6] Un terme ψ_1 subsume (est plus général) qu'un terme ψ_2 si ψ_2 contient toutes les variables et features de ψ_1, plus éventuellement des variables et/ou features supplémentaires et des sortes plus spécifiques dans $\mathcal{S}$ pour les variables déjà présentes dans ψ_1. On pourrait être tenté de faire l'analogie entre les sortes d'$\mathcal{OSF}$ et les concepts (ou classes) de $\mathcal{DL}$, mais ce serait abusif car en $\mathcal{OSF}$ les « classes » seraient elles-mêmes représentées par des termes, qui seraient simplement plus généraux que les termes correspondant à leurs « instances ». Ainsi, en $\mathcal{OSF}$ il n'y a pas comme en $\mathcal{DL}$ à faire de choix de granularité, autrement dit de choisir si une entité donnée doit être représentée comme classe (qui pourra encore être sous-catégorisée) ou comme instance (qui est terminale). Tout terme en $\mathcal{OSF}$ pourra toujours être sous-catégorisé, autrement dit rendu plus spécifique par l'ajout de features ou par le remplacement de la sorte s d'une variable par une sorte s' plus spécifique dans $\mathcal{S}$.

6. Dans la littérature sur les $\mathcal{OSF}$, le sens des relations d'ordre telles que $\leq$ ou $\sqsubseteq$ est souvent inversé par rapport à celle sur les $\mathcal{DL}$. Ainsi, en $\mathcal{DL}$ on a $A \sqsubseteq B$ qui signifie que A est plus *spécifique* que B, autrement dit que les *instances* de A sont inclues dans celles de B. À l'inverse, dans la littérature sur les $\mathcal{OSF}$, $A \sqsubseteq B$ signifiera que le terme A est plus *général* que B, autrement dit que les *caractéristiques* de A sont inclues dans celles de B. Il s'agit de deux manières différentes de voir la relation d'inclusion, la première plus proche d'une relation d'inclusion d'ensembles $\subseteq$ et la seconde d'une relation d'inclusion d'un graphe dans un autre.

2.3.1.3 Les ontologies et le Web sémantique dans le domaine médical

Le domaine du Web sémantique a été lié au domaine médical peu de temps après sa création. L'un des premiers apports attendus est l'amélioration de l'intéropérabilité des données. Le Web sémantique fournit en effet des standards ouverts pour représenter (RDF) et donner du sens (SKOS, RDFs, OWL) aux données. Les données des systèmes d'information hospitaliers sont la plupart du temps enfermées dans des formats propriétaires. Ceci rend plus difficile la tâche d'agrégation de données dans le cadre d'études cliniques. Le développement de ressources *termino-ontologiques* (RTO) médicales permet de plus facilement réutiliser et opérationaliser ces données.

Des efforts ont été déployés récemment pour utiliser plus amplement les langages du Web sémantique pour représenter des données cliniques. Tao et al. (2011) par exemple présente une étude visant à fournir une base en RDF à la représentation des méta-données d'études cliniques exprimées dans le modèle HL7 Detailed Clinical Models [7] ou ISO11179. Au delà de leur intérêt pour les études cliniques rétrospectives à partir de données issues d'examens particuliers, les standards du Web sémantique ont aussi un intérêt pour augmenter l'intéropérabilité des ressources à la disposition des médecins eux-mêmes. Ainsi, dans le domaine des guides de bonnes pratiques cliniques par exemple, Galopin et al. (2014) proposent une méthode basée sur un raisonnement ontologique pour détecter les discordances entre différents guides de bonnes pratiques à l'intention des médecins généralistes pour le traitement du diabète et de l'hypertension.

Pour pouvoir partager aussi bien les données que le sens précis qu'elles doivent avoir, de nombreux efforts ont été dépensés dans la création d'ontologies de domaines, autrement dit d'ontologies apportant la spécification d'une conceptualisation d'un sous-domaine médical particulier, par exemple en génétique (Ashburner et al. 2000), en obstétrique (Dhombres et al. 2010), en pneumologie (Baneyx 2007), en oncologie (Sioutos et al. 2007) et en médecine urgentiste (Charlet et al. 2012). Pour centraliser et répertorier cette masse grandissante de ressources, des portails en ligne tels que BioPortal [8] en anglais ou HeTOP [9] (multilingue) ont été mis en place afin de rendre plus facilement accessibles ces ontologies médicales. Ces portails sont notamment intéressants pour les *mappings* (alignements) qu'ils proposent entre ces ontologies et également entre ceux qu'ils établissent avec des ressources termino-ontologiques plus générales de la médecine et donc de taille beaucoup plus imposante, qui ont souvent été développées à partir de thésaurus. Il s'agit de ressources telles que la SNOMED-CT, le Medical Subject Headings (MeSH), Foundational Model of Anatomy (FMA) ou l'Unified Medical Language System (UMLS).

Diverses méthodologies existent pour construire une ontologie d'un domaine médical. Les premières se basent sur un corpus de textes, tels que des rapports d'actes médicaux, et le processus pourra être soit manuel (Bachimont et al. 2002) (Biébow et al. 2005) soit semi-automatique, par apprentissage (Kamel et al. 2012) ou par extraction de relations sémantiques depuis un corpus (Liu et al. 2015). Les secondes construisent une ontologie essentiellement à partir de connaissances d'experts et d'entretiens avec un ou plusieurs spécialistes du domaine (Aimé 2015) ou à partir de connaissances déjà compilées, telles que des ressources terminologiques existantes, comme OntoADR (Ontology of Adverse Drug

7. http://wiki.hl7.org/index.php?title=Detailed_Clinical_Models
8. http://bioportal.bioontology.org
9. http://hetop.eu

Reactions) construite à partir de MedDRA (Medical Dictionary for Regulatory Activities) (Souvignet et al. 2016). Aimé (2015) traite aussi de construction d'ontologies à partir de corpus, pour lequel l'auteur pose le critère de qualité suivant :

> [Le] corpus doit être suffisamment large pour couvrir tout le domaine, et consensuel pour répondre à l'objectif d'une ontologie qui est – par définition – la formalisation d'une conceptualisation consensuelle.

Comme on a pu déjà l'évoquer en section 1.1, une grande part des données médicales sont et resteront sous format textuel. Il est donc nécessaire à un moment donné de pouvoir relier les connaissances présentes dans les RTO avec les données existant dans un corpus. Ceci se nomme une tâche d'*annotation sémantique*. Funk et al. (2014) mentionnent plusieurs solutions spécalisées mais expose qu'une reconnaissance générale dans un texte de concepts arbitraires est toujours un problème ouvert. Pereira et al. (2009) présentent le système d'annotation utilisé dans le portail en ligne HeTOP. Thessen et Parr (2014) utilisent l'annotation pour repérer les données pertinentes dans un corpus de biologie (the Encyclopedia of Life [10]) afin de réaliser l'extraction de connaissances.

2.3.2 Mesures de similarités sémantiques

Nous avons vu en section 2.1 les méthodes permettant de concevoir une mesure de similarité entre des fragments de texte. Certaines de ces méthodes, comme la LDA, peuvent exploiter la sémantique *latente* d'un mot ou d'une expression si elle occure relativement fréquemment dans un corpus textuel. Mais étant donné que nous avons également vu en 2.3.1.1 des outils permettant de formaliser la sémantique des concepts (qui peuvent être utilisés ou non dans un corpus), nous pourrions souhaiter disposer d'un moyen d'exploiter cette sémantique *formelle* dans le cadre d'une mesure de similarité. Nous aurions donc ainsi des mesures de similarités entre concepts et non plus entre unités textuelles. Ces mesures existent et se nomment mesures de similarité sémantiques, et l'un des travaux séminaux sur le sujet est celui de Tversky (1977). Tversky présente dans son article de référence la façon dont on peut concevoir les objets comme des ensembles de caractéristiques, et la similarité entre deux objets comme découlant de l'intersection entre leurs ensembles respectifs. L'intuition derrière ce mode de représentation et de mesure des similarité est la même que celles des mesures de similarité sémantiques qui seront proposées par la suite : plus deux concepts ont de propriétés communes, plus ils sont sémantiquement proches. Une nuance introduire par Tversky qui sera rarement reprise par la suite est la considération qu'une mesure de similarité peut ne pas être symétrique, autrement dit que $sim(a, b) \neq sim(b, a)$. Il montre qu'effectivement, dans le langage courant nous ne concevons pas que dire « A est comme B » veuille dire la même chose que « B est comme A », et nous privilégierons des formules du type « cet homme ressemble au Président » plutôt que « le Président ressemble à cet homme », car nous mettrons toujours comme sujet de la phrase l'objet (ou « stimulus ») le moins salient, et comme référent l'objet le plus prototypique.

10. http://eol.org

Tversky considérera la similarité entre deux objets a et b comme étant

$$s(a,b) = F(A \cap B, A \setminus B, B \setminus A)$$

à savoir une fonction de trois paramètres :

— les caractéristiques communes à a et b (l'intersection de leurs ensembles de caractéristiques respectifs A et B),
— les caractéristiques spécifiques à a (A privé de B),
— les caractéristiques spécifiques à b (B privé de A).

Sous certaines conditions exposées par Tversky, on peut avoir une échelle de similarité S et une échelle positive f telles que pour tout quadruplet (a,b,c,d) appartenant au domaine de discours et leurs ensembles de caractéristiques associés (A,B,C,D) :

$$S(a,b) \geq S(c,d) \iff s(a,b) \geq s(c,d) \tag{2.4}$$

$$S(a,b) = \theta f(A \cap B) - \alpha f(A \setminus B) - \beta f(B \setminus A) \tag{2.5}$$

où θ, α et β sont des paramètres réels positifs. On a donc une échelle de similarité S qui préserve l'ordre des similarités observées s (formule 2.4) et exprime la similarité comme une combinaison linéaire de mesures sur les caractéristiques communes et distinctes des objets. Tversky appelle ceci le *modèle de contraste*. Il expose également le *modèle de ratio*, où 2.5 est remplacée par :

$$S(a,b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A \setminus B) + \beta f(B \setminus A)} \tag{2.6}$$

où S est maintenant normalisée. Dans ces deux modèles, toute différence entre les paramètres α et β introduira un biais, rendant la mesure de similarité non symétrique. On peut déjà remarquer qu'un modèle de ratio où $\alpha = \beta = 1$ rejoint l'intuition derrière l'utilisation de l'indice de Jaccard (évoqué en section 2.1.2) lors d'une analyse distributionnelle : une mesure de ce qui est commun à a et b divisé par ce qui est soit propriété de l'un, soit propriété de l'autre.

Faisons dès maintenant une distinction importante : est-ce la même chose de dire que deux objets sont *proches* et de dire qu'ils sont *similaires* ? Un certain nombre d'articles de la littérature font ici une distinction, et discriminent la *proximité sémantique* de la *similarité sémantique* (Harispe et al. 2013). De manière informelle, comme on l'a déjà évoqué en section 1.1.4, on dira que les concepts de « tasse » et de « café » partagent une *proximité* alors que « café » et « thé » partagent une *similarité*. Posons qu'on ait une ontologie telle que celle en figure 2.3 qui contienne ces concepts. « Café » et « thé » auront plusieurs ancêtres communs (tels que « boisson » ou « liquide ») autrement dit des concepts atteignables à partir de « café » comme de « thé » en suivant *uniquement* des liens de subsomption (`subclassOf` dans le langage OWL, $\sqsubseteq$ en $\mathcal{DL}$). Il est également fort probable qu'il y ait des liens indirects entre

« café »/« thé » et « tasse ». Ces liens ne seront toutefois cette fois-ci pas uniquement composés de liens de subsomption, mais également de propriétés spécifiques à cette ontologie, par exemple une propriété $estContenantDe$ que l'ontologie peut utiliser pour définir le concept de « récipient ».

FIGURE 2.3: Mini-ontologie des liquides et boissons

$$
\begin{aligned}
Récipent &\sqsubseteq (\forall estContenantDe.Liquide) \sqcap (\leq 1\ estContenantDe) \\
Boisson &= Liquide \sqcap ConsommableParHumain \\
BoissonSansAlcool &= Boisson \sqcap \neg Alcoolisé \\
Tasse &\sqsubseteq Récipent \sqcap (\forall estContenantDe.Boisson)\ ^{11} \\
(Café \sqcup Thé) &\sqsubseteq BoissonSansAlcool \\
Café &\sqsubseteq \neg Thé
\end{aligned}
$$

RESNIK (1995) définit une métrique qui sera fréquemment reprise par la suite dans la littérature : le contenu informationnel IC, calculé ainsi pour tout concept C : $IC(C) = -log(p(C))$, p étant la probabilité de voir le concept apparaître dans le texte (calculée par exemple par analyse de la fréquence de chaque concept dans un corpus).

Comme nous l'avons expliqué, il y a souvent beaucoup d'intuition en commun derrière les mesures de similarités proposées ces dernières décennies. La revue faite par HARISPE et al. (2013) établit des abstractions pour indiquer les points communs dans ces différentes mesures, et permet de s'apercevoir que certaines de ces mesures ont des formules qui deviennent identiques dès lors que l'on abstrait certains indicateurs, comme le contenu informationnel. Cette revue de la littérature classe les mesures de similarités sémantiques pouvant être appliquées sur des graphes acycliques (ce que sont la plupart des ontologies) en quatre catégories :

— Les mesures basées sur l'analyse de la structure des graphes, qui estiment la similarité comme une mesure du degré d'interconnection entre les classes et utilisent des techniques telles de la marche aléatoire ou la recherche de plus court chemin ;
— Celles basées sur l'analyse des caractéristiques des classes (concepts), qui estiment la similarité comme une fonction des caractéristiques partagées et distinctes des classes comparées, ce qui reste très proche de la caractérisation originale de TVERSKY (1977) ;
— Celles basées sur la théorie de l'information, notamment celles qui se basent sur l'IC au sens de RESNIK (1995), en évaluant la quantité d'information véhiculée par chaque classe ;
— Celles qui sont des hybrides des trois types précédents.

Dans le cadre du Web sémantique, KIEFER et BERNSTEIN (2008) présentent une extension au langage SparQL (nommée iSPARQL) permettant d'utiliser des mesures sémantiques lors de requêtes pour récupérer non plus uniquement les résultats correspondant exactement à la requête mais également ceux qui sont sémantiquement proches des résultats exacts.

11. Exemple d'engagement ontologique (cf. BACHIMONT 2000) : bien qu'une tasse de Sans Plomb 95 soit physiquement concevable, ici nous l'excluons de notre modèle.

2.3.3 Usage des technologies du Web sémantique en RàPC

Les problématiques de représentations des connaissances étant au coeur de la thématique RàPC, c'est donc naturellement que cette communauté a commencé à utiliser les ontologies pour formaliser les cas et les liens qu'ils entretiennent (Richter et R. Weber 2013a) ainsi que les langages issus du web sémantique pour représenter ces cas en base et pour les échanger.

Nous présentons par la suite des travaux en RàPC utilisant des connaissances pour réaliser diverses étapes du cycle de RàPC, puis des applications dans le domaine médical.

2.3.3.1 RàPC et connaissances du domaine

Les outils de RàPC ont assez rapidement intégré des méthodes de représentation et de comparaison de cas tenant compte de la sémantique. Ainsi, deux des principaux outils de création d'application de RàPC sans développement myCBR (Bach et al. 2014) et jCOLIBRI (Recio-García et al. 2014) proposent une intégration du langage OWL et de certaines mesures de similarité sémantiques. Avant cela, divers travaux s'étaient intéressés à l'application d'un système de raisonnement à partir de cas sur une base de connaissances formalisées. Bergmann et Schaaf (2003) tout d'abord se sont intéressés aux relations qui existent entre le RàPC structuré et la gestion de connaissances à base d'ontologies (F-Logic dans leur cas), et concluent à une forte relation entre les deux approches aussi bien technologiquement que méthodologiquement. Aamodt (2004) décrit le paradigme du *Knowledge-Intensive CBR* (KI-CBR), où le système est enrichi avec des connaissances générales du domaine considéré dans le but de « permettre au système de RàPC de raisonner avec des critères pragmatiques et sémantiques plutôt que purement syntaxiques ».

Diverses recherches ont eu lieu pour intégrer le Web sémantique et la prise en compte de la sémantique dans les diverses étapes du cycle de RàPC, comme l'adaptation (Lieber et al. 2008) (Cojan et Lieber 2011) ou l'acquisition de connaissances (Badra et al. 2009).

Ontañón et Plaza (2012) s'intéressent à l'utilisation des opérateurs de raffinement (généralisation et spécialisation) sur les *feature terms* pour la partie remémoration du RàPC.

2.3.3.2 Applications en informatique médicale

Le raisonnement à partir de cas étant l'application du raisonnement par analogie au sein de systèmes d'aide à la décision, il semble assez naturel de vouloir l'appliquer à des domaines dans lesquels les spécialistes procèdent souvent par analogie pour identifier des pathologies. Ainsi, les systèmes de RàPC médicaux sont une branche des systèmes d'aide à la décision médicale, qui eux-mêmes sont une branche de l'informatique médicale. Les liens entre l'intelligence artificelle, l'informatique et la médecine ne sont pas récents, et divers ouvrages existent sur le sujet (Fieschi 1984) (Grémy 1987) (Degoulet et Fieschi 2012).

Ici, par analogie nous n'entendons pas nécessairement de comparer des cas entre eux, mais également de comparer ces cas à des prototypes déjà établis, prototypes qui ne correspondent pas nécessairement à des cas réels ayant existé. Deux problématiques importantes sont donc ici étudiées : représenter et comparer. Représenter implique de séparer ce qui a trait à un cas particulier (les données de ce cas) de ce qui est connaissance générale d'un domaine médical. Pantazi et al. (2004), qui montrent une utilisation du raisonnement à partir de cas pour la détection de motifs en imagerie médicale, articulent les liens entre connaissances générales et connaissances individuelles (les cas venant des dossiers patient informatisés), et expliquent la difficulté en informatique médicale de faire se rencontrer ces deux types de connaissances.

Divers travaux de revue de la littérature scientifique sur les systèmes de RàPC médicaux ont été effectués : Gierl et al. (1998) présentent une revue des systèmes rapportés avant 1998, Nilsson et Sollenborn (2004) entre 1999 et 2003, Ahmed et al. (2009) entre 2004 et 2008 et Begum et al. (2011) entre 2004 et 2010. Les deux derniers présentent une trentaine de travaux de RàPC appliqués à des domaines médicaux variés, allant de la classification de types de leucémies à la planification à partir de cas pour l'assistance aux patients atteints de maladies neurodégénératives. Les buts généraux de ces systèmes peuvent être le diagnostic, la classification, l'acquisition de connaissances ou la planification. Ces revues de la littérature identifient plusieurs problèmes auxquels les systèmes de RàPC médicaux doivent faire face :

— l'extraction des caractéristiques (*features*) de chaque cas s'est complexifiée du fait de la nécessité de la prise en compte de données issues de capteurs, d'imagerie médicale et bien entendu de texte libre ;
— la sélection de ce qui doit être considéré comme une caractéristique d'un cas est plus subtile car elle fait appel à l'expertise du praticien ;
— il est difficile de constituer une base qui à la fois contienne suffisamment de cas (une base de cas exhaustive) et qui ne contienne que des cas représentatifs. Ceci résulte en des données éparses, en une augmentation du risque le système rencontre des nouveaux cas ne correspondant pas à un cas passé pertinent, en des capacités limitées de réutilisation des connaissances contenues dans la base et donc plus généralement en une diminution des performances du système ;
— la complexité et la rapidité de mise à jour des connaissances des domaines médicaux considérés décourage la plupart des systèmes d'implanter l'étape d'adaptation du cycle de RàPC. L'adaptation reste donc fréquemment une étape manuelle, ce qui contribue à augmenter encore la dépendance aux praticiens.

Tous ces travaux montrent une grande tendance à réaliser des systèmes hybrides, suivant le cycle de RàPC classique en incluant des éléments de fouille de données, d'apprentissage automatique, de logique floue ou bien entendu de bases de connaissances représentées et éventuellement formalisées avec les outils du Web sémantique. Cependant, parmi tous les systèmes répertoriés par Ahmed et al. (2009) et Begum et al. (2011), seulement trois semblent utiliser une technologie provenant du Web sémantique : KASIMIR (Lieber et al. 2008), Mémoire (Bichindaritz 2004) et le système présenté par D. Wu et al. (2004). Dans le champ plus restreint des maladies rares, ces revues mentionnent l'étude de Töpel et al. (2007) qui présente un composant de RàPC pour la base RAMEDIS de maladies métaboliques génétiques rares.

Parmi les applications moins communes du RàPC dans le domaine médical qui ne sont pas adressées par ces revues de la littérature, on trouve l'aide au consensus entre praticiens, telle que présentée par Thieu et al. (2004) et Steichen et al. (2006).

2.4 Comparaisons de structures de données

Dans cette partie nous nous intéresserons aux méthodes permettant la mesure de similarité entre structures de données séquentielles ou arborescentes, ainsi que les manières de réaliser des alignements entre deux structures, notamment via la notion de chaîne d'éditions, autrement dit de modifications qu'il faut réaliser pour passer d'une structure à une autre.

2.4.1 Mesure de similarités entre chaînes de symboles

Nous avons défini précédemment plusieurs manières de mesurer des distances entre des concepts (à l'aide de mesures de similarités sémantiques) ou entre des fragments de texte (à l'aide notamment des modèles vectoriels). Il existe également des cas où nous voulons comparer deux mots entre eux hors de tout contexte, notamment si l'orthographe de ces deux mots est incertaine. Cela est rendu possible grâce aux mesures de distance entre chaînes (*string metrics*), qui manipulent des chaînes de symboles sans connaissance préalable de la nature de ces symboles, et traiteront donc les mots de manière lexicale. Du fait de cette orientation, ces mesures de similarité sont principalement utilisées dans les correcteurs orthographiques et dans des systèmes à base de texte devant comparer des mots venant de sources potentiellement bruités. D'autres domaines en font également usage, notamment en biologie pour comparer des séquences d'ADN.

L'une des mesures les plus connues est celle de Damereau-Levenshtein (Wagner et Lowrance 1975), qui établit le nombre minimal d'ajouts, de suppressions, de transpositions de deux symboles adjacents et de remplacements de symboles pour passer d'une chaîne à l'autre. Ces modifications élémentaires sont appelées *opérations* d'édition sur les chaînes. Ce faisant, elle établit une correspondance entre les différentes portions des deux chaînes, également appelée *alignement*. Selon le cas d'utilisation, il est possible d'ajouter des poids différents aux quatre opérations (par exemple pour que les remplacements comptent plus, autrement dit induisent une distance plus grande, que les transpositions), mais la plupart du temps dans le cas de la mesure de distance entre mots ces poids sont tous égaux à 1. La raison motivant cette mesure est que, selon Damerau (1964), ces opérations d'édition correspondent à plus de 80% des fautes d'orthographe généralement faites par un humain.

La distance dite de Levenshtein (1966) est très similaire, mais ne tient pas compte des transpositions de caractères adjacents, car leur prise en compte augmente la complexité du calcul. À l'inverse, la distance de Jaro-Winkler (W. W. Cohen et al. 2003) ne tient compte que des transpositions.

2.4.2 Mise en correspondance et mesure de similarités entre arbres

Comme l'expliquent F. Li et al. (2014), l'utilisation d'arbres pour représenter des données du monde réel est devenue prépondérante du fait de l'apparition du Web qui favorise aujourd'hui ces représentations, notamment avec XML ou JSON. [12] Des approches basées sur la comparaison d'arbres ont déjà été utilisées en traitement du langage naturel : Kouylekov et Magnini (2006) l'utilisent pour reconnaître une relation d'implication entre deux fragments de texte ; Reis et al. (2004), Lin et al. (2010) et Lakkaraju et al. (2008) pour établir des similarités entre documents afin d'effectuer de la recherche d'information (notamment sur le web). Ce dernier article observe d'ailleurs une nette amélioration par rapport aux traditionnelles mesures de similarités basées sur un modèle vectoriel.

De la même manière qu'on l'a pu concevoir sur des chaînes, mesurer la similarité entre deux arbres peut se faire au travers de l'établissement de la séquence des opérations d'édition que l'on doit réaliser pour passer d'un arbre à l'autre. On appelera ces mesures des distances d'édition d'arbres. De cette manière, trouver les correspondances entre deux arbres et établir une métrique de similarité entre eux sont deux facettes d'un même problème, dans le sens où les portions d'arbres qui n'ont pas besoin d'être éditées sont les parties de leurs structures où ces arbres correspondent parfaitement, et celles où les éditions ont lieu sont celles qui correspondent partiellement. Dans la littérature ce problème est également appelé *tree-to-tree correction problem* (Tai 1979). Les arbres manipulés dans les sections qui suivent sont toujours enracinés (un noeud ancêtre de tous les autres est toujours présent), étiquetés et (sauf indication contraire) ordonnés. Leurs étiquettes peuvent être typées, auquel cas dire que T est un « arbre de X » signifie que chaque noeud de T porte une étiquette qui appartient à l'ensemble X, et dire que T est un arbre de X signifie que les étiquettes de T sont toutes des séquences d'éléments de X.

2.4.2.1 Distance d'édition d'arbres

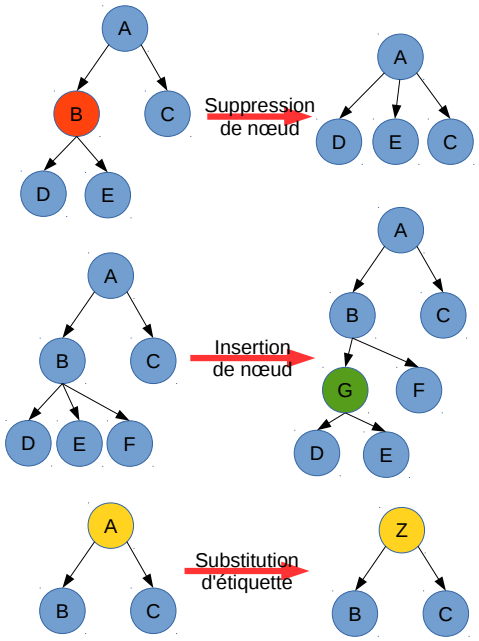
Étant donnés deux arbres T_1 et T_2, on définit la distance d'édition d'arbres (TED, *tree edit distance*) comme étant le nombre minimum d'operations pour passer de T_1 à T_2. Ces opérations (cf figure 2.4) sont classiquement au nombre de trois :

— suppression d'un noeud v (et déplacement de ses enfants vers son parent) ;
— insertion d'un fils v à un noeud u (et déplacement d'une portion contigüe des fils de u vers v) ;
— substitution d'étiquette d'un noeud v.

On définit λ comme étant le noeud vide, et on note $(l_1 \to l_2)$ une opération d'édition. Si $l_1 = \lambda$ on a donc une insertion et si $l_2 = \lambda$ une suppression. De la même manière que pour les séquences, on peut associer à chaque opération un coût. On aura ici une fonction de coût $c(l_1 \to l_2)$ renvoyant un réel positif qui peut donc dépendre des étiquettes mises en jeu lors de l'opération d'édition [13]. On cherchera ensuite à établir la séquence d'opérations de moindre coût total pour passer de T_1 à T_2. Ce coût total sera la distance d'édition entre T_1 et T_2. On appelera *mapping* (Tai 1979) la description de la manière

12. eXtended Markup Language et JavaScript Object Notation
13. Si ces étiquettes sont des chaînes de symboles, on peut donc par exemple considérer le coût d'une substitution comme étant la distance d'édition de *chaînes* entre les deux étiquettes.

FIGURE 2.4: Opérations d'édition d'arbres



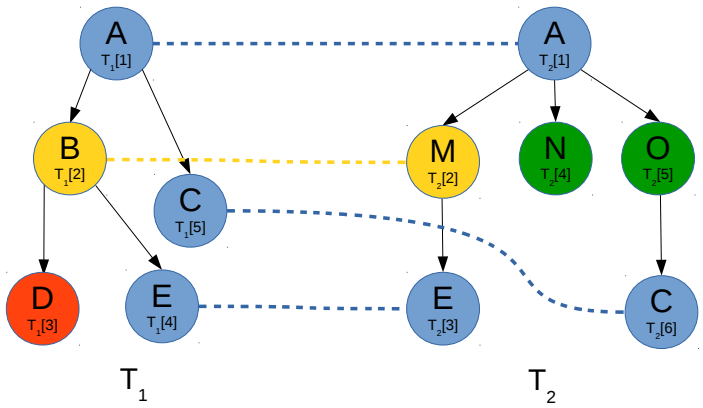
dont une séquence d'opérations transforme T_1 en T_2, sans rendre compte de l'ordre dans lequel les opérations sont appliquées. Ainsi, sur l'exemple donné en figure 2.5, les lignes pointillées désignent une correspondance entre deux noeuds (bleus si les étiquettes correspondent et jaunes si l'on doit faire une substitution d'étiquette). Les noeuds non « mappés » sont considérés comme supprimés (rouge) d'un côté et insérés (vert) de l'autre.

On note $|T|$ la taille (le nombre total de noeuds) d'un arbre T. Tout arbre T est indexé de telle sorte que pour tout i tel que $1 \leq i \leq |T|$ il existe $T[i]$, un noeud de T, et que les noeuds de T soient numérotés par leur position dans un préordre (ainsi, tout ancêtre a un index plus petit que ses descendants et tout noeud a un index plus petit que les frères à sa droite). TAI (1979) définit formellement un mapping comme étant un triplet (M, T_1, T_2) où M est un ensemble de paires d'entiers (i, j) telles que :

1. $1 \leq i \leq |T_1|, 1 \leq j \leq |T_2|$
2. $\forall((i_1, j_1), (i_2, j_2)) \in M^2$:
   a. $i_1 = i_2 \iff j_1 = j_2$ (règle de mapping *un-à-un*) ;
   b. $i_1 < i_2 \iff j_1 < j_2$;
   c. $T_1[i_1]$ est un ancêtre (resp. descendant) de $T_1[i_2] \iff T_2[j_1]$ est un ancêtre (resp. descendant) de $T_2[j_2]$.

Si n est un nœud de T_1 (resp. T_2), on note aussi $M(n)$ le nœud de T_2 (resp. T_1) auquel le mapping M le fait correspondre.

FIGURE 2.5: Mapping de deux arbres



Les conditions 2b. et 2c. imposent que le mapping respecte la structure générale des deux arbres, autrement dit que si l'on retire des arbres tous les noeuds devant être ajoutés ou supprimés et que l'on ne considère plus les étiquettes, alors les deux arbres sont parfaitement superposables sans que les liens de M ne se « croisent ». Ceci interdit que le mapping « transpose » deux noeuds : par exemple si $T_1[i_1]$ est à gauche de son frère $T_1[i_2]$ et que ces deux noeuds sont mappés respectivement avec $T_2[j_1]$ et $T_2[j_2]$, alors ces derniers seront frères et $T_2[j_1]$ sera à gauche de $T_2[j_2]$. Il en découle les trois règles de mapping par distance d'édition (SHAHBAZI et MILLER 2014) :

— mapping un-à-un des noeuds (déjà vue),
— préservation de l'ordre des frères,
— préservation de l'ordre des ancètres.

Le coût du mapping M est défini par :

$$c(M) = \sum_{(i,j)\in M} c(T_1[i] \to T_2[j]) + \sum_{i \notin M | \exists T_1[i]} c(T_1[i] \to \lambda) + \sum_{j \notin M | \exists T_2[j]} c(\lambda \to T_2[j])$$

et la distance d'édition entre T_1 et T_2 par :

$$dist(T_1, T_2) = min\{c(M)\}$$

L'algorithme classique (ZHANG et SHASHA 1989) a une complexité en temps en $O(n^4)$ et en espace en $O(n^2)$, n étant ici le nombre total de nœuds. Quel que soit l'algorithme utilisé, le calcul de la distance exacte d'édition d'arbres se fait toujours à notre connaissance avec au moins une complexité en temps en $O(n^3)$. DEMAINE et al. (2007) montre que dans le pire des cas, on ne peut pas descendre en dessous de cette complexité, et propose un algorithme en $O(n^3)$ en temps et $O(n^2)$ en espace. YANG et al. (2005) et LIN et al. (2010) proposent des méthodes de calcul de cette distance d'édition qui

opèrent sur des arbres transformés en séquences multidimensionnelles, autrement dit en des ensembles de séquences, chaque séquence étant les labels des nœuds se trouvant sur un chemin allant de la racine à l'une des feuilles. Pawlik et Augsten (2011) introduisent une classe de stratégies appelée LRH (*left-right heavy*), généralisent les approches précédentes de calcul de distance d'édition d'arbres, et s'attachent a conserver ses performances quelle que soit la forme des arbres en entrée, contrairement aux approches précédentes.

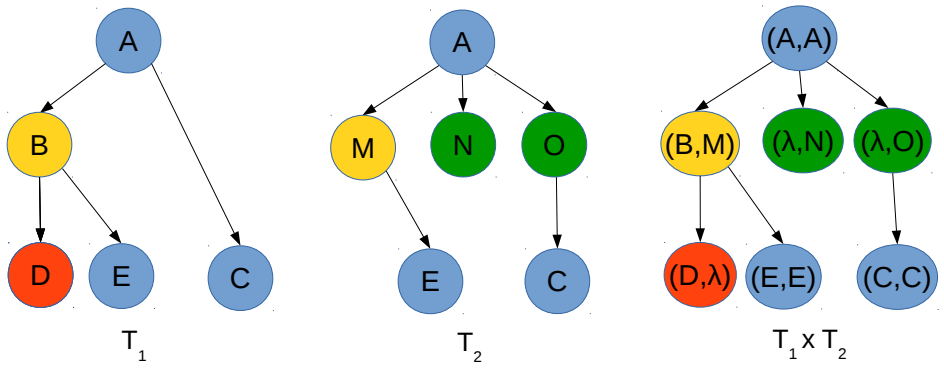
F. Li et al. (2014) évaluent plusieurs méthodes, appelées méthodes de filtrage et de raffinement, pour approximer cette distance d'édition. Ces méthodes consistent à calculer des bornes inférieures (autrement dit des approximations qui sous-estimeront toujours la distance d'édition réelle) sur des portions des arbres T_1 et T_2 et à les comparer à des seuils pour guider le calcul de la distance d'édition. Un exemple de ce type de procédé est de transformer les arbres en des chaînes et d'appliquer ensuite une distance d'édition de chaînes pour obtenir cette borne inférieure (G. Li et al. 2008). E. Tanaka et K. Tanaka (1988) et Zhang (1995) proposent une distance plus rapide à calculer au prix d'une contrainte supplémentaire : si (i_1, j_1), (i_2, j_2) et (i_3, j_3) sont dans le mapping, alors l'ancêtre commun le plus proche de i_1 et i_2 est un ancêtre de i_3 si et seulement si l'ancêtre commun le plus proche de j_1 et j_2 est un ancêtre de j_3. La mesure en résultant est parfois appelée *Isolated Subtree Distance* (ou *constrained editing distance*), puisque les noeuds de deux sous-arbres séparés de T_1 ne pourront se retrouvés mappés qu'avec les noeuds de deux sous-arbres séparés de T_2. Zhang (1995) propose un algorithme en complexité quadratique.

Différentes extensions des mesures de distance d'édition d'arbres ont été proposées par la suite. Augsten, Böhlen et Gamper (2005) présentent une distance appelée *pq-gram distance* qui approche cette distance d'édition avec une complexité en temps réduite, de $O(n\ log\ n)$, mais se comporte différemment avec les opération d'édition causant de gros changements dans la structure des arbres (ce type de changements augmente plus la distance que dans le cadre d'un calcul de distance d'édition classique), ce qui peut être souhaitable ou non selon le domaine d'application. Augsten, Böhlen, Dyreson et al. (2008) étendent cette mesure de distance aux arbres non-ordonnés avec la *windowed pq-gram distance*, autrement dit rendent cette mesure agnostique aux transpositions de fils d'un nœud donné, et donc remettre en cause la règle de préservation de l'ordre des frères. Shahbazi et Miller (2014) proposent une extension des mesures de distance d'édition d'arbres inspirée de celle de E. Tanaka et K. Tanaka (1988) mais avec de nouvelles règles pour le mapping des sous-arbres, sous la forme d'une fonction de similarité appelée EST (Extended Subtree). Celle-ci vise à lever deux verrous des distances d'édition d'arbres classiques : être agnostique des possibles transpositions de noeuds frères pouvant avoir lieu et remettre en cause la règle de mapping un-à-un en généralisant la suppression et l'insertion.

2.4.2.2 Alignement d'arbres

Une autre manière de concevoir la comparaison d'arbres est proposée par Jiang et al. (1994) sous la forme des alignements d'arbres. Ici, il s'agit tout d'abord de composer un arbre de X et un arbre de Y en uniformisant leurs structures pour obtenir un arbre de $X \cup \{\lambda\} \times Y \cup \{\lambda\}$ (cf. figure 2.6). On établit ensuite un coût total de l'alignement à partir des coûts d'alignement unitaires (définis de manière

FIGURE 2.6: Alignement de deux arbres



analogue aux coûts d'opérations d'édition), et l'on cherche l'alignement de moindre coût. HASSENA et MICLET (2009) montrent une variante gérant une multitude d'arbres, et permettant donc de passer d'une forêt de n arbres de X à un arbre de $(X \cup \{\lambda\})^n$. Dans les deux cas, les auteurs proposent une méthode basée sur la programmation dynamique pour calculer les alignements de moindre coût.

2.4.2.3 Etablissement d'un mapping entre deux arbres

Nous avons vu en 2.4.2.1 que la notion de *mapping* découlait de celle de séquence d'opérations d'édition entre deux arbres dans le cadre du *tree-to-tree correction problem*. Nous présentons ici diverses approches permettant d'établir un mapping entre deux arbres, qu'elles le fassent ou non dans le cadre du calcul d'une séquence d'opérations d'édition de coût minimal. Les approches présentées ici seront classées en fonction des propriétés qu'elles supposent des arbres, et donc des contraintes qu'elles imposent sur le mapping final.

Dans le tableau suivant, N_f représente le nombre total de feuilles dans les deux arbres à mapper.

Référence	Propriétés des arbres	Mapping calculé	Complexité en temps de l'algorithme proposé
AUGSTEN, BÖHLEN, DYRESON et al. 2008	Non-ordonnés	Approximé	$O(n\ log\ n)$
AUGSTEN, BÖHLEN et GAMPER 2005	Ordonnés	Approximé	$O(n\ log\ n)$
CHEN 2001	Ordonnés		
AKUTSU et al. 2014	Non-ordonnés	Exact	Exponentielle
SHASHA et al. 1994	Non-ordonnés, contrainte sur N_f	Exact	Polynomiale
SHASHA et al. 1994	Non-ordonnées	Approximé	Polynomiale

Le calcul de correspondances exactes dans des arbres non ordonnés sans ajout de contraintes supplémentaires sur les arbres est toujours en complexité exponentielle. Il est en effet prouvé que même

dans le cas où les labels des arbres sont tirés d'un alphabet binaire, ce problème est NP-complet (Zhang, Statman et al. 1992). Une idée introduite par Shasha et al. (1994) et reprise par Augsten, Böhlen, Dyreson et al. (2008) pour calculer un mapping approximé d'arbres non-ordonnés est d'utiliser une heuristique basée sur le tri des nœuds fils pour ensuite se ramener à un calcul de mapping d'arbres ordonnés, ou pour obtenir un mapping initial et ensuite procéder à son raffinement en utilisant une méthode de Monte Carlo tel qu'un algorithme de recuit simulé.

S. Cohen et Or (2014) s'attaquent au problème plus spécifique de retrouver dans un ensemble d'arbres Γ tous les *sous-arbres* des arbres de Γ qui sont les plus similaires à un arbre donné (problème qu'ils nomment *Subtree similarity-search*), en utilisant les mesures de distance entre arbres présentées en section 2.4.2.1.

Il existe donc dans la littérature une certaine quantité d'algorithmes permettant l'établissement de mappings approximés relativement souples de données arborescentes, et ce avec des complexités en temps permettant leur mise en pratique dans un système travaillant sur des données réelles, et donc des arbres pouvant atteindre plusieurs centaines de nœuds. En ce qui concerne leur opérationalisation justement, trois bibliothèques Java contiennent des fonctions dédiées à la réalisation de mappings d'arbres et/ou de calculs de distances d'édition : Approxlib [14] , Simpack [15] et APTED/RTED [16].

2.4.2.4 Mapping sans contraintes

Le *flexible tree matching* (Kumar, Talton et al. 2011) est un algorithme qui utilise une méthode stochastique pour échantillonner des approximations du meilleur mapping. Un mapping est un graphe biparti dont chaque arc est valué par un coût. Chaque coût de mise en correspondance de deux nœuds (N_1, N_2) est la somme de trois termes :

1. la distance entre les labels de N_1 et N_2 ;
2. le nombre de fils de N_1 ou N_2 non mis en correspondance avec des fils de l'autre nœud ;
3. le nombre de frères de N_1 ou N_2 non mis en correspondance avec des frères de l'autre nœud.

Les termes (2) et (3) – termes d'ascendance et de fratrie – couvrent le coût structurel du mapping, en évaluant la manière dont il respecte les formes globales des arbres. Un coût est également induit par chaque nœud non mis en correspondance (ceci est symbolisé par une mise en correspondance avec un « nœud vide » noté ψ). Chacun de ces termes est multiplié par un poids, permettant à l'utilisateur de régler l'importance de chaque terme en fonction de son domaine d'application. Voir (Kumar, Talton et al. 2011) pour l'algorithme détaillé – sur lequel nous reviendrons dans la méthode que nous proposons – et (Kumar, Satyanarayan et al. 2013) pour son application à la manipulation de pages Web.

14. http://www.cosy.sbg.ac.at/~augsten/src
15. https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/simpack/index.html
16. http://tree-edit-distance.dbresearch.uni-salzburg.at

2.5 Synthèse de l'état de l'art

Cet état de l'art apporte des éléments de réponses à la question suivante :

> Étant donné un texte chargé d'économies linguistiques et présentant une mise en forme matérielle sous forme de marqueurs non réguliers supportant une structure hiérarchique latente, que puis-je utiliser comme méthodes et outils pour (1) transformer ce texte en un arbre rendant manifeste cette structure et (2) comparer cet arbre avec un autre arbre produit de la même manière depuis un autre texte – ceci impliquant d'être capable (3) de reconnaître la connaissance contenue dans les nœuds des arbres et (4) de pouvoir comparer deux nœuds ?

Dans la méthode que nous proposons dans la suite de ce mémoire, le point (1) est adressé par la segmentation de séquences dont nous avons parlé en section 2.2.2, le point (2) en grande partie par le mapping évoqué en 2.4.2.4, le point (3) par le repérage dans le texte de concepts ontologiques (cf. 2.3.1.1) médicaux et le point (4) par les mesures de similarité textuelle et sémantique des sections 2.4.1 et 2.3.2.

Chapitre 3

Matériel

Le matériel utilisé pour ce travail de thèse est de deux types : d'un côté le corpus textuel de fœtopathologie et de l'autre les ressources termino-ontologiques médicales disponibles que nous utilisons.

3.1 Corpus d'Accordys

3.1.1 Provenance et contenu d'un cas documenté de fœtopathologie

Le corpus utilisé pour Accordys, que nous notons ACC dans la suite de ce mémoire, comprend environ 2 000 cas. Chaque cas comprend plusieurs documents de diverses natures :

— Comptes rendus d'échographies, d'IRM et de TDM, pré et post-natales ;
— Compte rendu d'examen de cytogénétique ;
— Photographies et diapositives d'autopsies du fœtus ;
— Radiographies du fœtus ;
— Courriers entre praticiens ;
— Compte rendu d'examen fœtoplacentaire(abrégé en CREFdans la suite de cette thèse).

Tous ces cas ont été sélectionnés par les docteurs Gonzalez et Razavi de l'hôpital Armand Trousseau à Paris, pour faire partie de ACC. Tous les cas de ACC sont présents dans l'hôpital sous forme de dossiers papier, ainsi l'intégralité de ACC est obtenue par la numérisation de ces dossiers, suivie d'une étape de reconnaissance des caractères (OCR) pour transformer les images en fichiers texte et enfin d'une étape d'anonymisation retirant les noms et adresses des patients et des médecins. Ces étapes sont réalisées par les partenaires du projet Accordys et ne font pas partie de ce travail de thèse. Le projet a également pour tâche la correction orthographique du corpus, notamment due aux erreurs d'OCR, mais cette tâche n'a pas encore été réalisée. Celle-ci nécessitant a minima la constitution d'un dictionnaire de termes du domaine, nous avons choisi de nous baser sur le corpus non corrigé. Nous

détaillons plus bas l'impact attendu.

Le travail de thèse présenté ici s'appuie uniquement sur les informations compilées dans le compte rendu d'examen fœtoplacentaire. C'est en effet, dans l'ordre chronologique des comptes rendus écrits sur un cas donné, le dernier compte rendu à être écrit et il rappelle la plupart du temps les observations importantes faites lors des examens précédents. Les photographies et radiographies ayant également déjà été interprétées par le(s) praticien(s), et ces interprétations présentes dans le compte rendu d'examen fœtoplacentaire, nous nous focalisons sur le texte et non sur les images. Ainsi, dans toute la suite de ce mémoire, le terme *compte rendu* utilisé seul se référe au compte rendu d'examen fœtoplacentaire. Chacun de ces comptes rendus contient en moyenne 4 ou 5 pages, soit 500 mots au total, et chaque fichier texte de ACC correspond à une page dans les dossiers papier originaux.

3.1.2 Détail d'un compte rendu d'examen fœtoplacentaire

Comme évoqué en introduction, un compte rendu d'examen fœtoplacentaire est un texte Unicode avec des éléments de structure hiérarchique guidant la lecture. Dans le contexte d'Accordys, chacun d'entre eux est obtenu par la numérisation et l'anonymisation d'un compte rendu papier de l'hôpital Trousseau à Paris. Chaque compte rendu contient au plus 9 sections principales correspondant chacune à un examen :

— Résumé clinique ;
— Examen macroscopique externe du fœtus ;
— Autopsie ;
— Radiographies ;
— Examen histopathologique des viscères ;
— Examen macroscopique du placenta ;
— Examen histologique du placenta ;
— Examen cytogénétique ;
— Conclusion.

Chaque section peut contenir des sous-sections ou directement de l'information sous forme soit de phrases complètes, soit de listes à puces (énumérations ou paires *(site anatomique, observation)*). Bien qu'étant du même type, les documents n'expriment pas toujours les informations de la même manière, la *présence* et *l'ordre* étant souvent différents : la plupart des comptes rendus contiennent seulement une fraction de ces sections (par exemple l'examen cytogénétique est souvent absent et l'autopsie peut avoir été refusée par les parents), pas toujours sous le même nom et pas exactement dans le même ordre.

Ainsi, la spécificité de ACC est qu'il est intégralement composé de structures énumératives (SE), qui sont réparties en divers niveaux. Toutefois, les marqueurs et indices indiquant la présence d'une SE sont ici réduits. Nous avons :

— des titres de sections ;
— des puces et patrons ponctuationnels réalisant des listes d'éléments de données ;

— des énumérations de mesures ou d'observations, parfois sans aucun autre séparateur qu'un espace entre chaque.

Les expressions énumérantes (séquenceurs et connecteurs, tels que « premièrement », « tout d'abord », « d'une part », « puis », « ensuite », etc.) sont quant à eux quasiment absents. La difficulté de la reconnaissance automatique réside dans la non consistance des marqueurs utilisés. Une liste correspondant au même énuméraThème dans deux comptes rendus différents pourra être réalisée avec des marqueurs différents, voire même par moments sans aucun marqueur.

3.1.3 Obtention de ACC et nommage des fichiers

L'ensemble de ACC a été numérisé, anonymisé et rendu disponible par lots, quatre lots exactement. Nous notons ACC[X] le X^e lot et ACC[X,Y] tous les lots du X^e au Y^e. ACC[1,3] contient des cas allant de 1993 à 2006 et ACC[4] de 1986 à 1993.

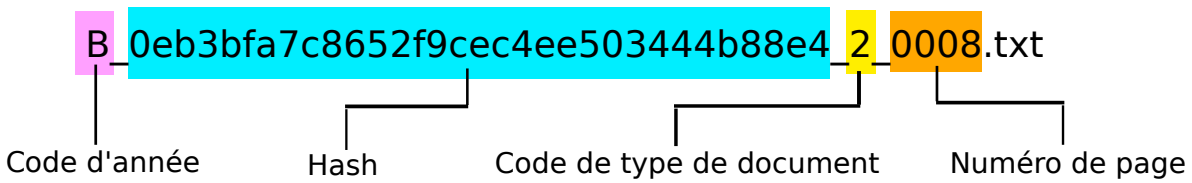
Chaque lot contient une successions de fichiers textes. Chaque fichier correspond à une page numérisée, et est identifié par un code d'année et un hash (un code hexadécimal), comme indiqué sur la figure 3.1. Tous les fichiers portant le même hash proviennent du même dossier papier et tous les fichiers ayant le même code d'année proviennent de dossiers rédigés originalement la même année. Le code de type de document peut-être 1 ou 2. 1 signifie qu'à la numérisation le fichier a été établi comme étant un brouillon, et 2 qu'il s'agit d'un compte rendu d'examen fœtoplacentaire ou de cytogénétique. Le numéro de page correspond à l'ordre dans lequel les pages ont été numérisées, il n'y a donc dans les noms des fichiers d'élément nous permettant de distinguer les fichiers provenant d'un compte rendu d'examen fœtoplacentaire et ceux venant d'un examen de cytogénétique.

À cette étape-là, les autres types de comptes rendus (échographie, IRM...) ont déjà été retirés et ne nous préoccupent donc pas. Cependant, des courriers entre praticiens sont toujours présents aussi bien avec le code 1 que le code 2. Sous un système Unix, la commande

```
$ rgrep "\bcher\b" -R corpus/
```

permet aisément d'en identifier au moins une partie, du fait de la prépondérance de l'expression « Mon cher confrère » dans ces documents.

FIGURE 3.1: Nommage d'un fichier de ACC



3.1.4 Qualité générale du corpus

En plus de la variabilité des marqueurs utilisés pour réaliser les énumérations d'éléments de données, ACC pose un problème quant à sa lisibilité : tous les documents n'ont pas la même qualité, et celle-ci est fonction de l'ancienneté des dossiers papier numérisés, puisque celle-ci détermine la qualité de l'impression.

La figure 3.2 montre la variabilité en termes de qualités qu'il peut exister. On estime que ACC contient environ 2 documents sur 15 d'une qualité comparable ou inférieure au document de gauche sur la figure. Une grande portion du corpus ne contient donc que peu d'erreurs d'OCR et est donc tout à fait exploitable, mais il existe une partie de ACC sur laquelle il est déjà à priori inutile d'espérer des résultats d'annotation sémantique exploitables, les méthodes et outils d'annotation actuels procédant par correspondance exacte (*exact match*).

FIGURE 3.2: Comparaison de la qualité d'un ancien compte rendu (à gauche) et d'un plus récent (à droite) après numérisation et anonymisation

Motif : IMG pour hydrocéphalie majeure
Sexe : ÀG : Poide
PRTNCIPÀLES CONSTÀTÀTTONS MÀCROSCOPIQIIES :

CERIIEIAU !
Àspect externe : hémisphères as ymé trique S, fluctuantes
Poïds : '72 g Mensuràtìons : OFD =58 mm Maturation 22s
OFG =65 mm
Mé.ninges : normales
Structures de la base : normales
TRONC EEREBR.JÀ,L

CERVELET : hypopla sie del 'hémisphère cérébelIeux droit réduit à 1,états
de moignon. Déviat l-on del 'hémisphère gauche.
: dilatation triventìculaire
COUPES FROIiIIÀIJES asymétrigue avec des-
truction des piliers du septum
PRTNCTPALES CONSTATATTONS MICROSCOPIQTTES :

Niveaux de préIèvements : hémisphère, Èronc, cervelet, yeux.
Le calibre de 1'aqueduc est réduit par rapport à un témoin
Confirmation de lihypoplasie unil-atêrale -àu cervelet et des noyaux
dentellés. Les pyramidês sont normaitr;r.
coNcr,usIoN 3

Dilatation Èriventriculaire avec hlpoplasie unilatérale du cerveLet.
Codificatibn NHNZI_43 0 NHNZ2110
LS VILLE-3037,le DÀTE-3781

III PRELEVEMENTS
. pour histologie : thyroïde, thymus, poumons, foie, rate, pancréas, reins, surrénales,
ovaire

POIDS : sans cordon ni membranes: 140 G
:
DIMENSIONS 14 X 10 x2 X CM
CONFIGURATION : placenta déchiqueté
CORDON :
. insertion centrale mais appliqué sur la plaque
  choriale sur ces 6 cm vus
. longueur vue en cm : 6
MEMBRÀNES :
PLAQUE BASÀLE : déchiquetée
TRANCHE DE SECTION : hydropique

EVALUATION GLOBALE DE LA CAPAGITE FONCTIONNELLE
PATHOLOGIQUE
ANOMALIE FUNICULAIRE

V EXAMEN HISTOPATHOLOGIQUE
a) des viscères
POUMONS DROIT ET GAUCHE : développement normal

REINS DROIT ET GAUCHE : développement normal

FOIE : développement normal - hématopoïèse active

RATE : développement normal

THYROIDE : développement normal

PANCREAS : développement normal

Un autre facteur limitant l'utilisabilité directe du corpus est la redondance de certaines informations : de nombreux dossiers papiers contiennent en effet des pages dupliquées avec de légères variations (cf. 5.1 pour une étude de la proportion de ces duplications dans ACC). Il s'agit de plusieurs versions différentes des comptes rendus qui ont toutes été imprimées et archivées ensemble dans le même dossier papier. À la numérisation du corpus, il a fréquemment été observé que les versions les plus récentes se retrouvaient au début dans le dossier, et que donc une fois numérisées elles avaient un numéro de page plus petit.

3.1.5 Fichiers de détection des duplications

À partir des deux premières livraisons du corpus ACC (notées ACC[1,2]), le projet Accordys a pu obtenir une liste des fragments de textes qui sont dupliqués au sein d'un même compte rendu. Cette liste de duplications a été originalement obtenue grâce à une méthode dont la publication est encore en attente. Cette méthode procède de manière très exhaustive, et détecte aussi bien les pages quasi-intégralement dupliquées, à quelques mots près, que les phrases qui se retrouvent quasiment à l'identique dans plusieurs pages d'un même dossier, sans que ces pages ne soient elles-mêmes à considérer comme des duplicatas. La sortie de cette méthode est un ensemble de fichiers XML, un par lot de livraison de ACC, dont un fragment est montré en figure 3.3.

FIGURE 3.3: Extrait d'un fichier XML contenant les duplications détectées au sein d'un même dossier

```
<folder name="B_03f637e6c2d7135a90673adb0b37955b">
  <detection>
   <passage document="2_0002.txt" end="270" offset="81">
   </passage>
   <passage document="2_0006.txt" end="189" offset="1">
   </passage>
   <score value="0.908046">
   </score>
  </detection>
  <detection>
   <passage document="2_0002.txt" end="517" offset="270">
   </passage>
   <passage document="2_0006.txt" end="436" offset="189">
   </passage>
   <score value="0.799197">
   </score>
  </detection>
```

Chaque entrée `detection` concerne deux passages dans deux documents différents (identifiés par code de type et numéro de page) d'un même dossier (identifié par code d'année et hash). Les positions de départ (`offset`) et de fin (`end`) de chaque passage sont indiquées. Un `score` entre 0 et 1 a été attribué à chaque `detection` en fonction de la ressemblance des deux passages.

3.2 Ontologies et terminologies de domaine

Aucune ontologie de la fœtopathologie n'existe à l'heure actuelle, mais les ressources termino-ontologiques (RTO) médicales sont nombreuses et un certain nombre de termes médicaux utilisés dans les comptes rendus peuvent être retrouvés dans ces RTO. Celles qui nous intéressent sont les suivantes :

— Le MeSH (Medical Subjects Headings) [1], qui contient des concepts médicaux, biologiques et anatomiques assez généraux ;
— HRDO (Human Rare Diseases Ontology) [2], une ontologie des maladies rares ;
— HPO (Human Phenotype Ontology) [3], qui concerne les anomalies phénotypiques recontrées dans les maladies chez l'homme.
— OMIM (Online Mendelian Inheritance in Man) [4], un catalogue du génome humain et des maladies génétiques ;
— OntoDPN, une ontologie du diagnostic pré-natal (Dhombres et al. 2010) ;
— FMA (Foundational Model of Anatomy) [5], l'ontologie la plus complète en matière d'anatomie. Elle ne couvre cependant que l'anatomie du corps adulte normalement constitué ;

Toutes ces ontologies sont disponibles ou ont été incorporées pour nos besoins à l'outil ECMT [6] (Extracteur de Concepts Multi-Terminologique) développé par l'équipe du CISMeF [7], du CHU Charles Nicolle à Rouen. Cet outil nous servira pour la phase d'annotation de notre corpus.

1. https://www.nlm.nih.gov/mesh
2. https://bioportal.bioontology.org/ontologies/HRDO
3. http://human-phenotype-ontology.github.io
4. http://www.omim.org
5. https://bioportal.bioontology.org/ontologies/FMA
6. http://ecmt.chu-rouen.fr/
7. http://www.chu-rouen.fr/cismef/

Chapitre 4

Méthode

Ce chapitre présente la méthodologie adoptée lors de la thèse. En introduction nous avons pu voir que la méthode finale devait tenir compte de trois aspects pour produire une similarité entre deux comptes rendus :

— la structure présente de façon latente dans un compte rendu ;
— les éléments pouvant être identifiés (annotés) par un concept médical présent dans une terminologie de référence ;
— les éléments de texte (notamment aux niveaux de granularités *fin* et *détaillé*) ne pouvant pas être associés à des termes ou concepts.

Ceci nous permet d'identifier plusieurs sous-tâches à adresser, chacune pouvant intervenir à plusieurs endroits de la méthode, et chacune pouvant être implantée de plusieurs manières différentes :

1. la segmentation d'un compte rendu en arbre (détection de la structure pour repérer les nœuds de l'arborescence) ;
2. l'annotation sémantique automatique (repérage des concepts potentiellement utilisés dans un compte rendu) ;
3. la mesure de similarité sémantiques (mesure d'une distance entre deux concepts trouvés dans les comptes rendus à l'issue de l'annotation sémantique) ;
4. la mesure des similarités textuelles (pour les parties ne pouvant pas être annotées) ;
5. la mise en correspondance de deux arbres (identification des similarités structurelles, établissement d'un *mapping* un-à-un entre les nœuds de deux arbres et calcul d'une valeur de similarité globale entre ces deux arbres).

Nous rajoutons également une tâche « zéro », la constitution d'un modèle de cas noté MOD. Celui-ci est également un arbre, et permet d'uniformiser les cas transformés en arbres en les mettant en correspondance avec le modèle. Dans la terminologie du RàPC, ceci signifie que l'on instancie le modèle de cas.

4.1 Méthodologie générale

On présente tout d'abord une vision globale de l'approche poursuivie pour permettre la comparaison de deux cas. Cette approche a adopté dès le début deux points de vue :

— Pour que la comparaison puisse avoir du sens, l'information contenue dans les cas doit être conservée de bout en bout ;
— Plus l'on peut homogénéiser la représentation des deux cas, plus leur comparaison sera pertinente et efficace.

Nous cherchons donc ici à poursuivre une méthode holistique, et non une extraction d'entités et de relations prédéfinies comme le font par exemple Hassanpour et Langlotz (2015) ou le projet MedOC (Baneyx 2007). La raison à cela vient de la nature des documents utilisés. Les comptes rendus d'examen fœtoplacentaire représentent déjà, comme évoqué en section 3.1, un condensé de l'information importante sur le cas, et chaque élément peut être important pour le praticien. De plus, l'approche poursuivie cherche à être plus souple et à nécessiter moins d'encodage manuel de règles et de relations entre les éléments ciblés par la recherche d'information que les approches basées sur des patrons pré-établis (Malaisé et al. 2004). Ceci permettra de faciliter sa généralisabilité à de futurs corpus de comptes rendus d'examens en fœtopathologie issus de différents hôpitaux, voire même à d'autres domaines médicaux que la fœtopathologie.

4.1.1 Construction d'un modèle de cas

Les fœtopathologistes de l'hôpital Trousseau suivent un protocole lors des examens fœtoplacentaires. Ce protocole comprend un formulaire de quelques pages qui doit servir de modèle au compte rendu. Comme nous l'avons dit en introduction, ce formulaire n'est pas suivi à la lettre à chaque fois, ne détaille pas le contenu des différentes sections (Radiographies, Autopsie, etc.) et ne spécifie que les observations les plus communément réalisées lors d'un examen fœtoplacentaire. Dans l'optique d'homogénéiser les cas avant de tenter de les comparer, un modèle sous forme d'arbre basé sur ce formulaire a été réalisé avec le concert des fœtopathologistes. Des renseignements sur le type ou le contenu possible de chaque nœud présent ont été ajoutés. Ainsi, le nœud `Autopsie` aura comme type `examen` et le nœud `Capacité fonctionnelle` aura pour fils les nœuds `Normale`, `Paranormale` et `Pathologique`. Chaque nœud a donc un label et éventuellement un type et si possible un concept associé dans une ontologie ou terminologie médicale de référence. Ces types serviront lors du mapping et de calcul de la similarité entre deux arbres. Dans la suite de ce document ce modèle sera simplement appelé *arbre modèle*. Cette étape est d'autant plus importante que contrairement par exemple à la radiologie (Hassanpour et Langlotz 2015) il n'existe pas en fœtopathologie de modèle d'information déjà établi.

4.1.1.1 Extrait du fichier Yaml contenant le modèle de cas

L'extrait suivant montre un extrait du modèle de cas MOD réalisé, ici présenté au format Yaml (ou YML) :

```yaml
Examen_macroscopique_du_placenta:
    Date: date
    Opérateur: texte_libre
    Poids: rationnel
    Dimensions:
        Longueur: rationnel
        Largeur: rationnel
    Configuration:
        choix: [Normale, Marginée, Extrachoriale, Circumvallée,
                Ronde, {Autre: texte_libre}]
    Cordon:
        Longueur_vue: rationnel
        Insertion:
            choix: [Centrale, Paracentrale, Marginale, Vélamenteuse,
                    Interposition]
    Membranes:
        choix_multiple: [Couleur_normale, Méconiales, Sales, Hémorragiques,
                         Épaisses, Teintées, Ivoirées,
                         Blanchâtres, Jaunâtres, Gélatineuses, Incomplètes]
    Plaque_choriale:
        Vaisseaux_congestion:
            choix_multiple: [Diffuse, Veines, Thrombose, Focale, Artères]
        Thrombose_sous_choriale:
            choix: [Absente, Quelques_mm, Quelques_cm, En_placards_disséminés,
                    Massive_récente, Massive_ancienne]
    Plaque_basale:
        valeur: [Normale, Lisse, Déchiquetée, Calcifiée, Incomplète, Blanche]
        Hématome_décidual_marginal:
            choix: [Simple, Avec_rupture_des_espaces_intervilleux, Taille]
        Hématome_décidual_basal:
            choix: [Simple, Avec_cupule, Taille]
    Tranche_de_section:
        valeur:
            choix: [Homogène, Non_homogène, À_lobules_apparents]
        Couleur:
            choix: [Normale, Foncée, Claire, Foncée_et_claire, Hydropique]
        Kystes_cytotrophoblastiques: texte_libre
        NIDF_villositaire:
```

```
            choix: [Marginale_normale, Marginale_étendue, Sous_choriale_étendue,
                    Juxtabasale, Diffuse_étendue]
        Thrombose_intervilleuse: texte_libre
        Infarctus:
            Rouge: &desc_infarctus
                Siège_marginal: texte_libre
                Partout: texte_libre
                Siège_central: texte_libre
                Siège_paracentral: texte_libre
                Diamètre: rationnel
            Blanc: *desc_infarctus
            Rouge_et_blanc: *desc_infarctus
        Volume:
            Inférieur_au: rationnel
            Supérieur_au: rationnel
    Autre_lésion_grave_exceptionnelle: texte_libre
    Évaluation_globale_de_la_capacité_fonctionnelle:
        choix: [Normale, Paranormale, Pathologique]
Examen_histologique_du_placenta:
    Date: date
    Opérateur: texte_libre
    Cordon: texte_libre
    Membranes:
        Épithélium_amniotique: texte_libre
        Mésenchyme_sous_amniotique: texte_libre
        Chorion: texte_libre
        Caduque: texte_libre
    Placenta:
        Épithélium_amniotique: texte_libre
        Chorion: texte_libre
        Sous_chorion: texte_libre
        Villosités: texte_libre
        Espaces_intervilleux: texte_libre
        Gros_troncs_villositaires: texte_libre
        Plaque_basale: texte_libre
    Conclusion: texte_libre
```

Il s'agit d'un arbre décrivant toutes les informations communément attendues, dans l'extrait présenté dans les sections concernant les examens macroscopiques et histologique du placenta. Les libellés commençant par une majuscule sont les labels qui seront comparés au contenu d'un compte rendu lors de la phase d'instanciation du modèle de cas, et ceux commençant par une minuscule sont des mots-clefs spéciaux renseignant sur le type de valeur attendu (`rationnel` ou `date` lorsqu'une donnée numérique est attendue, `texte_libre` lorsque l'élément de donnée est de nature qualitative) ou sur un

ensemble de valeurs pouvant fréquemment se trouver à cet endroit-là (`choix`, `choix_multiple`). Le mot-clef `valeur` indique que la section dans laquelle il se trouve est fréquemment associée à une valeur particulière mais qu'on y trouve aussi fréquemment des informations complémentaires.

MOD contient 597 nœuds se répartissant dans les trois catégories énoncées précédemment :

— Libellés nommant des sections ou des éléments de données : 304 nœuds ;
— Annotations de type : 201 nœuds ;
— Valeurs fréquemment attendues pour renseigner sur une observation donnée : 92 nœuds.

Une version annotée où les nœuds ne sont pas textuels mais conceptuels est également produite. Elle est notée MOD.A.

La notation `&étiquette` crée une référence vers un sous-arbre, et la notation `*étiquette` permet d'y faire référence ailleurs. Ceci permet d'indiquer que deux sous-sections contiennent les mêmes éléments de données, et sont donc usuellement renseignées de la même manière, sans avoir à explicitement tout recopier.

4.1.2 Élaboration de différentes méthodes d'évaluation de similarités à comparer

L'approche structurelle étant la principale contribution de ce travail de thèse, nous avons besoin pour l'évaluer de la mettre en regard de méthodes qui ne tiennent pas compte de la structure, afin de quantifier son apport. Selon l'utilisation qui est faite des sous-tâches évoquées en introduction au chapitre 4, on peut distinguer plusieurs méthodes d'évaluation de similarités entre deux comptes rendus, ou même entre un compte rendu et le modèle de cas :

— *MET.Sim.StructSem* : toutes les sous-tâches sont utilisées ;
— *MET.Sim.Struct* : on n'utilise pas l'annotation sémantique (et de ce fait pas les mesures de similarités sémantiques) ;
— *MET.Sim.Sem* : on n'utilise pas de détection de structure et de mise en correspondance d'arbres ;
— *MET.Sim.Txt* : on n'utilise ni annotation ni structure, la méthode utilise uniquement des mesures de similarités textuelles.

De plus, nous cherchons à évaluer l'apport que peuvent avoir les ontologies du domaine sur lequel nous travaillons. C'est pourquoi les approches *MET.Sim.Sem* et *MET.Sim.StructSem* peuvent encore être subdivisées selon les ontologies utilisées, car chaque ontologie va engendrer des résultats différents.

Établir des similarités dans un document non structuré (qu'il ait été annoté ou non) peut être fait au travers des méthodes vectorielles et/ou de *topic modelling*. Ceci nous permettra d'établir une *baseline* pour quantifier l'apport de notre contribution.

Le reste de ce chapitre présentera les composants de ces méthodes, et la manière dont ils s'agencent lors de la réalisation de chacune. La Figure 4.1 présente par exemple l'agencement et le paramétrage de

ces différents composants lors de la réalisation de *MET.Sim.StructSem* et le traitement d'une portion de compte rendu donnée (avec la portion de modèle correspondante) par cette méthode. Les composants découlent des sous-tâches identifiées en 4.

4.1.3 Constitution d'une base de cas

Cette partie de la méthode est le traitement du corpus ACC à proprement parler, elle a pour but de permettre la réalisation de *MET.Sim.StructSem*. Les grandes lignes sont ici de :

— grouper les fichiers du corpus par cas et par type de compte rendu, filtrer chaque cas pour conserver uniquement les documents propres aux traitements suivants. Il s'agit de garder parmi les documents uniquement ceux qui sont des CR d'examen fœtoplacentaire (CREF) et de retirer ceux qui contiennent trop d'erreurs de reconnaissance de caractères. Le résultat est un corpus filtré appelé ACC.F.
— annoter sémantiquement ACC.F pour permettre les comparaisons sémantiques (de *MET.Sim.Sem* et *MET.Sim.StructSem*). Le corpus produit en sortie est appelé ACC.A.
— segmenter les corpus ACC.F et ACC.A pour séparer les éléments de données constitutifs des CREF et identifier les niveaux de granularité auxquels ils appartiennent, afin de recréer la structure hiérarchique latente aux CREF. On obtient donc deux collections d'arbres : ARB où les noeuds des arbres sont uniquement des chaînes de caractères et ARB.A où chaque noeud a un ou plusieurs concepts ontologiques (issus de l'annotation sémantique) associés ;
— faire correspondre les arbres de ARB à MOD pour obtenir la collection ARB.H, et ceux de ARB.A à MOD.A pour obtenir ARB.A.H. Dans ces deux nouvelles collections, la structure de chaque arbre a été homogénéisée autant que possible sans perdre d'information.

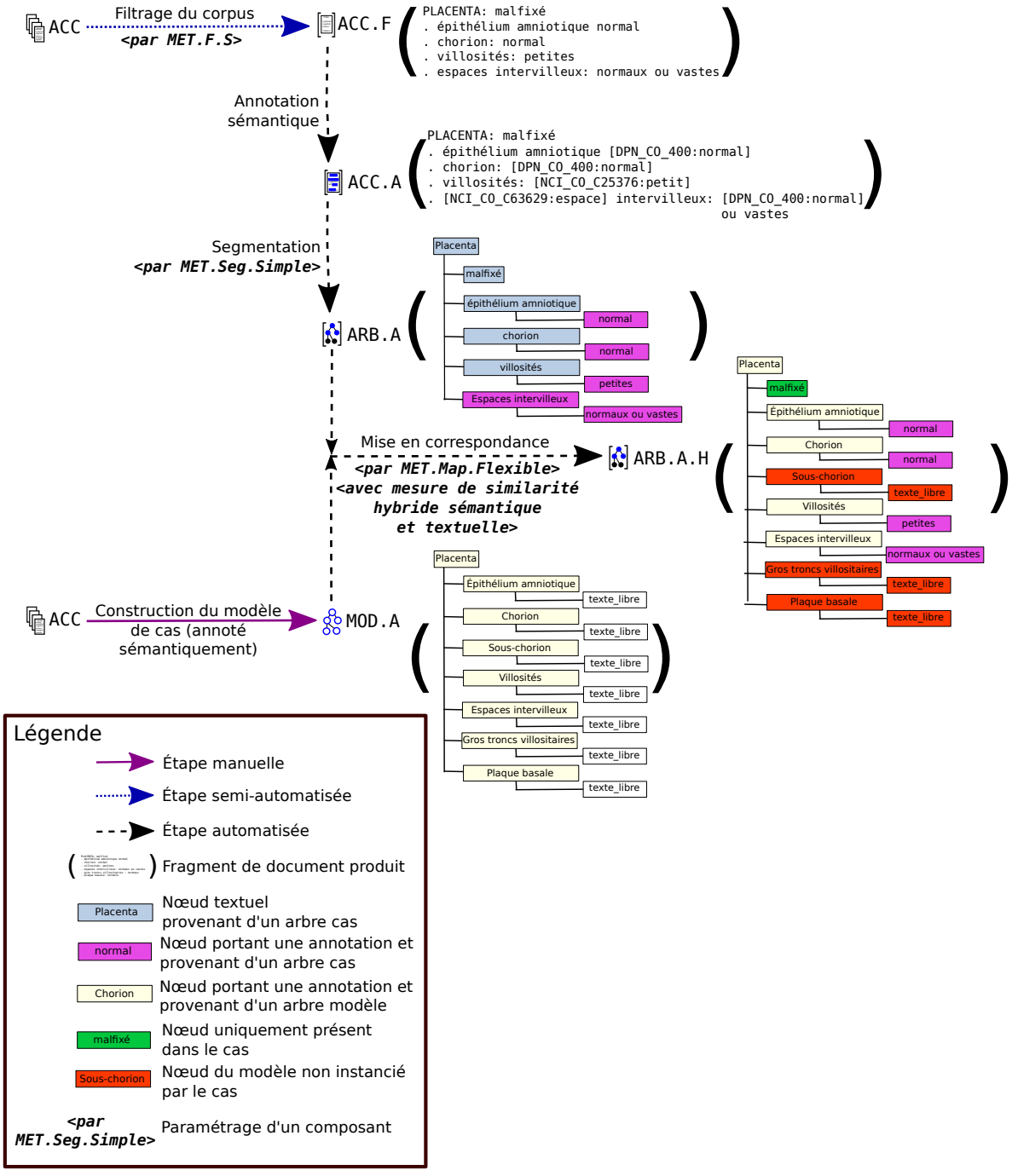
4.2 Mise au propre et filtrage du corpus d'Accordys

La première étape est de grouper par cas chacun des fichiers de chacun des lots de ACC. Ensuite, comme évoqué en section 3.1.4 dans la description du matériel utilisé pour cette thèse, l'intégralité du corpus ACC est numérisé depuis des dossiers papier. Selon l'ancienneté de ces dossiers, cette numérisation résulte en un nombre plus ou moins important d'erreurs d'OCR, celles-ci pouvant parfois aller jusqu'à rendre le fichier texte difficilement compréhensible pour un lecteur humain. De plus, nous nous intéressons uniquement au compte rendu d'examen fœtoplacentaire, ce qui représente une partie seulement des documents numérisés. Chaque fichier texte étant une page du dossier papier d'origine, nous utiliserons donc le mot *page* pour désigner chaque fichier. Chaque page devra donc être classifiée :

— comme étant utilisable ou non par le système final ;
— comme faisant partie ou non d'un compte rendu d'examen fœtoplacentaire ;

et seules les pages satisfaisant ces deux critères pourront être considérées par la suite. Le corpus résultant sera nommé ACC.F (pour *filtrage*).

FIGURE 4.1: Composants de *MET.Sim.StructSem*



Afin de classer les quelques 40 000 pages de ACC, nous procédons par apprentissage automatique. Deux méthodes seront comparées pour cette étape : l'une posant l'hypothèse que les pages sont indépendantes (*MET.F.I*) et l'autre partant du principe que la classification de chaque page peut aider à la classification de la page suivante, et considérant donc les pages de façon séquentielle (*MET.F.S*).

4.2.1 Sélection du corpus d'entraînement

Afin d'éviter une trop grande variance des paramètres, le corpus d'entraînement (noté ACC.E.F) sera le même pour les deux méthodes de filtrage, mais il devra donc contenir des pages séquentielles pour convenir à l'entraînement de *MET.F.S*. La taille moyenne d'un compte rendu d'examen fœtoplacentaire étant de 4 pages, nous sélectionnons aléatoirement des fenêtres de 6 pages (autrement dit des sous-ensembles de 6 pages consécutives) dans ACC, puis repassons manuellement sur chacune pour lui rajouter les pages nécessaires pour que la fenêtre ni ne démarre ni ne se termine en plein milieu d'un document. Ainsi, chaque fenêtre contiendra au final au minimum deux documents qui se suivent dans ACC, et ACC.E.F inclura aussi bien des successions de pages d'un même document que la transition d'un type de document à un autre.

Nous ne prenons pas un seul bloc contigu de ACC pour constituer ACC.E.F afin d'éviter un biais de sélection : les dossiers adjacents dans ACC sont ceux qui ont été numérisés en même temps, et qui souvent étaient stockés ensemble dans les archives de l'hôpital Trousseau, et sont donc la plupart du temps des dossiers rédigés à la même période.

4.2.2 Filtrage du corpus avec *MET.F.I*

Chaque page du corpus d'entraînement sera annotée avec l'un des libellés suivants :

CR Si la page de contient pas de fautes d'OCR nuisant à sa compréhension et si elle fait partie d'un compte rendu d'examen fœtoplacentaire ;
AUTRE Si la page est propre mais ne fait pas partie d'un CREF ;
IMPROPRE Si la page contient trop de fautes d'OCR.

L'apprentissage sera fait avec un classifieur Bayésien entraîné par maximum d'entropie, fourni par l'outil `dbacl` [1] qui sera également utilisé pour classifier le reste de ACC. Les caractéristiques de chaque page qui seront considérées par l'apprentissage et la classification sont l'ensemble des tétragrammes de caractères qu'elle contient.

4.2.3 Filtrage du corpus avec *MET.F.S*

Chaque page de ACC.E.F sera annotée avec l'un des libellés suivants :

1. http://dbacl.sourceforge.net

B_CR Si la page démarre un compte rendu d'examen fœtoplacentaire ;
I_CR Si la page continue un CREF ;
B_AUTRE Si la page démarre un document autre qu'un CREF ;
I_AUTRE Si la page continue un document autre qu'un CREF ;
IMPROPRE Si la page contient trop de fautes d'OCR.

Les caractéristiques de chaque page considérées sont toujours les tétragrammes qu'elle contient. La classification sera cette fois réalisée grâce à l'outil `wapiti` [2] par entraînement d'un CRF (*Conditional Random Field*) linéaire d'ordre 1. Wapiti (Lavergne et al. 2010) fournit un ensemble d'outils pour l'étiquetage de séquences par application de modèles discriminatifs (par opposition aux modèles génératifs), dont les CRF font partie.

4.3 Segmentation

Deux composants de segmentation sont proposés, *`MET.Seg.Simple`* et *`MET.Seg.Apprentissage`*.

4.3.1 *`MET.Seg.Simple`*

Le traitement se fait en deux temps, d'abord un typage des lignes du texte selon leur mise en forme matérielle (MFM), puis la construction de l'arbre à proprement parler.

4.3.1.1 Typage des lignes

La première étape est tout d'abord d'établir les caractéristiques d'une ligne, afin d'avoir une première idée de sa MFM, sous formes de simples valeurs booléennes :

— *marqueur*? : si elle démarre par un marqueur indiquant une structure énumérative de type liste à puces (tiret, point, astérisque) ;
— *séparateur*? : si elle contient un caractère « : », auquel cas nous appelons *label* le fragment de texte avant le « : » et *reste* ce qui se trouve après. Si la ligne de contient pas de « : », *label* correspond à l'intégralité de la ligne et *reste* est considéré comme vide ;
— *maj*? : si plus de 50% des caractères du *label* sont alphabétiques, et s'ils sont tous en majuscules.

Pour toute chaîne de caractères S, *vide*?(S) est vrai si S est vide ou ne contient que des espaces blancs.

Ensuite, un ensemble de règles simples basées sur des patrons portant sur la forme de la ligne servent à déterminer à quel niveau de granularité on considère qu'elle se place (et donc les types des nœuds

2. https://wapiti.limsi.fr

qu'elle génèrera et leur placement dans l'arbre). Pour chaque ligne, nous vérifions les clauses suivantes dans l'ordre et conservons la première qui s'avère vraie :

$\neg marqueur? \wedge maj? \wedge vide?(reste) \rightarrow$ on considère qu'il s'agit d'un titre soit du *niveau de granularité grossier* (titre de section correspondant à un examen) soit du niveau *intermédiaire* (sous-section), mais nous ne pouvons pas trancher dès maintenant ;

$séparateur? \wedge \neg vide?(reste) \rightarrow$ il s'agit soit d'un titre du niveau *intermédiaire* soit d'un nom d'attribut appartenant au niveau *fin*

$séparateur? \rightarrow$ il s'agit d'un titre au niveau *intermédiaire*

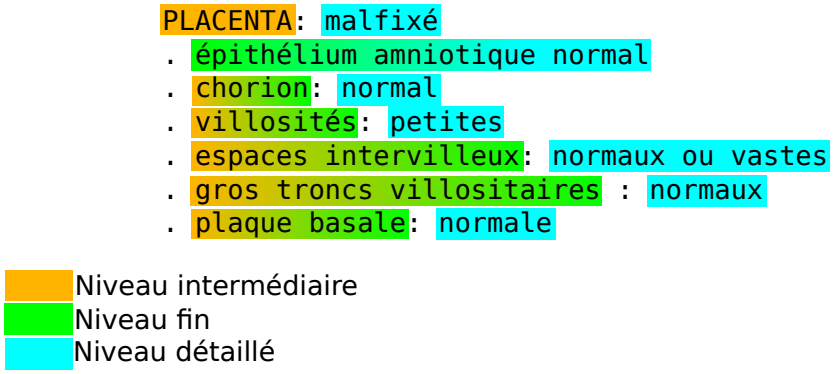
$marqueur? \rightarrow$ il s'agit d'un item au niveau *fin* ou d'un fragment de texte au niveau *détaillé*.

Si rien ne correspond, alors le cas par défaut est de considérer qu'il s'agit d'un fragment de texte que l'on placera au niveau *détaillé*. Dans tous les cas, *reste* est considéré comme étant un nœud appartenant au niveau *détaillé*.

La fonction Clojure présentée en annexe (section 7.1.1) montre l'implantation de ce typage des lignes, et la figure 4.2 en montre un exemple. Les dégradés indiquent une incertitude quand au niveau auquel appartient l'élément de donnée considéré.

Ce traitement est volontairement superficiel, afin de ne pas risquer d'être trop dépendant des spécificités de chaque document, et de ne pas avoir à couvrir toutes les alternatives de MFM pour un niveau de granularité donné. De plus, certaines MFM sont les mêmes dans des documents différents pour des informations qui n'appartiennent pas aux mêmes niveaux de granularité. Ce traitement fonctionne grâce à des heuristiques de classification qui sont issues de l'observation de ACC. Elles sont donc conçues avec ce corpus en tête, mais restent assez flexibles et leur simplicité rend leur adaptation assez facile.

FIGURE 4.2: Classification des lignes de l'extrait de la figure 1.2



4.3.1.2 Construction de l'arbre à partir du texte

À cette étape, on va générer à partir de chaque ligne un ou deux noeuds, auxquels sera éventuellement donné un type parmi *examen* (niveau *grossier*), *soussection* (niveau *intermédiaire*), *attribut* ou *item* (niveau *fin*). Les nœuds vont être placés dans l'arbre en fonction de la classification de la ligne dont ils sont issus et des types du/des noeuds générés par la ligne précédente. Les noeuds restant non-typés sont considérés comme étant des descriptions appartenant au niveau *détaillé* dont la MFM ne sera pas analysée plus avant.

Par exemple, la ligne contenant juste `DISSECTION`, entièrement en lettres capitales, aura été reconnue comme titre de section au niveau *grossier* ou au niveau *intermédiaire* et génèrera un nœud de type `examen`, et la ligne

```
- Foie : développement normal
```

ayant été reconnue comme une paire *(site anatomique, observation)* au niveau *fin*, génèrera le nœud `Foie` et son fils `développement normal`, tous deux non typés. Tous les arbres produits sont placés dans une base de données documentaire, telle que MongoDB [3] qui stocke les arbres sous une forme binaire reprenant le modèle de données de JSON [4].

Ce traitement se fait donc encore une fois ligne par ligne, mais n'est pas cette fois sans mémoire puisqu'il est conditionné par les noeuds générés à partir de la ligne précédente.

4.3.2 *MET.Seg.Apprentissage*

Exposés initialement par Lafferty et al. (2001), les *Conditional Random Fields* sont un modèle statistique discriminatif qui peut être utilisé lors d'une tâche de segmentation (par exemple d'un texte ou d'une image) afin d'apprendre une distibution de probabilité $p(Y|X)$ où, pour nous, X est la séquence des caractéristiques (*features*) de chaque élément de données relevé dans le texte et Y est la séquence des libellés renseignant sur la position relative ou absolue de ces mêmes éléments dans la structure de l'arbre.

À partir d'un corpus de texte d'apprentissage où l'on fournit aussi bien les caractéristiques de chaque mot que les libellés de position, cette distribution est apprise et permet de segmenter les éléments de données contenus dans le reste du corpus et de les replacer dans les différents niveaux de granularité (niveau *grossier*, niveau *intermédiaire*, etc.).

Cependant, avant de donner les caractéristiques de chaque élément d'une séquence issue d'un texte, nous devons déjà établir une première manière de séparer les différents éléments d'un texte. Ceci s'appelle la *tokenisation*. Un token est donc associé à un texte (court) et à une position dans le document.

3. https://www.mongodb.com
4. JavaScript Object Notation. Voir www.json.org/xml.html

Plus fondamentalement, nous définissons la segmentation comme l'utilisation de deux fonctions, *tokenisation* et $features$. On définit la première ainsi :

$$tokenisation : Document \rightarrow [Token]$$

où la notation [...] désigne une séquence ou une liste.

On pourrait être tenté de définir la seconde comme ceci :

$$features : Token \rightarrow Features$$

et ainsi de simplement appliquer $features$ sur chaque token obtenu. Ceci ne permet toutefois pas de définir les caractéristiques d'un token en fonction des tokens environnants.

$features$ sera donc définie ainsi :

$$features : ([Token], Token, [Token]) \rightarrow Features$$

Où le domaine de la fonction est un token et son voisinage à gauche et à droite.

On définit pour chaque token les caractéristiques suivantes (qui généralisent celles énoncées pour les lignes au 4.3.1.1) regroupées sous le type $Features$:

— le texte correspondant au token en lui-même ($TexteCourt$)
— si ce texte est intégralement en majuscules ou non ($Booléen$)
— si le token est précédé d'un saut de ligne ou non ($Booléen$)
— si le token est précédé d'une indentation (alinéa) ou non ($Booléen$)
— si le token est suivi de deux-points (« : ») ($Booléen$)
— si le token est précédé d'un marqueur de structure énumérative (point, tiret, virgule, etc.) ($Booléen$)

Les libellés indiquant l'emplacement dans la structure hiérarchique de chaque token sont les suivants :

BEGIN_TOP_SECTION Indique que le token démarre un en-tête de section, et donc appartient au niveau *grossier* ;
BEGIN_SUBSECTION Indique que le token démarre un en-tête de sous-section, et donc appartient au niveau *intermédiaire* ;
INSIDE_SECTION_HEADER Indique que le token est la continuation d'un en-tête, et donc appartient au même niveau de granularité que le token précédent ;
FIRST_CHILD Indique que le token démarre le premier fils du token précédent, et appartient donc soit au niveau *fin*, soit au niveau *détaillé* ;
BEGIN_BROTHER Indique que le token démarre un nouveau nœud, frère du dernier nœud établi (nœud dans lequel se trouvait donc le token précédent) ;
LAST_CHILD Indique que le token démarre le dernier nœud de sa fratrie, et donc que le nœud suivant remontera d'au moins un niveau de granularité ;

ONLY_CHILD Indique que le token démarre un noeud enfant unique (c'est une combinaison de FIRST_CHILD et LAST_CHILD) ;
INSIDE_NODE Indique que le token se trouve dans le même nœud que le token précédent.

Ce que nous cherchons donc ici à faire est de segmenter les *nœuds*. Un exemple est montré en figure 4.3 dans le cas où l'on choisit de faire correspondre les tokens aux mots du texte (et où la séparation est donc réalisée sur les espaces).

On remarque donc que cette stratégie de labélisation n'impose pas une correspondance 1 à 1 entre tokens et nœuds. En effet, à ce stade du travail, nous traitons encore l'entité « $Token$ » comme quelque chose d'abstrait, dont nous savons qu'elle *correspond* à une portion de texte courte (nous avons donc une application $contenuDuToken : Token \rightarrow TexteCourt$) et à une position dans un document mais qui peut également être sémantiquement enrichie. En revanche, un token ne peut *jamais* se retrouver à cheval sur deux nœuds, on a donc également une application injective $contenuDuNoeud : Noeud \rightarrow [Token]$. Nous reviendrons là-dessus dans la section 4.4.

Le but de la segmentation est donc d'apprendre la fonction :

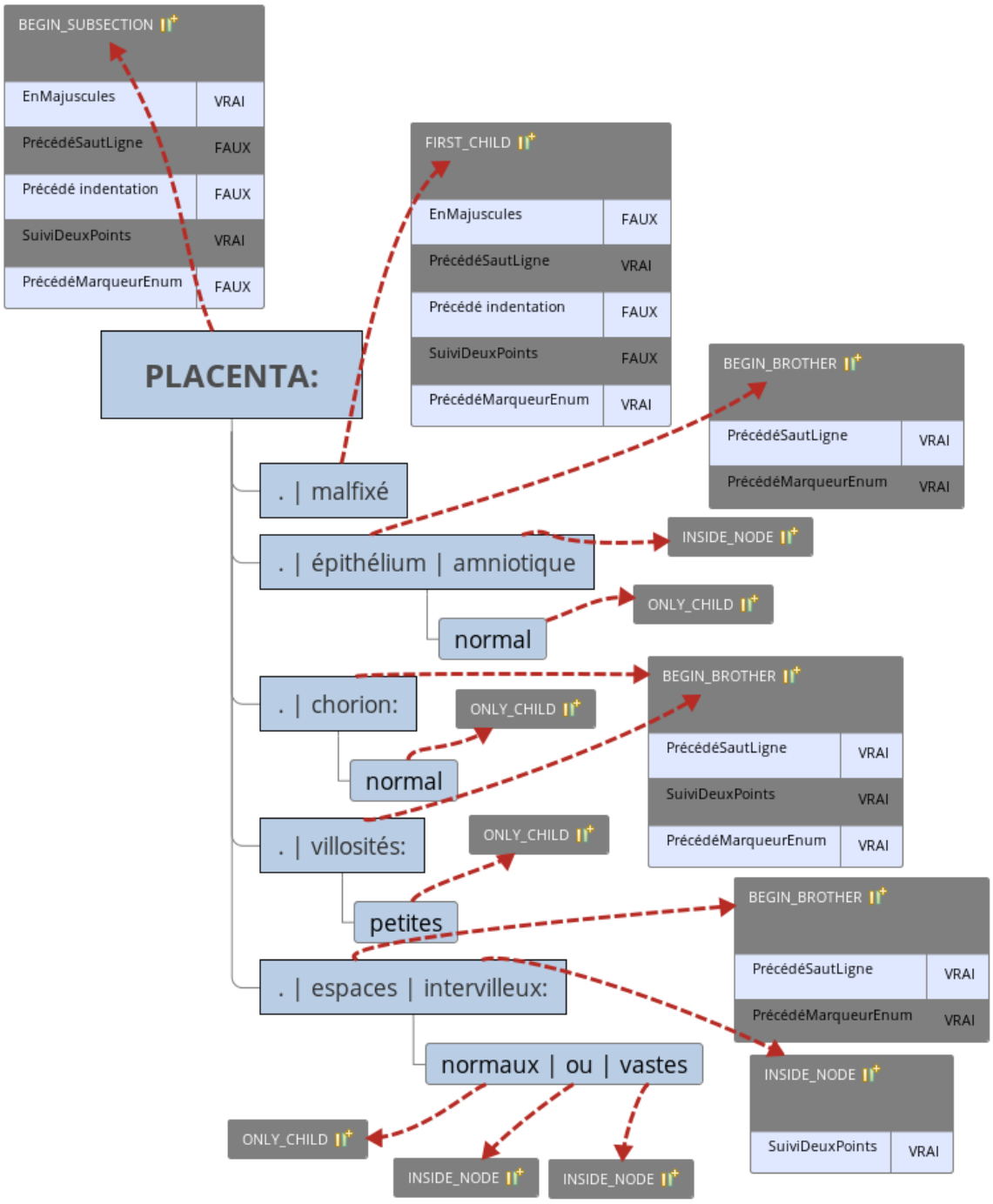
$$labelisation : [(Token, Features)] \rightarrow [Label]$$

et une fois que chaque token est associé à un label, nous pouvons recombiner les tokens pour former des nœuds et recréer la hiérarchie de ces nœuds, en partant d'un arbre n'ayant qu'un nœud racine *« Compte rendu d'examen fœtoplacentaire »* et en ajoutant :

1. des nouveaux nœuds aux niveaux identifiés de manière absolue par les libellés (cas des tokens classés comme **BEGIN_TOP_SECTION** ou **BEGIN_SUBSECTION**), chaque nœud ajouté devenant toujours le frère le plus à droite des nœuds déjà présents à ces niveaux,
2. des nouveaux nœuds comme fils ou frère du dernier nœud produit (**FIRST_CHILD**, **ONLY_CHILD**, **BEGIN_BROTHER** ou **LAST_CHILD**), ou
3. des tokens au dernier nœud produit afin de mettre à jour le *contenuDuNoeud* (**INSIDE_SECTION_HEADER** ou **INSIDE_NODE**).

Pour cette étape, nous allons sélectionner aléatoirement 50 comptes rendus fœtoplacentaires de `ACC.F` et annoter manuellement chacun des tokens (les mots et les signes de ponctuation dans notre cas) avec les libellés présentés précédemment. Les caractéristiques peuvent être trouvées automatiquement. Nous constituons donc un fichier d'apprentissage par dossier, chaque ligne contenant le même nombre d'éléments : un token, son libellé et ses caractéristiques, et utilisons l'outil `wapiti` déjà évoqué en section 4.2.3 pour apprendre un modèle de CRF puis l'utiliser pour classifier les tokens du reste de `ACC.F`.

FIGURE 4.3: Segmentation de l'extrait de corpus présenté en figure 1.2.



Le texte annoté est présenté sous forme d'arbre. Au sein d'un même noeud, les différents tokens sont séparés par des barres verticales. Afin d'alléger le schéma, les caractéristiques qui ne sont pas montrées doivent être comprises comme étant égales à FAUX. Les caractéristiques des tokens correspondant uniquement à des signes de ponctuation ne sont pas montrées.

4.4 Annotation sémantique automatique

Cette étape consiste à obtenir un enrichissement sémantique de notre corpus. On cherche à connaître tous les concepts pouvant correspondre aux mots ou groupes de mots présents dans le document. On va donc ajouter aux fonctions précédemment établies la suivante :

$$enrichissement : [Token] \rightarrow [Concept\,OU\,Token]$$

Le but de l'annotation est de fournir tous les concepts potentiels correspondant à un fragment de texte (quelques tokens qui se suivent dans le texte, donc), et de signaler lorsqu'un token ou succession de tokens ne correspond à aucun concept (d'où le fait que le type de retour d'*enrichissement* soit une liste où chaque élément est soit un concept détecté, soit un token provenant du texte original). Ici, nous faisons annoter notre texte par l'outil ECMT du CISMeF, dont les avantages sont les suivants :

— Il utilise la base terminologique du CISMeF, qui contient une bonne partie des RTO médicales principales à l'heure actuelle, et il nous est possible d'y ajouter une nouvelle ontologie (OntoDPN, l'ontologie du diagnostic prénatal) ;
— De nombreux labels de ces ontologies de référence ont été traduits en français notamment par les membres du CISMeF. Ces traductions sont pour nous une nécessité pour le processus d'annotation ;
— Des alignements (manuels ou automatisés) entre les RTO ont été réalisés par le CISMeF. Ils s'ajoutent à ceux disponibles dans l'UMLS et permettre de connaître pour un concept donné les concepts équivalents ou proches dans d'autres RTO.

À partir de `ACC.F` nous avons donc un nouvel état du corpus `ACC.A` qui est composé d'un ensemble de fichiers texte à côté desquels se trouvent les fichiers XML contenant les annotations correspondantes renvoyées par ECMT.

Voici un extrait de compte rendu :

```
b) du placenta
cordon : normal
membranes:
. épithélium amniotique: normal
. mésenchyme sous amniotique : normal
. chorion : normal
. caduque: normale
```

Voici ce que renvoie l'outil d'annotation pour la première occurence de `placenta` dans ce fragment de compte rendu :

```
<cis:indexation end="175" idterm="C13272" idcismef="NCI_CO_C13272"
 offset="167" start="167" ter="NCI" umlscui="C0032043"
```

```
 matchterms="placenta" typeid="T_DESC_NCIT_CODE">
  <cis:term>
    <cis:label lang="fr">placenta</cis:label></cis:term>
  <cis:categorization>
    <cis:category idcategory="T018" idcismef="UML_ST_T018" origin="umls">
      <cis:label lang="fr">structure embryonnaire</cis:label></cis:category>
  </cis:categorization>
  <cis:ancestors>
    <cis:ancestor idcismef="NCI_CO_C12219">
      <cis:label lang="fr">
        structure, système ou substance anatomique</cis:label></cis:ancestor>
    <cis:ancestor idcismef="NCI_CO_C34144">
      <cis:label lang="fr">
        structure ou système embryologique</cis:label></cis:ancestor>
    [...]
  </cis:ancestors>
  <cis:relateds>
    <cis:related relationTypeId="T_REL_PTS_TO_PTS_VALID"
      relationLabel="Alignements automatiques CISMeF supervisés"
      idcismef="MSH_D_010920">
      <cis:label lang="fr">placenta</cis:label></cis:related>
    [...]
  </cis:relateds>
</cis:indexation>
```

On observe ici plusieurs résultats :

- L'indexation en elle-même (`cis:indexation`) : le concept « placenta », sa provenance (`ter="NCI"`) : le thésaurus du National Cancer Institute (NCIt), son indentifiant dans le NCIt (`C13272`), et sa position dans le fichier original (attributs `start` et `end`). L'`idcismef` est toujours de la forme « IdentifiantTerminologie_CodeCismef_IdentifiantConceptDansSaTerminologie » ;
- Sa catégorie (`cis:category`) établie par le CISMeF, ici « structure embryonnaire » ayant pour origine un concept de l'UMLS (identifiant `T018` dans l'UMLS) ;
- Les ancêtres (`cis:ancestors`) de ce terme dans le NCIt avec leurs `idcismef` desquels on peut retrouver leurs identifiants dans le NCIt ;
- Les termes et concepts reliés (`cis:relateds`) dans d'autres terminologies et ontologies que le NCIt avec le type de relation dont il s'agit (alignement automatique, supervisé ou non, alignement fait à la main, etc.), encore une fois avec les `idcismef` de ces concepts. On voit donc ici que le terme du MeSH d'identifiant `010920` (labellisé aussi « placenta ») est aligné avec `C13272` dans le NCIt.

ECMT est normalement un outil en ligne, sous forme d'un webservice, mais pour les besoins du projet Accordys les comptes rendus sont annotés sur place, dans les locaux du CISMeF à Rouen.

4.5 *MET.Sim.Txt* et *MET.Sim.Sem* : comparaison par modèle vectoriel

Une fois les comptes rendus annotés et ACC.A obtenu, on constitue quatre enrichissements sémantiques différents de ACC.A :

- ACC.A.TI contenant le texte où les parties qui ont pu être annotées par l'ECMT ont été remplacées par les idcismef des concepts des indexations renvoyées par l'ECMT ;
- ACC.A.TIA contenant le texte, les idcismefs des concepts des indexations et de leurs ancêtres (renvoyés par l'ECMT également) ;
- ACC.A.I contenant uniquement les indexations et plus aucun fragment de texte des comptes rendus originaux ;
- ACC.A.IA contenant uniquement les indexations et les ancêtres.

chacun contenant un unique fichier par compte rendu. À titre d'exemple :

```
I . EXAMEN MACROSCOPIQUE
- fœtus de sexe maGêùlin
- état fixé dans !e formol
- macération moyenne
- poids 376 G
- mensurations PC 18,S CM plED 3,S CM
```

devient donc (en conservant les labels pour la lisibilité) dans ACC.A.TI :

```
I . EXAMEN MACROSCOPIQUE
- fœtus de [NCI_CO_C17127:sexe] [MSH_D_012723:sexe] [NCI_CO_C28421:sexe]  maGêùlin
- [NCI_CO_C25688:état] [DPN_CO_468:état] [NCI_CO_C25687:Etat] [NCI_CO_C87194:Etat]
  fixé dans !e [SNO_NO_C-21402:formaldéhyde]
- [SNO_NO_M-54315:macération]  [NCI_CO_C37917:moyenne] [NCI_CO_C53319:moyenne]
- [MSH_D_001835:poids_du_corps] [NCI_CO_C48192:poids] [NCI_CO_C25208:Poids]  376 G
- [MSH_D_049628:mensurations]  PC 18,S CM plED 3,S CM
```

ACC.F et les deux versions enrichies contenant du texte (ACC.A.TI et ACC.A.TIA) sont tokénisées et racinisées afin de réduire les variantes lexicales. Trois étapes sont ensuite réalisées sur chacun :

1. Création d'un modèle vectoriel pondéré par TF-IDF, et obtention d'une matrice termes/documents ;
2. Application d'une indexation sémantique latente (LSI, cf. section 2.1.3) pour réduire la dimension de cette matrice et établir des groupements thématiques de termes.

3. Obtention d'une matrice de similarités cosinus S entre les différents vecteurs. S est donc une matrice carrée d'autant de lignes qu'il y a de comptes rendus dans ACC.F.

On aura donc produit cinq matrices de similarité :

— S_T depuis le texte seul de ACC.F,
— S_{TI} depuis ACC.A.TI,
— S_{TIA} depuis ACC.A.TIA,
— S_I depuis ACC.A.I et
— S_{IA} depuis ACC.A.IA.

La raison de l'utilisation d'une LSI vient d'une étude préalable effectuée sur un fragment de ACC de 74 comptes rendus : les scores de similarités cosinus se distinguaient peu sur des vecteurs seulement pondérés par TF-IDF. Ceci vient du fait que dans ACC, les documents se différencient sur peu de mots, et la haute dimensionnalité des données impactait les résultats. Appliquer une LSI augmentait les écarts de score. Nous utilisions une LSI avec 25 topics (dimensions) lors de cette étude. La taille de notre corpus étant maintenant plus importante, il nous faudra déterminer le nombre de topics adéquat.

Nous chercherons ensuite à déterminer la contribution à S des indexations et des ancêtres par rapport au texte seul en observant les correlations entre les distances observées (plus de détails à ce sujet en section 4.7.1). Comme indiqué précédemment (cf. section 4.1.2) ces mesures de similarités serviront de *baseline* aux méthodes tenant compte de la structure des documents.

4.6 *MET.Sim.Struct* : mise en correspondance d'arbres

Le but ici est de déterminer pour un arbre cas lesquels de ses nœuds (et combien) peuvent trouver une correspondance dans un autre arbre. Cette mise en correspondance a deux utilités :

— Établir les correspondances structurelles entre deux arbres,
— Obtenir une mesure de similarité entre ces deux arbres.

Elle peut être utilisée lors de deux tâches :

— Instancier le modèle de cas avec un arbre cas issu de l'application de *MET.Seg.Simple* (4.3.1) à un compte rendu,
— Comparer deux arbres cas ensemble lors de la réalisation de *MET.Sim.Struct* et *MET.Sim.StructSem*.

Trois méthodes sont exposées ici : *MET.Map.Flexible*, *MET.Map.Inst* (instanciation) et *MET.Map.Hybride*. Ces méthodes sont toutes paramétrables dans le sens où elles abstraient la comparaison des nœuds en eux-mêmes. Si Γ est un ensemble de symboles, on note $A(\Gamma)$ l'ensemble des arbres labellisés avec les symboles de Γ. Étant donné deux ensembles Γ_1 et Γ_2, les trois méthodes requièrent comme paramètre une fonction :

$$distLabels : \Gamma_1 \times \Gamma_2 \longrightarrow \mathbb{R}^+$$

distLabels sera utilisée pour évaluer les distances entre les labels des arbres dans les trois méthodes. On note $Ns(\Gamma)$ un ensemble de nœuds provenant d'un $A(\Gamma)$. Chaque méthode prendra la forme d'une fonction de mapping, dont la forme la plus générale est la suivante :

$$mapping : \big(\Gamma_1 \times \Gamma_2 \to \mathbb{R}^+\big) \longrightarrow \big(A(\Gamma_1) \times A(\Gamma_2) \to A(\Gamma_1 \times \Gamma_2 \times \mathbb{R}^+) \times Ns(\Gamma_1) \times Ns(\Gamma_2) \times \mathbb{R}^+\big)$$

où l'argument de *mapping* est la fonction *distLabels*, et où $mapping(distLabels)$ est elle-même une fonction de deux arbres dont les résultats sont, dans l'ordre :

— l'arbre résultant, portant les libellés des nœuds mis en correspondance ainsi que leur distance,
— les nœuds du premier arbre n'ayant pas pu être mappés,
— les nœuds du second arbre n'ayant pas pu être mappés,
— la distance entre les deux arbres.

Un concept intéressant pour faciliter le parcours d'arbres est celui de *zipper* (Adams 2007). Un zipper est un outil permettant d'explorer et de modifier une structure de données, linéaire ou arborescente, en conservant l'endroit (locus) où l'on se trouve dans la structure. Dans le cas d'arbres, un zipper pointe donc toujours vers un nœud d'un arbre donné. Rosskopf (2015) montre une opérationalisation des zippers dans le langage de programmation Haskell ainsi que la manière dont on peut les concevoir comme des dérivées de types de données algébriques. On nomme $Z(\Gamma)$ l'ensemble de tous les zippers sur les arbres de $A(\Gamma)$. ψ dénote un locus vide, qui n'existe pas dans les arbres de $A(\Gamma)$. On a les opérateurs suivants :

— $zipper : A(\Gamma) \longrightarrow Z(\Gamma)$. Pour un nœud a donné, $zipper(a)$ pointera sur a et le prendra comme racine ;
— $noeud : Z(\Gamma) \longrightarrow A(\Gamma)$ donne le nœud, et donc le sous-arbre, actuellement pointé par un zipper ;
— $racine : Z(\Gamma) \longrightarrow A(\Gamma)$ remonte à la racine et donne donc l'arbre entier ;
— $haut, bas, droite, gauche : Z(\Gamma) \longrightarrow Z(\Gamma) \cup \{\psi\}$ permettent de se déplacer dans la structure. Si $noeud(z)$ a des fils, $bas(z)$ pointera vers le fils le plus à gauche, sinon $bas(z) = \psi$. De même si $noeud(z)$ n'est pas la racine de l'arbre dans lequel ce nœud se trouve, $haut(z)$ pointera vers son parent, sinon $haut(z) = \psi$. $droite(z)$ et $gauche(z)$ permettent quant à eux d'explorer les frères de $noeud(z)$;
— $parcoursP : Z(\Gamma) \longrightarrow Z(\Gamma) \cup \{\psi\}$, tel que $parcoursP(z)$ donne le locus qui suit le locus z lors d'un parcours en profondeur de l'arbre ;
— $edition : (A(\Gamma) \to A(\Gamma)) \times Z(\Gamma) \longrightarrow Z(\Gamma)$ permet de changer le libellé ou même le sous-arbre du nœud pointé par un zipper en y appliquant une fonction.

4.6.1 *MET.Map.Flexible*

Nous utilisons ici le *flexible tree matching* (FTM) de KUMAR, TALTON et al. (2011) afin de fournir la fonction de mapping, qui s'appelera ici $mapping_{ftm}$. Ici, seule $distLabels$ peut être biaisée ou non, par exemple en présupposant que le premier arbre est issu d'un cas et le second du modèle. Ceci implique que si $\Gamma_1 = \Gamma_2$ et que $distLabels$ est commutative, alors $mapping_{ftm}(distLabels)$ procèdera de manière non biaisée et sera agnostique de quel arbre est son premier argument et quel arbre est son second.

4.6.1.1 Fonctionnement de l'algorithme

Le FTM procède sur la base d'un algorithme de Metropolis-Hastings. Le processus est itératif. Étant donné un mapping M_p initial (une séquence de paires de nœuds représentant chacune un arc entre les deux arbres, ou bien entre un nœud d'un des deux arbres et le nœud vide ψ) et un ensemble $S_{mappings}$ initialement vide, on va à chaque itération :

1. Remplacer une partie du mapping précédent M_p par de nouveaux arcs. La taille de la portion à remplacer est aléatoire et suit une distribution uniforme. Une borne basse et une borne haute en terme de coût sont calculées pour chaque nouvel arc potentiel, et plus la moyenne de ces deux bornes est faible pour un arc potentiel, plus il a de chances d'être sélectionné.
2. Calculer le coût total du nouveau mapping obtenu M_n, et ajouter M_n à $S_{mappings}$.
3. Déterminer si M_p devient M_n ou s'il conserve son ancienne valeur, autrement dit si l'on continue l'itération à partir de M_n ou de M_p. La probabilité de choisir M_n plutôt que M_p est égale au ratio $\frac{e^{-\beta.cout(M_n)}}{e^{-\beta.cout(M_p)}}$. β est une constante, un paramètre de l'algorithme. La valeur conseillée par KUMAR, TALTON et al. (2011) est $\beta = 1$. Ainsi, si M_n est moins coûteux que M_p, il est certain d'être choisi, mais s'il est plus coûteux, on ne repart pas forcément de M_p pour autant, afin d'éviter un optimum local.

Le mapping initial peut être obtenu en fixant des arcs aléatoirement entre les arbres. À la fin des itérations, on prend le mapping de coût minimal parmi tous ceux qui ont été produits et ajoutés à $S_{mappings}$.

Nous prenons dans la suite de cette section l'exemple d'une mise en correspondance avec l'arbre modèle. La tâche ici est de trouver les paramètres du processus de calcul du mapping qui donnent les meilleurs résultats dans notre contexte de comptes rendus de fœtopathologie. Comme exposé dans l'état de l'art, le *flexible tree matching* vise le mapping de moindre coût. Le coût total d'un mapping est calculé en faisant la somme des coûts de chacun de ses arcs et en la divisant par la somme de la taille des deux arbres. La fonction calculant le coût de mise en correspondance des labels est indépendante du mapping. Le coût minimal permet de déterminer à la fois :

— le meilleur mapping (celui de coût minimal) ;
— la dissimilarité entre les deux arbres (le coût en lui-même).

Notons toutefois qu'étant donné qu'il s'agit d'un algorithme de Monte Carlo, nous n'avons pas de

garantie de converger lors de deux exécutions sur le même mapping et le même coût minimal finaux. Toutefois, un grand nombre d'itérations peut permettre d'affiner le mapping.

4.6.1.2 Coût de mise en correspondance des labels des nœuds

KUMAR, TALTON et al. (2011) nomme *distLabels* le *relabeling term*, noté $c_r([n_1, n_2])$, à savoir le coût induit dans un mapping M par la distance entre le label d'un nœud n_1 et le label d'un nœud n_2, si l'arc $[n_1, n_2]$ fait partie de M. Deux sous-cas sont à distinguer ici, celui de ***MET.Sim.Struct***, qui ne peut opérer que sur du texte, et celui de ***MET.Sim.StructSem***, qui doit opérer à la fois sur du texte et sur des concepts médicaux.

4.6.1.2.1 Cas de *MET.Sim.Struct* : Ici, aucune annotation préalable n'est faite sur le compte rendu et le coût de mise en correspondance de deux nœuds est calculé sur la base d'une distance d'édition. $distLabels_{Struct}$ est ici la distance de Levenshtein entre les deux libellés divisée par la longueur du libellé le plus court et multipliée par 2 si les nœuds ont des types incompatibles. Ce choix a deux effets :

1. tenir compte des tailles des labels. En effet, une édition a beaucoup plus d'impact lorsqu'une séquence est courte que lorsqu'elle est longue. La distance d'édition entre *« Configuration normale »* et *« Configuratlon nrmale »* est de 2 (1 remplacement, 1 suppression), tout comme celle entre *« Main »* et *« ai »* (2 suppressions). Cependant, dans le premier cas on peut considérer que les deux chaînes se ressemblent, car le ratio $\frac{nombre\ de\ caractères\ différents}{nombre\ total\ de\ caractères}$ reste faible. Dans le second cas, ce ratio est de 50%, et est beaucoup trop grand pour qu'on puisse considérer ces deux éditions comme ayant le même impact.
2. tenir compte des types. Nous avons vu que les types des nœuds, aussi bien du modèle que de l'arbre cas, caractérisaient les niveaux de granularité auxquels les nœuds appartenaient. Deux nœuds n'appartenant pas au même niveau de granularité ont moins de chance de pouvoir se correspondre, même si leurs labels sont presque identiques, c'est pourquoi nous appliquons un multiplicateur à la distance entre deux nœuds qui ne sont pas du même niveau de granularité. Ainsi, si deux nœuds ont des labels proches mais que l'un correspond à un titre d'examen et l'autre à une observation, alors leur distance sera artificiellement augmentée.

La fonction Clojure présentée en annexe (section 7.1.2) montre l'implantation de $distLabels_{Struct}$.

4.6.1.2.2 Cas de *MET.Sim.StructSem* : Ici, les nœuds peuvent être comparés en utilisant une mesure de distance sémantique. Cependant, l'annotation du corpus ne pouvant pas être complète, ACC.A contient toujours du texte non annoté. $distLabels_{StructSem}$ est donc une mesure qui détaille plusieurs sous-cas :

— si les deux nœuds à comparer n'ont pas pu être annotés et sont restés purement textuels, nous utilisons $distLabels_{Struct}$ comme précédemment ;

— si l'un des deux nœuds contient des concepts mais pas l'autre, nous récupérons les labels de ces concepts et utilisons encore une fois $distLabels_{Struct}$;
— si les deux nœuds contiennent des concepts provenant de la même ontologie, nous utilisons la mesure de Z. Wu et Palmer (1994), classée selon Harispe et al. (2013) dans les approches structurelles (c'est à dire celles qui ne se basent que sur l'analyse de l'ontologie en tant que graphe pour établir une mesure).

La métrique de Z. Wu et Palmer (1994) donne toujours un résultat compris entre zéro et 1. Entre deux concepts u et v, elle se calcule ainsi :

$$sim_{WP}(u,v) = \frac{2\,prof(AC(u,v))}{2\,prof(AC(u,v)) + pcc(u, subClassOf, AC(u,v)) + pcc(v, subClassOf, AC(u,v))}$$

Les notations utilisées ici sont :

— $AC(u,v)$ qui désigne l'ancêtre commun de u et v le plus proche, autrement dit le moins général ;
— $prof(u)$ qui est le nombre minimal de sauts à réaliser en suivant des liens `subClassOf` dans l'ontologie pour atteindre u depuis la racine (`Thing` en OWL) de l'ontologie ;
— $pcc(u,r,v)$ est la taille du plus court chemin entre u et v (le nombre de sauts à réaliser) en ne suivant que des relations de type r.

Vu que l'on ne suit ici que les liens de subsomption (`subClassOf` en OWL), il s'agit donc bien d'une similarité qui est mesurée, et non d'une proximité. On peut également noter le parallèle qu'il existe entre cette métrique et un modèle de ratio de Tversky (1977) (montré en section 2.3.2 dans la formule 2.6) où $\alpha = \beta = 1$. $AC(u,v)$ agit en effet comme l'ensemble des caractéristiques communes à u et v, $pcc(u, subClassOf, AC(u,v))$ comme la quantité de caractéristiques ajoutées par u et $pcc(v, subClassOf, AC(u,v))$ comme la quantité de caractéristiques ajoutées par v.

4.6.1.3 Coûts structurels : coûts d'ascendance et de fratrie

Le coût d'ascendance $c_a([n_1,n_2])$ (*ancestry cost*) d'un arc $[n_1,n_2]$ a pour but de pénaliser cet arc au sein d'un mapping M dans lequel les enfants de n_1 ne sont pas mis en correspondance avec les enfants de n_2. Ainsi, le coût d'ascendance va augmenter si l'un des enfants de m (resp. n) « brise » l'ascendance, en étant mis en correspondance ailleurs que parmi les enfants de n (resp. m). Pour rappel, $M(n)$ dénote le nœud correspondant à n dans un mapping M. Ainsi, si $V_M(n)$ est l'ensemble des fils de n qui ne sont pas mis en correspondance avec un fils de $M(n)$, alors le coût d'ascendance d'un arc $[n_1,n_2]$ est égal à $|V_M(n_1)| + |V_M(n_2)|$.

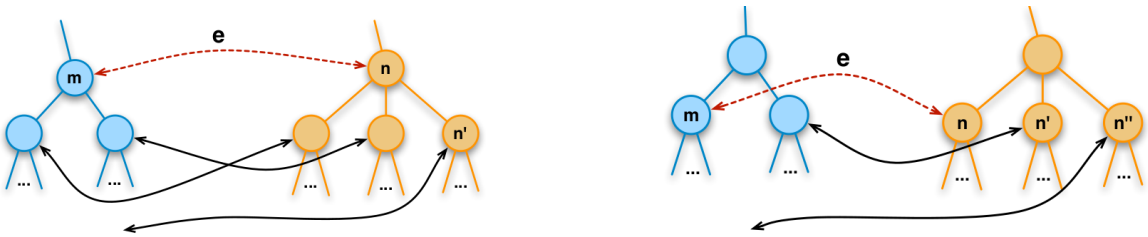
De la même manière, le coût de fratrie $c_a([n_1,n_2])$ (*sibling cost*) d'un arc $[n_1,n_2]$ pénalise cet arc si le mapping M dans lequel il se trouve ne fait pas correspondre les frères de n_1 à des frères de n_2. Son calcul est un peu plus compliqué. On note $S(n)$ l'ensemble comprenant les frères de n ainsi que n lui-même. On note $I_M(n)$ l'ensemble des nœuds *invariants* de $S(n)$, c'est à dire ceux qui sont mis en correspondance par M vers des nœuds de $S(M(n))$. On note $D_M(n)$ l'ensemble des nœuds *divergents* de $S(n)$, c'est à dire ceux qui n'appartiennent pas à $I_M(n)$ *et qui ne sont pas mis en correspondance avec le nœud vide* ψ. Pour finir, on définit $F_M(n)$, l'ensemble des *familles de frères distinctes*, autrement dit

l'ensemble de tous les parents des nœuds vers lesquelles les nœuds de $S(n)$ sont mis en correspondance par M. Ainsi si $P(n)$ désigne le parent d'un nœud n :

$$F_M(n) = \bigcup_{n' \in S(n)} P(M(n'))$$

La figure 4.4 montre une traduction de l'exemple donné par Kumar pour l'établissement de ces deux coûts structurels.

FIGURE 4.4: Coûts d'ascendance et de fratrie



À gauche on détermine la pénalité d'ascendance pour un arc $e = [m, n]$ en comptant les enfants de m et n qui brisent l'ascendance. Dans cet exemple n' brise l'ascendance car il n'est pas mis en correspondance avec un enfant de m ; de ce fait n' induit un coût d'ascendance sur e.

À droite on détermine le coût de fratrie pour un arc $e = [m, n]$, et l'algorithme calcule les sous-ensembles de frères invariants et divergents de m et n. Dans cet exemple, $I_M(n) = \{n'\}$ et $D_M(n) = \{n''\}$. Ainsi, n' diminue le coût de fratrie de e et n'' l'augmente.

4.6.1.4 Paramétrage et pondération

Nous avons montré comment calculer le coût induit dans un mapping par un arc reliant deux nœuds n'ayant potientellement pas le même label et étant situés à des endroits différents de l'abre. Le FTM nécessite quelques paramètres pour calculer le coût final d'un arc. Chaque coût structurel (coût d'ascendance et de fratrie) est multiplié par un poids : w_a pour le coût d'ascendance et w_s pour le coût de fratrie. Un troisième poids existe, appelé w_n et qui est le coût de la mise en correspondance d'un nœud vers ψ. KUMAR, TALTON et al. (2011) citent un quatrième poids, w_r qui est à multiplier à la distance entre les labels, que nous avons choisi d'ignorer car nous pouvons l'englober dans la fonction *distLabels*.

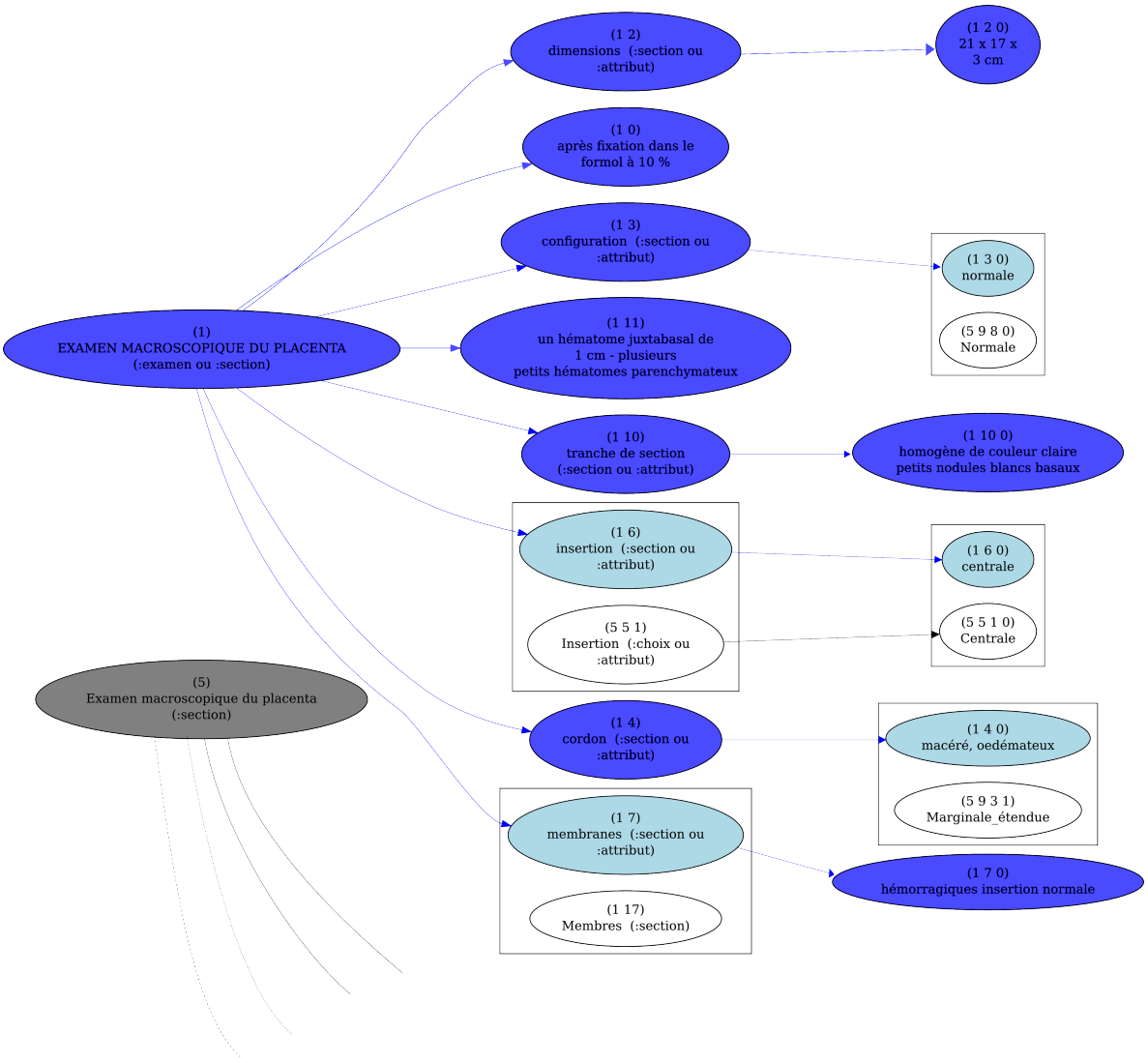
Le coût total d'un arc $[n_1, n_2]$ d'un mapping M est donc :

$$coût([n_1, n_2]) = \begin{cases} w_n \text{ si } n_1 = \psi \text{ ou } n_2 = \psi \\ distLabels(n_1, n_2) + w_a \times c_a([n_1, n_2]) + w_s \times c_s([n_1, n_2]) \text{ sinon} \end{cases}$$

Les paramètres que nous devons régler sont donc les poids w_a, w_s et w_n.

Pour finir, la figure 4.5 montre un exemple de la mise en correspondance d'éléments de données de la section concernant l'examen du placenta.

FIGURE 4.5: Exemple de mapping



Cette portion de mapping contient des nœuds correspondant à des observations portant sur le placenta. Les nœuds en bleu proviennent du compte rendu, ceux en blanc/gris du modèle. Les nœuds clairs contenus dans des boîtes sont ceux qui ont étés mis en correspondance. Les numéros (par ex. `(5 5 1)`) donnent le chemin d'un nœud dans l'arbre, et permettent donc de l'identifier.

4.6.2 *MET.Map.Inst*

Contrairement à *MET.Map.Flexible*, *MET.Map.Inst* est développée directement pour les besoins de la tâche d'instanciation du modèle. On appelera donc ici Γ_{cas} l'ensemble des libellés possibles chez un cas et $\Gamma_{modèle}$ ceux existant dans l'arbre modèle. *MET.Map.Inst* suit *MET.Seg.Simple* (présentée en 4.3.1), et suppose donc que l'on adoint à chaque nœud de $A(\Gamma_{cas})$ l'information du niveau de granularité auquel chaque nœud doit se trouver. La fonction de mapping est appelée ici $mapping_{inst}$, et $mapping_{inst}(distLabels)$ adoptera donc un biais et ne traitera pas de la même manière les nœuds du premier arbre (ceux du cas) et les nœuds de second (ceux du modèle). La fonction *distLabels* que nous utilisons sera la même que $distLabels_{Struct}$ présentée en section 4.6.1.2.1. Il est nécessaire également de trouver un paramètre *seuil* qu'on comparera à la valeur de retour de *distLabels* pour déterminer si les deux nœuds doivent être mis en correspondance ou non. *MET.Map.Inst* fonctionne de plus sur des arbres ordonnés, et ne tient donc pas compte de possibles interversions d'éléments d'informations entre un cas et le modèle.

Pour les besoins de *MET.Map.Inst*, on définit l'opérateur

$$parcoursP2 : Z(\Gamma_{cas}) \times Z(\Gamma_{modèle}) \longrightarrow (Z(\Gamma_{cas}) \times Z(\Gamma_{modèle})) \cup \{(\psi, \psi)\}$$

qui étant donné deux zippers pointant sur des locus des deux arbres, va soit progresser d'un nœud dans le parcours en profondeur dans ces deux arbres à la fois, soit retourner la paire (ψ, ψ) si le parcours de l'un des deux arbres est arrivé à son terme.

Soit $distLocs(z_1, z_2) = distLabels(label(noeud(z_1)), label(noeud(z_2)))$. L'algorithme de *MET.Map.Inst* est exposé dans l'algorithme 1.

La fonction *scinderLoc* va éclater le nœud cas N_0 et son libellé l_0 en deux, en prenant la taille n (en nombre de caractères) du libellé du nœud modèle comme base. Elle va donc produire deux libellés : l_1 contenant les n premiers caractères de l_0 et l_2 contenant le reste. Toutefois, l_1 et l_2 seront ajustés afin que l'on ne coupe pas l_0 en plein milieu d'un mot (ce qui est le cas si l_1 ne finit pas et que l_2 ne commence pas par un espace blanc). Si l_0 a été coupé au milieu d'un mot et que l_1 (resp. l_2) contient la portion de mot la plus courte, alors on déplace cette portion au début de l_2 (resp. à la fin de l_1). À la suite de cela, *scinderLoc* produit deux nouveaux nœuds, N_1 contenant l_1 et N_2 contenant l_2, tel que :

— N_2 soit le premier fils de N_1 ;
— N_1 soit placé au même niveau de granularité que N_0 ;
— N_2 soit considéré comme étant au niveau *détaillé*.

scinderLoc est basée sur l'idée qu'il n'est pas toujours possible pour la segmentation de distinguer uniquement sur la base de leur MFM les élements du niveau *intermédiaire* de ceux du niveau *fin* ou ceux du niveau *fin* de ceux du niveau *détaillé*. Par exemple si l'on revient à la figure 4.2, on peut noter que la deuxième ligne, `épithélium amniotique normal`, est reconnue comme un seul et même nœud, ce qui est normal puisqu'ici aucune MFM ne permettrait d'en faire autre chose. Or, le fragment de modèle présenté en 4.1.1.1 (dans la section `Examen_histologique_du_placenta`, sous-section `Placenta`) indique que `Épithélium_amniotique` est un élément de donnée attendu avec une

Algorithme 1 : Fonction $mapping_{inst}(arbreCas, arbreModèle, seuil)$

Résultat : *mappés* (ensemble paires de locus cas/modèle), *restants* (ensemble de locus du cas)

$mappés \leftarrow \emptyset$

$restants \leftarrow \emptyset$

$locCas \leftarrow zipper(arbreCas)$

$locModèleP \leftarrow zipper(arbreModèle)$

tant que $locCas \neq \psi \wedge locModèleP \neq \psi$ **faire**

- $locModèle \leftarrow locModèleP$
- $mappingTrouvé \leftarrow Faux$
- **tant que** $locModèle \neq \psi \wedge \neg mappingTrouvé$ **faire**
  - **si** $distLocs(locCas, locModèle) < seuil$ **alors**
    - $mappingTrouvé \leftarrow Vrai$
  - **sinon si** *noeud*$(locCas)$ *et noeud*$(locModèle)$ *appartiennent au niveau* intermédiaire *ou au niveau* fin **alors**
    - $locCasScindé \leftarrow scinderLoc(locCas, label(noeud(locModèle)))$
    - **si** $distLocs(locCasScindé, locModèle) < seuil$ **alors**
      - $locCas \leftarrow locCasScindé$
      - $mappingTrouvé \leftarrow Vrai$
  - **si** $mappingTrouvé$ **alors**
    - $mappés \leftarrow mappés \cup \{(locCas, locModèle)\}$
    - $locModèleP \leftarrow locModèle$
  - **sinon**
    - $locModèle \leftarrow droite(locModèle)$
- **si** $\neg mappingTrouvé$ **alors**
  - $restants \leftarrow restants \cup \{locCas\}$
- **si** $mappingTrouvé \vee droite(locCas) = \psi$ **alors**
  - $(locCas, locModèleP) \leftarrow parcoursP2(locCas, locModèleP)$
- **sinon**
  - $locCas \leftarrow droite(locCas)$

valeur associée de type `texte_libre`. On a donc typiquement ici un cas d'utilisation de *scinderLoc*.

4.6.3 *MET.Map.Hybride*

Le *MET.Map.Flexible* a pour avantage sa capacité a pouvoir potentiellement retrouver des éléments de structure présentés de manière très différentes entre deux arbres. Mais cette flexibilité a un coût important : le temps. En effet, dans le cas d'un arbre cas correspondant plutôt déjà bien au modèle, nous n'avons déjà pas de garantie qu'il converge vers le même résultat que *MET.Map.Inst*, et de plus nous n'avons pas de garantie quant au temps qu'il mettra à converger. *MET.Map.Hybride* est tout simplement une combinaison des deux méthodes précédentes afin de bénéficier des propriétés des deux là où ces propriétés sont utiles.

Il s'agit ici d'initialiser le premier mapping qui démarrera l'itération du flexible tree matching non pas aléatoirement mais grâce à *MET.Map.Inst*. Ceci devrait nous permettre de réduire le nombre d'itérations à faire réaliser par l'algorithme de Metropolis-Hastings derrière le FTM.

4.7 Protocole d'évaluation des différentes méthodes

Nous exposons ici les deux parties de l'évaluation des résultats : d'abord de manière purement statistique et ensuite via le concours des praticiens du projet Accordys.

4.7.1 Comparaison de deux métriques de similarité

Nous présentons dans cette section le test de Mantel (1967) et l'utilité qu'il revêt dans notre cas.

4.7.1.1 Test de Mantel

Pour savoir s'il existe une relation entre les résultats produits par les différentes méthodes utilisées, il est possible de calculer la corrélation entre les matrices de similarité produites. Introduit par Mantel (1967), ce test statistique utilise des permutations de lignes et colonnes pour calculer la corrélation entre matrices de dissimilarité ou de similarité. Il existe différentes méthodes de calcul telles que le coefficient de corrélation de Pearson qui permet d'analyser les relations linéaires ou le ρ de Spearman qui permet d'analyser les relations non-linéaires monotones. Une relation est linéaire si le nuage de points peut s'ajuster correctement à une droite. Une relation est monotone si elle est strictement croissante ou strictement décroissante. Ce type de test est utilisé en biologie pour comparer différentes manières de calculer les distances entre populations, par exemple leur éloignement géographique et leur divergence génétique (Diniz-Filho et al. (2013)).

Pour notre part, nous utiliserons le ρ de Spearman car il examine s'il existe une relation entre le *rang*

des observations pour deux variables. En utilisant le ρ de Spearman, une haute corrélation entre deux matrices démontrera que même si les similarités calculées sont différentes en termes de valeurs exactes, l'ordre est conservé. Par exemple, pour deux matrices de similarité M_1 et M_2, si M_1 montre que les documents les plus similaires à doc_1 sont (du plus au moins similaire) doc_2, doc_3, doc_4, etc. et que le test de Mantel entre M_1 et M_2 montre une corrélation proche de 1, cela signifie que les documents qui étaient les plus proches de doc_1 dans M_1 sont les mêmes dans M_2 (et toujours dans un ordre proche de doc_2, doc_3, doc_4, etc).

Le ρ de Spearman varie entre -1 et +1 :

— Si la corrélation est proche de 0, il n'y a pas de relation de rang entre les matrices ;
— Si la corrélation est proche de -1, il existe une forte relation négative entre les matrices ;
— Si la corrélation est proche de 1, il existe une forte relation positive entre les matrices.

4.7.1.2 Intérêt et utilisation du test de Mantel

Dans notre cas, l'obtention de matrices trop fortement corrélées démontrerait la redondance des méthodes utilisées. Le test de Mantel effectué sur chaque paire de matrices nous indiquera de manière immédiate si l'ajout des identifiants des ancêtres des concepts trouvés change les résultats finaux. Il nous montrera aussi si la conservation du texte qui n'a pu être annoté a un impact ou si les identifiants des concepts seuls suffisent à donner un résultat pareillement exploitable. Le calcul de la corrélation via le ρ de Spearman nous permet pour sa part d'être indépendant des scores de similarité entre comptes rendus en eux-mêmes, qui n'intéressent pas les praticiens.

L'étape faisant autorité en matière d'établissement de la pertinence des différentes méthodes de calcul de similarités est la validation par les fœtopathologistes du projet Accordys. Cependant, il n'est pas possible de demander aux spécialistes de scruter intégralement une matrice de similarité de 2000 par 2000 pour établir la pertinence de chacun des rapprochements entre cas effectués, et encore moins de le faire une fois pour chaque méthode sur laquelle notre étude porte. Ainsi, cette mesure de corrélation inter-méthodes via le test Mantel nous donnera un premier indicateur pour savoir quelles méthodes et quels rapprochements entre cas devraient être soumis à l'évaluation des spécialistes à la fin du projet Accordys.

Nous utiliserons le test de Mantel afin de comparer les différentes méthodes (via les matrices de similarité qu'elles produisent) selon deux cas de figure :

— En conservant intactes toutes les valeurs des matrices de similarité comparées par le test ;
— En aplanissant partiellement les valeurs de similarité : pour chaque ligne de chaque matrice, on ne conserve intactes que les valeurs des percentiles extrêmes (les 5% de similarités les plus fortes et les 5% de similarités les plus faibles) et centraux (les 10% autour de la médiane). On met les valeurs du 6^{è} au 44^{è} percentile à la valeur du 25^{è}, et les valeurs du 56^{è} au 94^{è} percentile à la valeur du 75^{è}. Un exemple est montré en figure 4.6. Ce sont les valeurs extrêmes qui intéressent les praticiens, et non le classement de l'intégralité des comptes rendus selon leur similarité avec

un compte rendu donné. Ainsi, une forte corrélation entre deux méthodes mais uniquement dans les percentiles extrêmes est pour nous pareillement significative. Toutefois, nous faisons le choix de conserver également les valeurs médianes pour ce test.

4.7.2 Intervention des fœtopathologistes

À l'issue de l'évaluation statistique, nous établirons la liste des méthodes (parmi *MET.Sim.StructSem*, *MET.Sim.Struct*, *MET.Sim.Txt* et *MET.Sim.Sem*) à faire valider par les spécialistes d'Accordys, et pour chaque méthode la sélection de paires de cas à leur présenter.

Pour chaque paire de méthodes hautement correlées dans les quartiles extrêmes (ie. avec une corrélation supérieure à 0.95) nous ne conserverons que la méthode la plus simple, puisque celle nécessitant plus de préparation aura été jugée comme n'apportant pas suffisamment. Pour chaque méthode conservée, nous demanderons aux spécialistes d'évaluer chacune des paires de cas des quartiles extrêmes ainsi que des quartiles autour de la médiane et de leur attribuer une note sur 5 :

— 0 (rapprochement « nul ») : aucun rapport entre les cas présentés ;
— 1 (rappr. « faible ») : un vague rapport existe, mais la mise en relation des cas n'est pas pertinente ;
— 2 (rappr. « moyen ») : il existe un rapport entre les cas ;
— 3 (rappr. « fort ») : le praticien s'attend à voir le système lui suggérer ce rapprochement de cas ;
— 4 (rappr. « très fort ») : le médecin estime que les deux cas présentent quasiment les mêmes caractéristiques.
— 5 (rappr. « total ») : le médecin estime que les deux cas présentent exactement le même développement, les mêmes caractéristiques, etc. Il est tout à fait possible qu'aucune paire n'obtienne une note de $\frac{5}{5}$.

Bien entendu, nous ne leur indiquerons pas de quels quartiles proviennent les paires de cas qui leur sont présentées. Nous observerons ensuite :

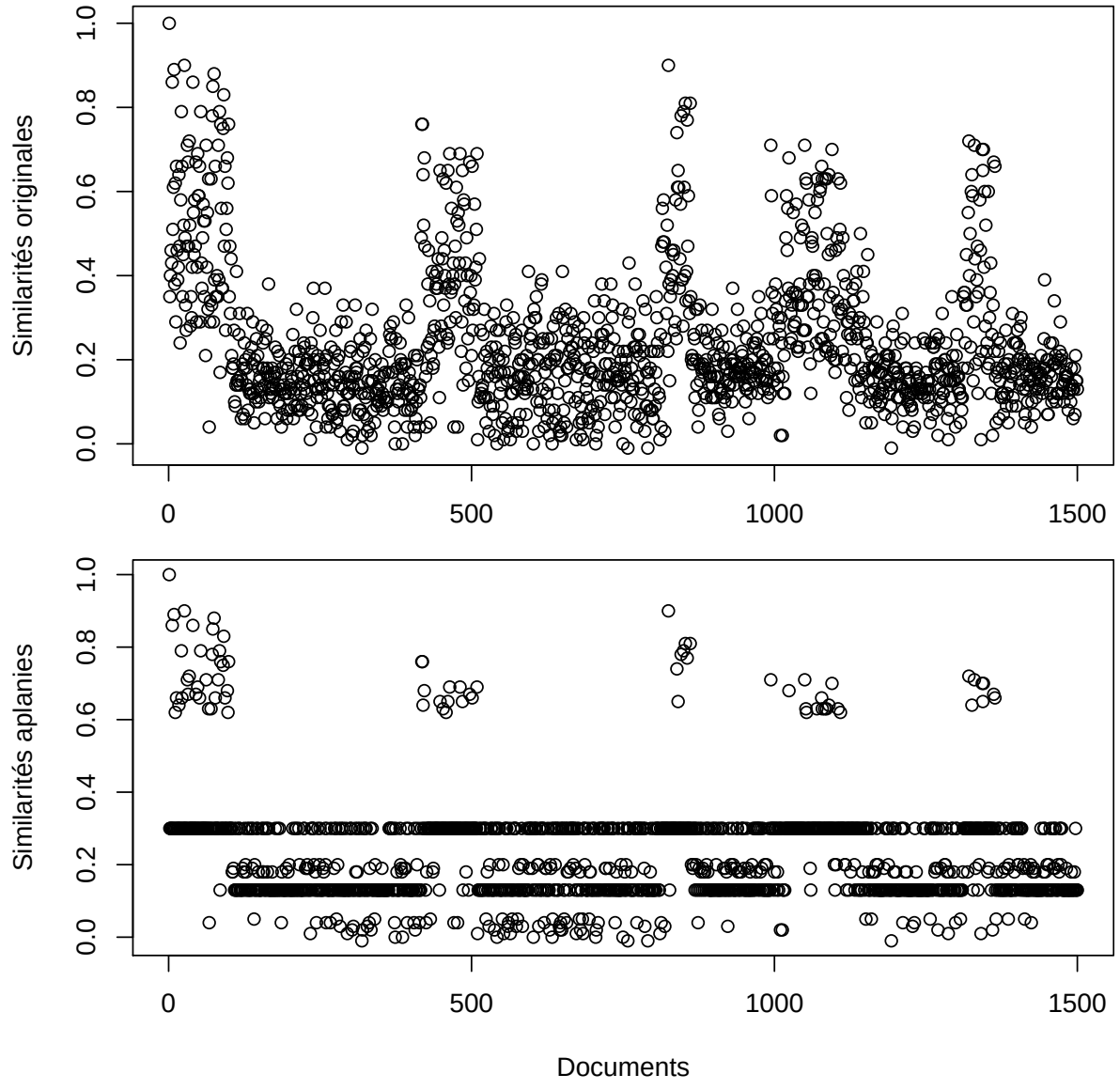
— si toutes les paires provenant du quartile le plus faible sont bien évaluées comme étant « nulles » ;
— si toutes celles des quartiles autour de la médiane ont bien une note inférieure ou égale à 2 ;
— si aucune des paires du quartile le plus fort n'est « nulle » ;
— si dans le quartile le plus fort il y a corrélation positive entre les rangs des similarités trouvées par le système et les notes des spécialistes.

4.8 Conclusion

Dans ce chapitre, nous avons présenté la méthode que nous suggérons pour :

— l'intégration d'annotations sémantiques à notre corpus ACC ;

FIGURE 4.6: Aplanissement partiel d'une ligne d'une matrice de similarité. Les points sont les scores de similarité avec tous les autres documents du corpus.



— la comparaison de comptes rendus représentés dans des modèles vectoriels ;
— la création d'un modèle de cas ;
— l'instantiation de ce modèle de cas par une méthode en deux phases (un mapping strict partiel servant de base à un mapping flexible fait de manière itérative), afin d'uniformiser la représentation des cas sans perdre d'information.

La suite de ce mémoire présentera les premiers résultats qui ont été obtenus lors de la mise en pratique de cette méthode sur ACC. Tous les points de la méthode n'ont en effet pas pu être testés, car ce travail est aujourd'hui (Septembre 2016) encore en cours dans le cadre du projet Accordys.

Chapitre 5

Réalisations et discussion

Du fait des problématiques légales recontrées par le projet Accordys et du délai d'obtention des divers lots du corpus ACC, les résultats présentés dans la suite de ce chapitre n'ont pas pu être obtenus sur ACC en intégralité. Seul ACC[1,3] seront utilisés. Toutefois, ACC[4] est la partie du corpus la plus ancienne et celle où la reconnaissance des caractères est la plus difficile, et donc celle de moins bonne qualité.

5.1 Filtrage des fichiers dupliqués

Le premier travail effectué a été d'analyser les détections de duplications potentielles de ACC[1,2] exposées en section 3.1.4. Ce travail n'a en effet pu être réalisé que sur les deux premières parties du corpus, le matériel ne comprenant pas de fichiers XML de détections pour ACC[3].

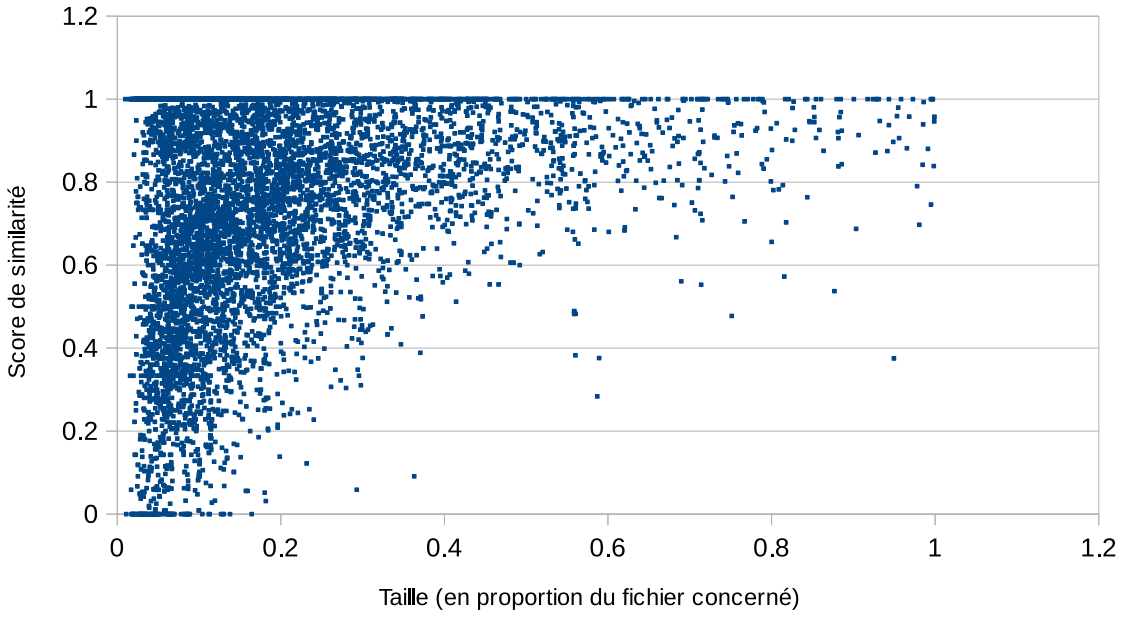
Pour chaque détection d d'une potentielle duplication, nous récupérons son score de similarité $score(d)$ (contenu dans le fichier XML exposé en section 3.1.5) et déterminons $taille(d)$, qui est calculée proportionnellement à la taille des fichiers concernés. Une détection d porte toujours sur deux fragments de texte $fragments(d)$ dans deux fichiers différents du même cas de fœtopathologie. La taille relative de la détection est définie par

$$taille(d) = \max_{f \in fragments(d)} \frac{taille\,du\,fragment\,f}{taille\,du\,fichier\,contenant\,f}$$

où les tailles de fragments et de fichiers sont exprimées en octets. Ainsi, une $taille(d)$ proche de zéro signifie que d touche une très faible proportion d'un des deux fichiers, et $taille(d) = 1$ signifie que d indique potentiellement une duplication de l'entièreté d'un fichier. Pour rappel, chaque fichier est une page scannée depuis ACC[1,3].

Chaque point sur la figure 5.1 représente une détection dans ACC[1,2]. On observe ici plusieurs choses :

FIGURE 5.1: Détections de potentielles duplications



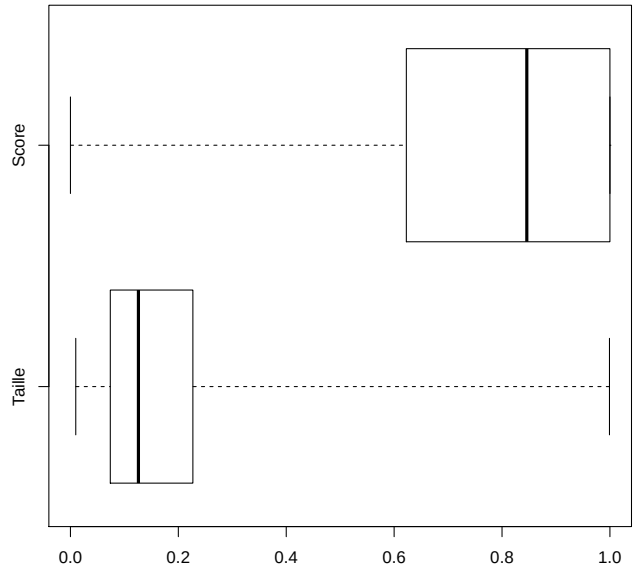
— malgré une absence de relation bien définie entre score et taille, il n'existe pas de détection de taille importante avec un score de similarité faible ;
— il n'y a pas vraiment de discontinuité dans les tailles de détections, toutes les tailles sont représentées ;
— la plupart des détections sont de taille relative moindre et de score de similarité important. Ceci est rendu plus clair lorsqu'on observe la répartition des scores et des tailles sous forme de boîtes à moustaches (cf. figure 5.2).

Seulement 25% des détections sont de taille relative supérieure à 0.22 (environ un quart de fichier). Ceci représente dans notre cas environ 2700 fichiers. 22% d'un fichier répliqué est pour nous une taille relative bien trop faible pour considérer ce fichier comme une réelle duplication, quand bien même le score de similarité serait maximal. Nous nous intéressons uniquement aux fichiers contenant des détections de plus de 50% en taille relative, ce qui ne représente que 5% des détections.

Une fois filtrés, les 5% de détections se répartissent comme indiqué sur la figure 5.3. Au final, nous observons deux choses :

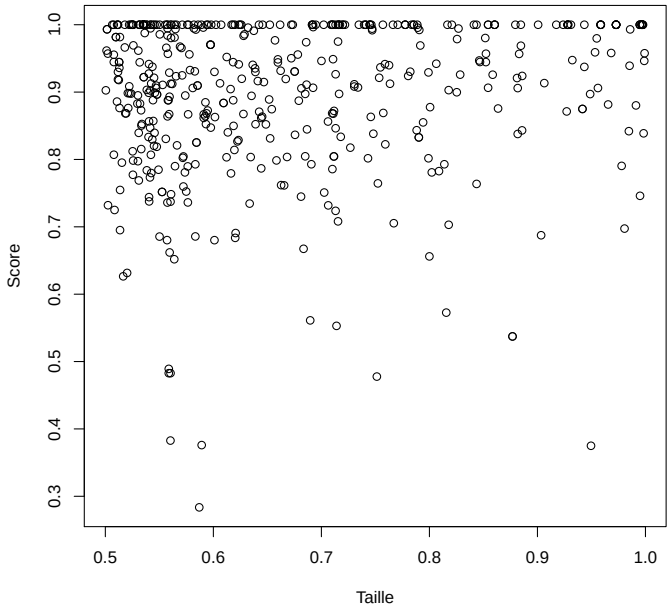
— les détections de taille importante concernent relativement peu de fichiers. 468 paires au total (tous scores confondus), et ce uniquement pour `ACC[1,2]`, puisque nous n'avons pas de fichier de détections pour `ACC[3]` ;
— il est assez difficile de fixer un seuil en termes de taille/score sur qui doit être considéré comme un duplicata.

FIGURE 5.2: Répartition des duplications



La première (resp. seconde) boîte représente les deuxième et troisième quartiles des scores (resp. tailles). Les lignes verticales épaisses correspondent aux médianes.

FIGURE 5.3: Duplications de taille relative supérieure à 0.5 détectées



Suite à ces observations, il a donc été décidé, pour le moment, que de ne pas retirer les fichiers contenant de potentielles duplications ne mettait pas en danger les résultats des étapes suivantes. De plus, les cas se différenciant souvent sur peu de mots, le risque de faux positif menant à une suppression des informations importantes d'un cas est présent. La suppression des duplications sera intégrée dans une future passe plus poussée de mise au propre du corpus qui sera réalisée par le projet Accordys, et accompagnera une étape de correction des erreurs d'OCR déjà évoquées. Dans notre cas, `ACC.F` sera donc composé des fichiers de `ACC` provenant d'un compte rendu d'examen fœtoplacentaire, mais groupés et concaténés en un seul fichier par compte rendu.

5.2 Annotation et analyse du corpus

L'annotation a été réalisée grâce à l'outil ECMT. Cet outil a permis d'annoter 41,4% de `ACC[1,3]` en termes de nombre de caractères recouverts par des annotations (4 001 145 caractères sur 9 663 333). La figure 5.4 montre le nombre de concepts distincts et le nombre total d'indexations trouvées, classées par ressource termino-ontologique (RTO) de provenance. Voici les RTO d'où proviennent les indexations trouvées, avec leurs abréviation et version utilisées par l'ECMT au moment de l'étude :

RTO	Abréviation	Version
National Cancer Institute Terminology (NCIt)	NCI	2012
SNOMED-CT	SNO	
Medical Subject Headings (MeSH), 2016	MSH	2016
Foundational Model of Anatomy	FMA	2009
International Classification of Diseases	ICD	v. 10
Human Phenotype Ontology	HPO	Mars 2014
Ontologie du Diagnostic Prénatal (OntoDPN)	DPN	v. 1
Medline	MED	v. 18.1
Metatermes CISMeF	CIS	2014
Classification Commune des Actes Medicaux (CCAM)	CCA	v. 41
Unified Medical Language System (UMLS)	UML	2012AB
Racines des médicaments	PHA	2009
Online Mendelian Inheritance in Man (OMIM)	MIM	2014

HRDO (Human Rare Diseases Ontology) n'est pas présente ici car aucun concept en provenant n'a été détecté par l'ECMT.

La figure 5.5 montre les mêmes informations que la 5.4 mais cette fois-ci réparties par catégories de concepts CISMeF, chaque catégorie contenant des termes de plusieurs RTO. Les noms des catégories font partie des métatermes CISMeF (CIS). Ceci permet de se rendre compte qu'une très grande partie des concepts et termes trouvés sont, comme on pouvait s'y attendre, des concepts anatomiques (catégorie « partie du corps, organe ou composant d'un organe »). Les concepts de l'ontologie du Diagnostic Prénatal et de la Human Phenotype Ontology n'ayant pas été catégorisés, ces deux ontologies ont leur propres catégories (appelées comme leurs abréviations DPN et HPO).

Ces concepts se retrouvent donc tous dans ACC.A. On remarque qu'OntoDPN, malgré son adéquation normalement plus forte au domaine de la fœtopathologie, n'est que peu représentée comparé aux autres ressources plus générales que sont la SNOMED, NCIt et le MeSH. Sur les 100 concepts les plus fréquemment retrouvés dans ACC.A, seulement 6 viennent d'OntoDPN :

DPN_CO_400 (« normal ») : présent 19193 fois
DPN_CO_463 (« examen ») : 5734 fois
DPN_CO_687 (« placenta ») : 4912 fois
DPN_CO_386 (« pied ») : 2279 fois
DPN_CO_468 (« état ») : 2116 fois
DPN_CO_486 (« absent ») : 2100 fois

Les trois premiers étant présents dans les 11 concepts les plus fréquemment retrouvés dans ACC.A.

FIGURE 5.4: Indexations et concepts distincts par ressource termino/ontologique dans ACC.A

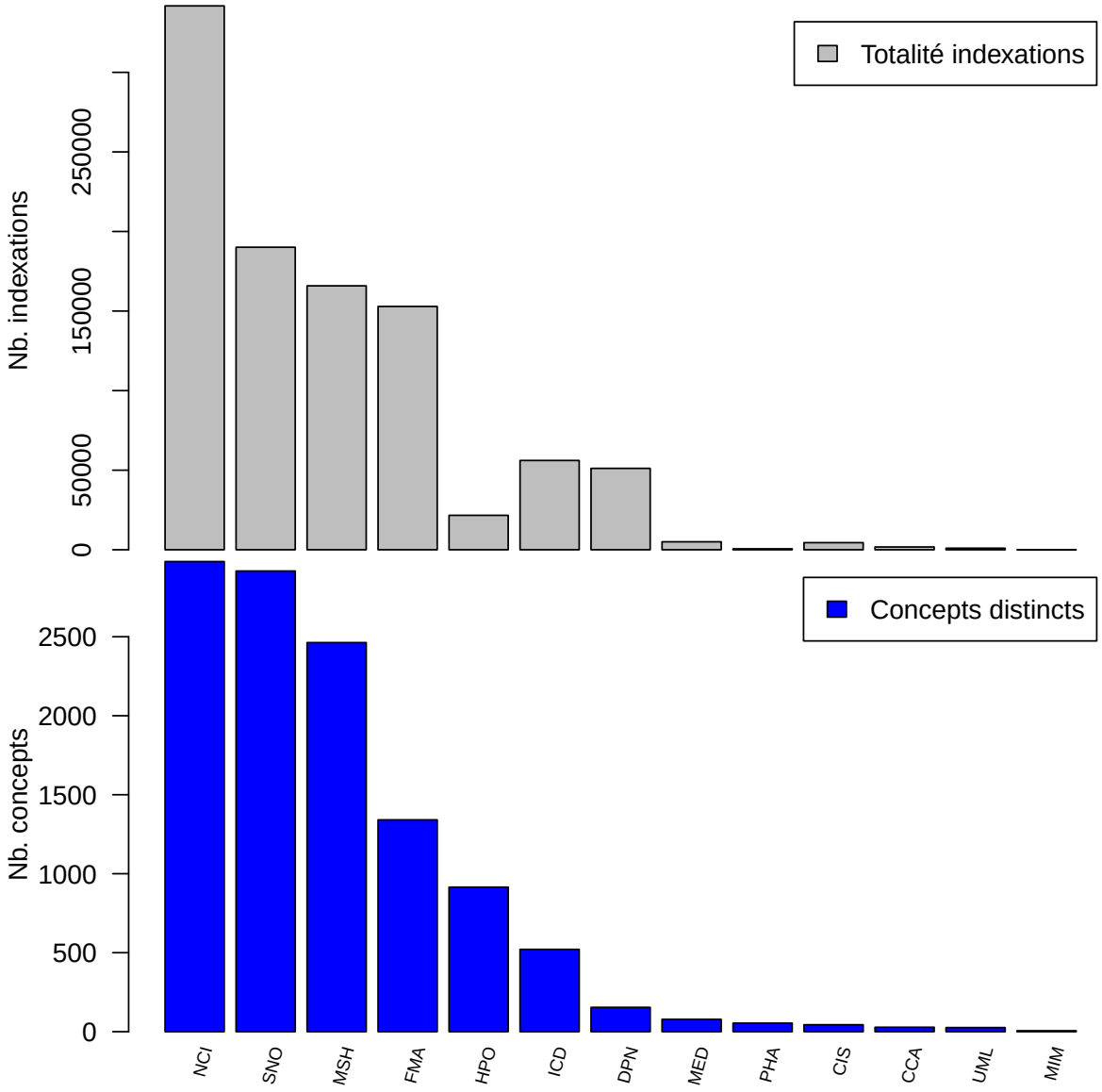
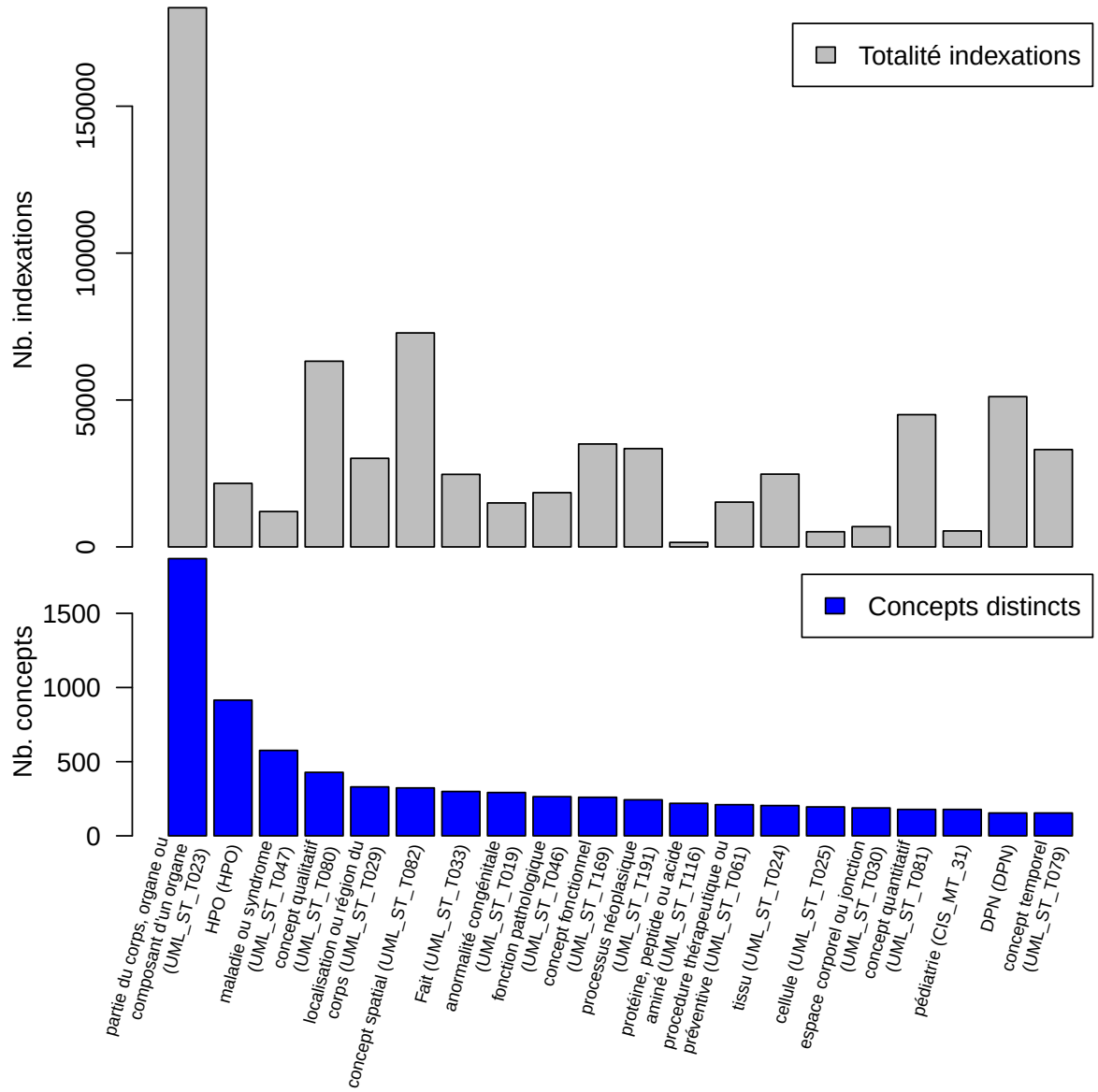


FIGURE 5.5: Indexations et concepts distincts par catégorie CISMeF dans ACC.A



5.2.1 Concepts les plus fréquemment retrouvés

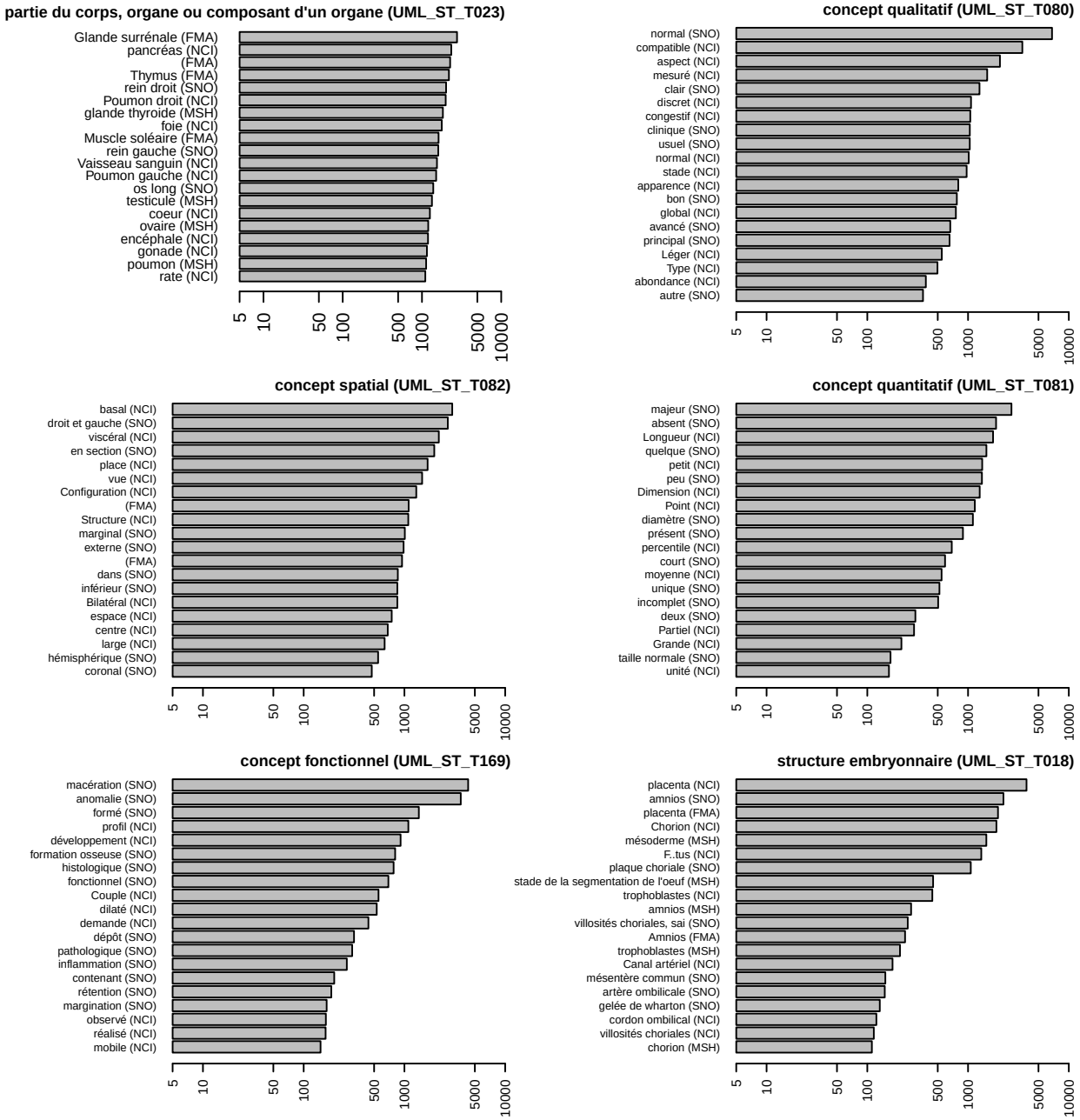
Pour analyser les concepts qui occurrent le plus souvent, nous devons d'abord effectuer un post-traitement sur ACC.A, afin de retirer les indexations qui se recouvrent intégralement (notamment lorsque deux concepts venant de deux ontologies différentes sont détectés au même endroit dans le texte). Si ceci n'est pas fait, on observe par exemple dans les annotations les plus communes trois concepts équivalents pour le pancréas dans trois ontologies différentes :

FMA_CO_7198 (« Pancréas ») : 2376 occurrences
NCI_CO_C12393 (« pancréas ») : 2343 occurrences
MSH_D_010179 (« pancréas ») : 2341 occurrences

Ces trois annotations co-occurrent, elles sont quasiment tout le temps retrouvées aux mêmes endroits dans ACC et elles ont toutes les raisons de l'être. Notons qu'elles ne sont pas toujours listées dans le même ordre dans le fichier XML renvoyé par l'ECMT (cf. section 4.4). Cependant, pour cette analyse des concepts médicaux distincts les plus fréquemment retrouvés dans les indexations, nous ne sommes pas intéressés par les concepts équivalents, car ils créent des répétitions. Ainsi dans l'exemple précédent, nous voulons que « Pancréas » apparaisse une et une seule fois, toujours sous le même concept de la même RTO.

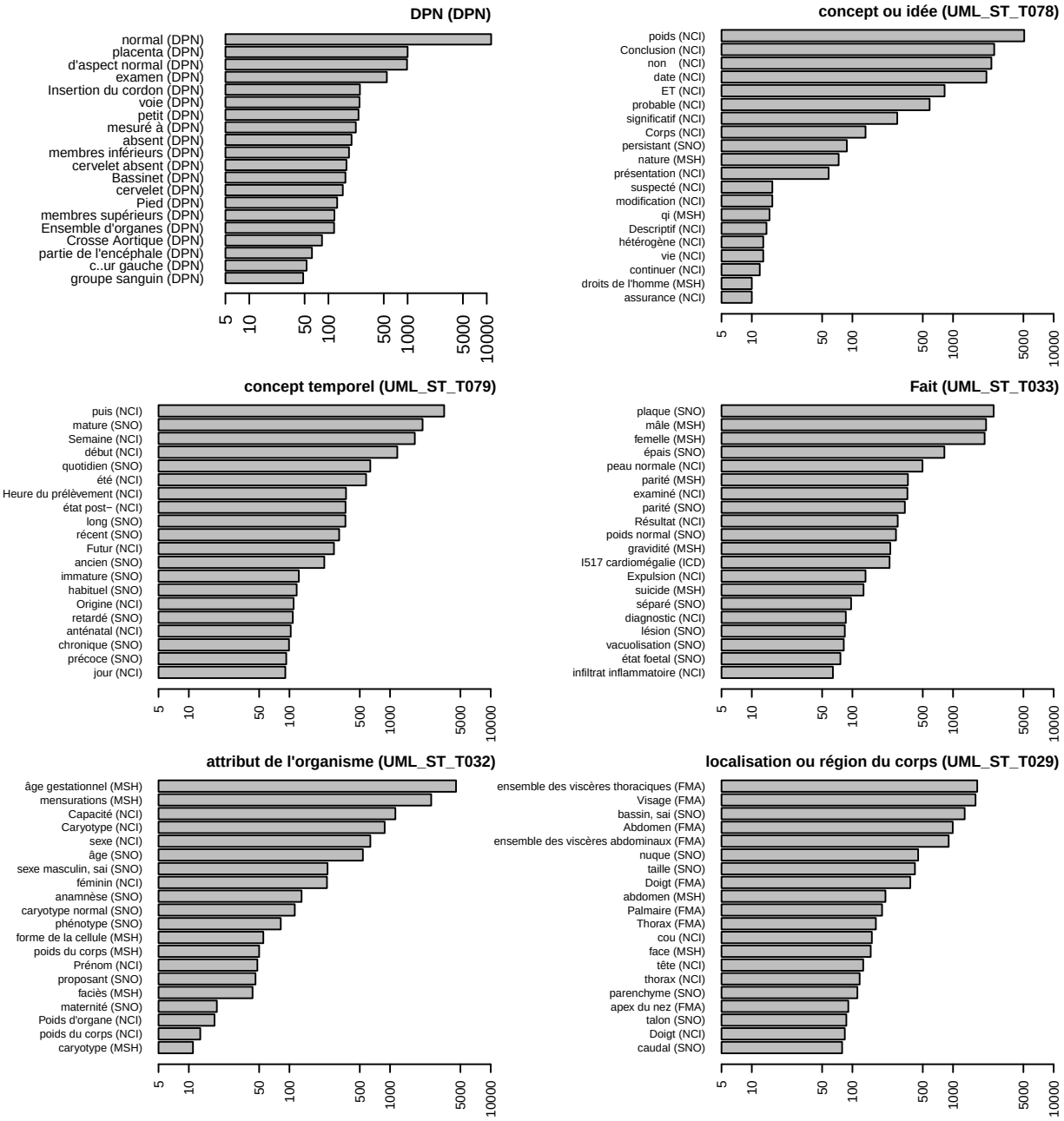
Il a donc été choisi lors de cette analyse de scinder les indexations en deux groupes : celles qui sont *recouvertes* et celles qui ne le sont pas. Pour clarifier, on appelle ACC.A.NR l'état du corpus ne contenant que les indexations ***non**-recouvertes* de ACC.A. L'heuristique pour déterminer si une annotation est *recouverte* est la suivante : pour chaque fichier, regrouper les indexations par position, ainsi toutes celles qui ont à la fois le même `start` et le même `end` seront dans le même groupe. Ensuite, dans chaque groupe prendre celle qui a le label le plus succinct. Si plusieurs labels ont la même taille, on prend comme *non-recouverte* l'indexation dont l'idcismef arrive le premier dans l'ordre alphabétique. Toutes les autres indexations à cette position sont considérées comme *recouvertes*. Le but n'est pas de retirer des indexations que l'on n'estime pas pertinentes, mais de pouvoir lister tous les concepts distincts et qui ne sont pas équivalents les uns aux autres. Nous voulons donc nous assurer que dans deux comptes rendus différents, l'indexation choisie comme *non-recouverte* soit toujours la même lorsqu'un même groupe de concepts (avec des labels dont les longueurs peuvent être identiques) se retrouve toujours ensemble mais que ces concepts ne sont pas à chaque fois mentionnés dans le même ordre dans le fichier XML (comme on l'a vu pour le pancréas).

La figure 5.6 montre pour chacune des 20 catégories les plus représentées les 20 concepts les plus fréquemment détectés, avec le nombre d'occurrences de ces concepts dans ACC.A.NR affiché dans une échelle logarithmique, car le nombre d'occurrences dans chaque catégorie décroit très vite. Ces résultats permettent de constater quelque chose d'important : malgré les erreurs d'OCR encore présentes dans ACC et le manque d'une ontologie spécifique au domaine, les indexations et catégories d'indexations que l'on retrouve le plus souvent sont pertinentes, et en rapport avec le domaine des comptes rendus. S'il y a bien évidemment du bruit dans les indexations trouvées, il n'apparait pas dans les indexations qui reviennent le plus souvent.

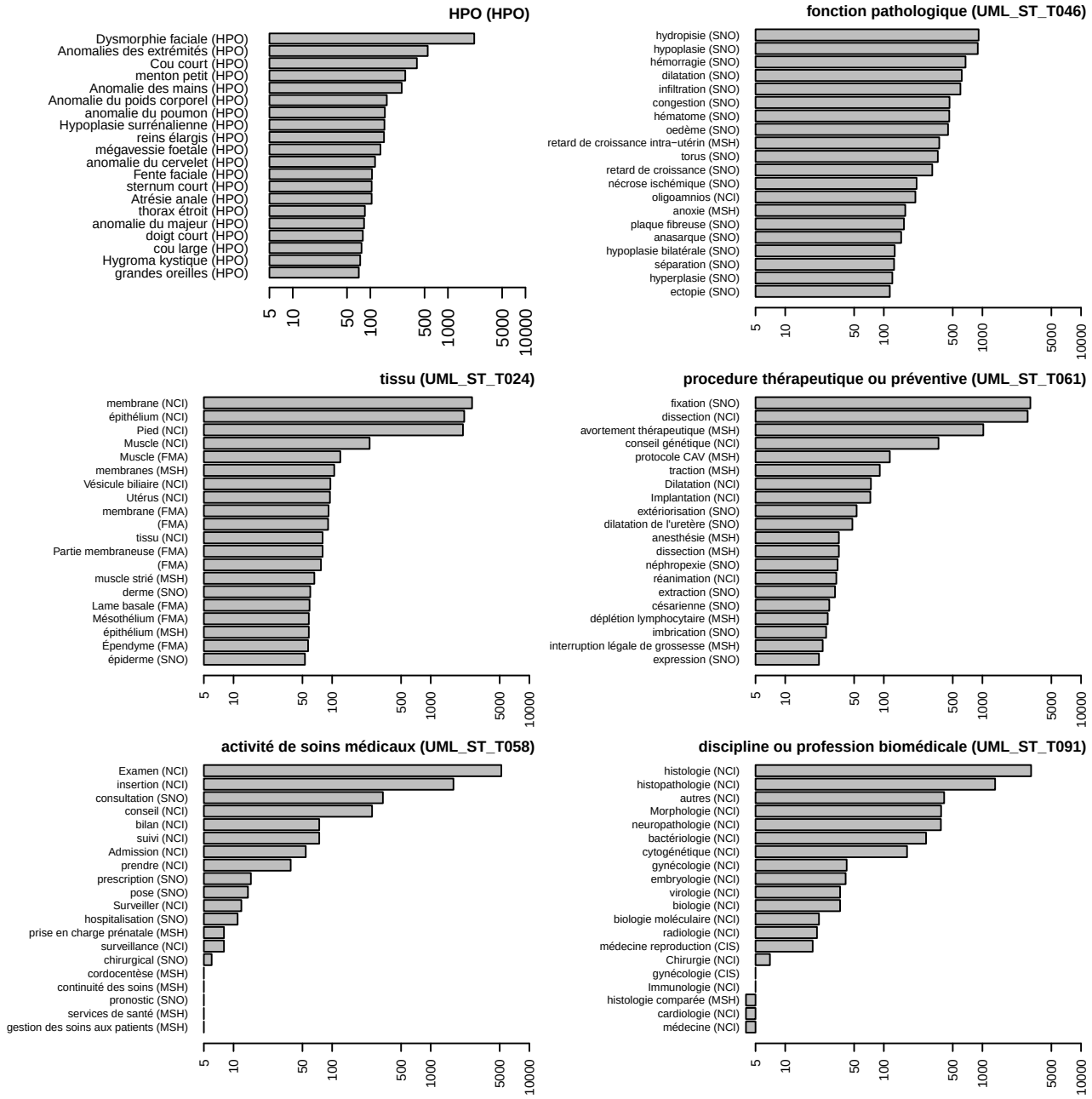


(a) (1/4)

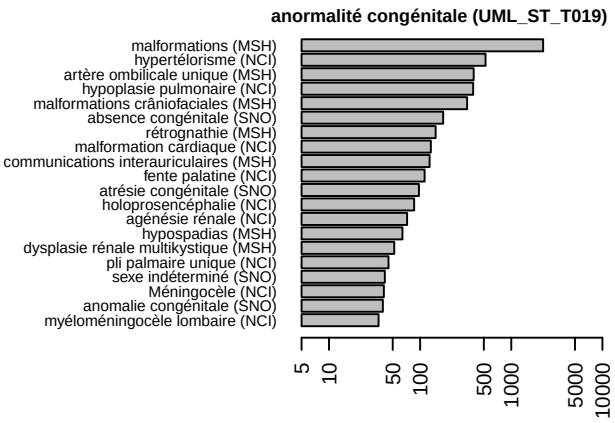
FIGURE 5.6: Indexations les plus fréquentes dans chaque catégorie CISMeF dans ACC.A.NR et nombre d'occurrences de chaque concept

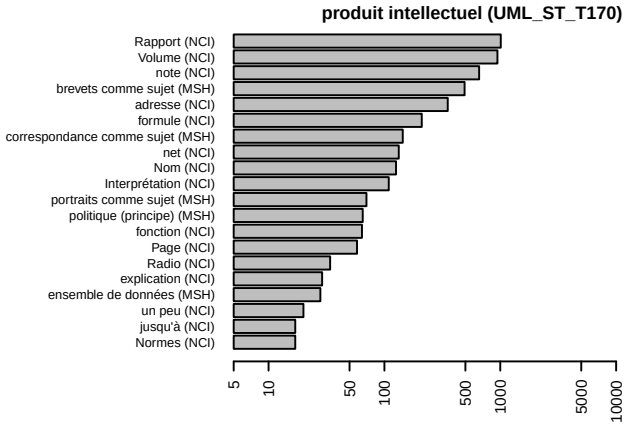

DPN (DPN)
normal (DPN)
placenta (DPN)
d'aspect normal (DPN)
examen (DPN)
Insertion du cordon (DPN)
voie (DPN)
petit (DPN)
mesuré à (DPN)
absent (DPN)
membres inférieurs (DPN)
cervelet absent (DPN)
Bassinet (DPN)
cervelet (DPN)
Pied (DPN)
membres supérieurs (DPN)
Ensemble d'organes (DPN)
Crosse Aortique (DPN)
partie de l'encéphale (DPN)
c..ur gauche (DPN)
groupe sanguin (DPN)
concept ou idée (UML_ST_T078)
poids (NCI)
Conclusion (NCI)
non (NCI)
date (NCI)
ET (NCI)
probable (NCI)
significatif (NCI)
Corps (NCI)
persistant (SNO)
nature (MSH)
présentation (NCI)
suspecté (NCI)
modification (NCI)
qi (MSH)
Descriptif (NCI)
hétérogène (NCI)
vie (NCI)
continuer (NCI)
droits de l'homme (MSH)
assurance (NCI)
concept temporel (UML_ST_T079)
puis (NCI)
mature (SNO)
Semaine (NCI)
début (NCI)
quotidien (SNO)
été (NCI)
Heure du prélèvement (NCI)
état post− (NCI)
long (SNO)
récent (SNO)
Futur (NCI)
ancien (SNO)
immature (SNO)
habituel (SNO)
Origine (NCI)
retardé (SNO)
anténatal (NCI)
chronique (SNO)
précoce (SNO)
jour (NCI)
Fait (UML_ST_T033)
plaque (SNO)
mâle (MSH)
femelle (MSH)
épais (SNO)
peau normale (NCI)
parité (MSH)
examiné (NCI)
parité (SNO)
Résultat (NCI)
poids normal (SNO)
gravidité (MSH)
I517 cardiomégalie (ICD)
Expulsion (NCI)
suicide (MSH)
séparé (SNO)
diagnostic (NCI)
lésion (SNO)
vacuolisation (SNO)
état foetal (SNO)
infiltrat inflammatoire (NCI)
attribut de l'organisme (UML_ST_T032)
âge gestationnel (MSH)
mensurations (MSH)
Capacité (NCI)
Caryotype (NCI)
sexe (NCI)
âge (SNO)
sexe masculin, sai (SNO)
féminin (NCI)
anamnèse (SNO)
caryotype normal (SNO)
phénotype (SNO)
forme de la cellule (MSH)
poids du corps (MSH)
Prénom (NCI)
proposant (SNO)
faciès (MSH)
maternité (SNO)
Poids d'organe (NCI)
poids du corps (NCI)
caryotype (MSH)
localisation ou région du corps (UML_ST_T029)
ensemble des viscères thoraciques (FMA)
Visage (FMA)
bassin, sai (SNO)
Abdomen (FMA)
ensemble des viscères abdominaux (FMA)
nuque (SNO)
taille (SNO)
Doigt (FMA)
abdomen (MSH)
Palmaire (FMA)
Thorax (FMA)
cou (NCI)
face (MSH)
tête (NCI)
thorax (NCI)
parenchyme (SNO)
apex du nez (FMA)
talon (SNO)
Doigt (NCI)
caudal (SNO)
5
10
50
100
500
1000
5000
10000

(b) (2/4)


HPO (HPO)
Dysmorphie faciale (HPO)
Anomalies des extrémités (HPO)
Cou court (HPO)
menton petit (HPO)
Anomalie des mains (HPO)
Anomalie du poids corporel (HPO)
anomalie du poumon (HPO)
Hypoplasie surrénalienne (HPO)
reins élargis (HPO)
mégavessie foetale (HPO)
anomalie du cervelet (HPO)
Fente faciale (HPO)
sternum court (HPO)
Atrésie anale (HPO)
thorax étroit (HPO)
anomalie du majeur (HPO)
doigt court (HPO)
cou large (HPO)
Hygroma kystique (HPO)
grandes oreilles (HPO)
fonction pathologique (UML_ST_T046)
hydropisie (SNO)
hypoplasie (SNO)
hémorragie (SNO)
dilatation (SNO)
infiltration (SNO)
congestion (SNO)
hématome (SNO)
oedème (SNO)
retard de croissance intra−utérin (MSH)
torus (SNO)
retard de croissance (SNO)
nécrose ischémique (SNO)
oligoamnios (NCI)
anoxie (MSH)
plaque fibreuse (SNO)
anasarque (SNO)
hypoplasie bilatérale (SNO)
séparation (SNO)
hyperplasie (SNO)
ectopie (SNO)
tissu (UML_ST_T024)
membrane (NCI)
épithélium (NCI)
Pied (NCI)
Muscle (NCI)
Muscle (FMA)
membranes (MSH)
Vésicule biliaire (NCI)
Utérus (NCI)
membrane (FMA)
(FMA)
tissu (NCI)
Partie membraneuse (FMA)
(FMA)
muscle strié (MSH)
derme (SNO)
Lame basale (FMA)
Mésothélium (FMA)
épithélium (MSH)
Épendyme (FMA)
épiderme (SNO)
procedure thérapeutique ou préventive (UML_ST_T061)
fixation (SNO)
dissection (NCI)
avortement thérapeutique (MSH)
conseil génétique (NCI)
protocole CAV (MSH)
traction (MSH)
Dilatation (NCI)
Implantation (NCI)
extériorisation (SNO)
dilatation de l'uretère (SNO)
anesthésie (MSH)
dissection (MSH)
néphropexie (SNO)
réanimation (NCI)
extraction (SNO)
césarienne (SNO)
déplétion lymphocytaire (MSH)
imbrication (SNO)
interruption légale de grossesse (MSH)
expression (SNO)
activité de soins médicaux (UML_ST_T058)
Examen (NCI)
insertion (NCI)
consultation (SNO)
conseil (NCI)
bilan (NCI)
suivi (NCI)
Admission (NCI)
prendre (NCI)
prescription (SNO)
pose (SNO)
Surveiller (NCI)
hospitalisation (SNO)
prise en charge prénatale (MSH)
surveillance (NCI)
chirurgical (SNO)
cordocentèse (MSH)
continuité des soins (MSH)
pronostic (SNO)
services de santé (MSH)
gestion des soins aux patients (MSH)
discipline ou profession biomédicale (UML_ST_T091)
histologie (NCI)
histopathologie (NCI)
autres (NCI)
Morphologie (NCI)
neuropathologie (NCI)
bactériologie (NCI)
cytogénétique (NCI)
gynécologie (NCI)
embryologie (NCI)
virologie (NCI)
biologie (NCI)
biologie moléculaire (NCI)
radiologie (NCI)
médecine reproduction (CIS)
Chirurgie (NCI)
gynécologie (CIS)
Immunologie (NCI)
histologie comparée (MSH)
cardiologie (NCI)
médecine (NCI)
5
10
50
100
500
1000
5000
10000

(c) (3/4)


anormalité congénitale (UML_ST_T019)
malformations (MSH)
hypertélorisme (NCI)
artère ombilicale unique (MSH)
hypoplasie pulmonaire (NCI)
malformations crâniofaciales (MSH)
absence congénitale (SNO)
rétrognathie (MSH)
malformation cardiaque (NCI)
communications interauriculaires (MSH)
fente palatine (NCI)
atrésie congénitale (SNO)
holoprosencéphalie (NCI)
agénésie rénale (NCI)
hypospadias (MSH)
dysplasie rénale multikystique (MSH)
pli palmaire unique (NCI)
sexe indéterminé (SNO)
Méningocèle (NCI)
anomalie congénitale (SNO)
myéloméningocèle lombaire (NCI)
5
10
50
100
500
1000
5000
10000


produit intellectuel (UML_ST_T170)
Rapport (NCI)
Volume (NCI)
note (NCI)
brevets comme sujet (MSH)
adresse (NCI)
formule (NCI)
correspondance comme sujet (MSH)
net (NCI)
Nom (NCI)
Interprétation (NCI)
portraits comme sujet (MSH)
politique (principe) (MSH)
fonction (NCI)
Page (NCI)
Radio (NCI)
explication (NCI)
ensemble de données (MSH)
un peu (NCI)
jusqu'à (NCI)
Normes (NCI)
5
10
50
100
500
1000
5000
10000

(d) (4/4)

5.2.2 Autres remarques concernant le résultat de l'annotation

Nous avons observé dans certains fichiers des annotations qui moralement sont incorrectes, ceci étant du au fait que l'annotateur ignore les sauts de ligne et que ne nous pouvons pas systématiquement remplacer les sauts de ligne par des points avant d'envoyer le texte à l'annotateur. Par exemple dans l'extrait suivant

```
rate: hémorragique
thyroïde : développement normal - un petit ilôt de thymus accessoire
pancréas : développement normal
surrénales droite et gauche : congestives
```

l'annotateur a trouvé le concept « Pancréas accessoire (MSH_SC_536003) » à cheval sur la deuxième et troisième ligne. Il s'agit bien entendu ici d'un faux positif, et cette annotation devrait être ignorée par notre système au moment de l'intégration des indexations à ACC.A. Cependant, ceci requièrerait de faire d'abord une passe d'analyse structurelle afin de savoir quelles fins de ligne sont à considérer comme significatives (car elles séparent des éléments de données) et quelles autres sont à considérer comme du bruit. La majorité des sauts de ligne étant significatifs dans notre corpus (du fait de l'importance de sa mise en forme matérielle), nous avons fait le choix de simplement ignorer pour la constitution de ACC.A toutes les annotations se trouvant à cheval sur deux lignes ou plus. Cependant, ceci ne veut pas dire que toutes les annotations sur deux lignes seraient non pertinentes pour notre corpus, ainsi par souci d'exhaustivité ces annotations ont bien été incluses dans les statistiques présentées précédemment sur ACC.A.NR.

5.3 *MET.Sim.Txt* et *MET.Sim.Sem* : comparaison par modèle vectoriel

Nous exposons ici les résultats de *MET.Sim.Sem*. Les tokens de ACC.F et des différentes versions de ACC.A utilisées pour cette analyse ont été séparés par expression rationnelle avec NLTK (version 3.2.1) en Python. Chaque séquence de caractères dans chaque document correspondant à l'expression rationnelle `"\w[\w\-:]*"` (appliquée de manière gloutonne) génère un nouveau token. Autrement dit nous considérons comme un token chaque assemblage de lettres et de chiffres contenant potentiellement des tirets ou un deux-points (certains idcismefs contiennent des deux-points). La table 5.1 montre le nombre de tokens différents contenus dans ces états annotés du corpus.

Suite à ceci, la librairie Python gensim (version 0.12.4) a été utilisée pour la transformation des documents dans un modèle vectoriel avec pondération par TF-IDF et la réduction de dimension par LSI. Nous avons tout d'abord essayé une LSI avec 50 topics, car avec ses 1500 documents notre corpus reste tout de même de petite taille [1]. Nous avons donc à ce stade nos cinq matrices de similarité S_{IA} ,

1. http://stackoverflow.com/questions/9582291/how-do-we-decide-the-number-of-dimensions-for-latent-semantic-analysis

État du corpus	Nombre de tokens uniques
ACC.A.IA	16 237
ACC.A.I	10 777
ACC.A.TIA	67 331
ACC.A.TI	61 876
ACC.F	53 522

TABLE 5.1: Nombre de tokens uniques dans chaque état du corpus annoté sémantiquement

S_I , S_{TIA} , S_{TI} et S_T, chacune d'une taille de 1500 par 1500. Chacune de ces matrices contient des scores de similarité allant de -0,26 à 1 (le score minimum va de -0,26 à -0,18 selon la matrice).

La table 5.2 montre les corrélations entre les rangs des lignes des matrices de similarité obtenues (comme explicité en section 4.7.1.2).

	S_{IA}	S_I	S_{TIA}	S_{TI}	S_T
S_{IA}	-	0.825	0.912	0.703	0.597
S_I	-	-	0.789	0.808	0.651
S_{TIA}	-	-	-	0.839	0.746
S_{TI}	-	-	-	-	0.861
S_T	-	-	-	-	-

TABLE 5.2: Résultats des tests de Mantel effectués sur les matrices de similarité

On peut déjà noter qu'il y a à chaque fois une forte corrélation positive, toutes les valeurs étant dans l'intervalle $[0,5; 1]$. Le ρ de Spearman le plus faible est de presque 0,6 et il a lieu comme on pouvait s'y attendre entre les matrices S_T et S_{IA}, issues respectivement de ACC.F et ACC.A.IA qui sont des états du corpus n'ayant aucun token en commun. À ce stade là nous pouvons déjà attester que l'annotation a eu un impact significatif, quand bien même aucune ontologie spécifique au domaine de la fœtopathologie n'existe. Cependant, nous devrons attendre l'évaluation des spécialistes pour affirmer que cet impact dans les rapprochements de cas a été positif.

La corrélation la plus forte est entre S_{TIA} et S_{IA}. En effet, inclure les identifiants des ancêtres des concepts détectés change grandement le corpus, au point que le texte non annoté devient minoritaire face aux annotations ajoutées. Il était donc à attendre que ACC.A.TIA et ACC.A.IA soient proches, mais ceci nous permet quand même de constater que la conservation du texte qui n'a pas pu être annoté (qui représente originalement 59% de notre corpus) a un impact, car la corrélation bien que forte est de 0,91. Ceci est appuyé par la baisse importante de corrélation entre S_T/S_{TI} et S_T/S_I (de 0,861 à 0,651). Dans ce cas on note que la conservation du texte non annoté change assez fortement l'ordre des similarités.

La dernière observation concerne l'impact de l'inclusion des identifiants des ancêtres des concepts détectés et plus de ceux des concepts détectés eux-mêmes. Cela a bien eu un impact, étant donné la

corrélation de seulement 0,825 entre S_I et S_{IA}. Elle est d'ailleurs proche de celle entre S_{TI} et S_{TIA} (0,839). Confirmer que l'ordre des similarités est plus pertinent lorsqu'on inclut les ancêtres nécessite encore une fois l'évaluation des praticiens, mais il semblerait qu'on ait tout intérêt à les inclure.

Si nous observons maintenant la matrice des corrélations entre les matrices de distances partiellement aplanies (où, pour chaque ligne, toutes les similarités n'étant ni dans les 5 premiers percentiles, ni entre les percentiles 45 et 55, ni dans les 5 derniers percentiles ont été unformisées), nous obtenons la table 5.3.

	S_{IA}	S_I	S_{TIA}	S_{TI}	S_T
S_{IA}	-	0.765	0.866	0.644	0.540
S_I	-	-	0.728	0.748	0.588
S_{TIA}	-	-	-	0.779	0.683
S_{TI}	-	-	-	-	0.799
S_T	-	-	-	-	-

TABLE 5.3: Résultats des tests de Mantel effectués sur les matrices de similarité aplanies

Celle-ci montre une diminution uniforme de toutes les corrélations par rapport à précédemment. Ainsi, l'impact des différentes annotations incluses est augmenté si l'on ne considère que les percentiles extrêmes et centraux. Autrement dit, le fait de se focaliser uniquement sur ces percentiles contribue à différencier encore plus les méthodes d'annotation. Ceci se poursuit si l'on n'observe maintenant le résultat de ces corrélations sur des matrices où l'on n'a pour chaque ligne gardé que les 5% de similarité les plus hautes, à savoir celles qui intéressent le plus les praticiens, et mis le reste à zéro, comme montré en table 5.4.

	S_{IA}	S_I	S_{TIA}	S_{TI}	S_T
S_{IA}	-	0.612	0.753	0.468	0.384
S_I	-	-	0.586	0.597	0.446
S_{TIA}	-	-	-	0.628	0.521
S_{TI}	-	-	-	-	0.673
S_T	-	-	-	-	-

TABLE 5.4: Résultats des tests de Mantel effectués sur les matrices de similarité ne contenant que les 5% les plus élevés

5.4 Comparaison des méthodes d'homogénéisation de cas

Cette partie est celle sur laquelle la majorité des développements logiciels effectués se sont concentrés. Nous présentons d'abord les résultats obtenus avec notre implantation de l'algorithme de flexible tree matching utilisée seule, puis la contribution de l'implantation de la méthode d'instantiation du modèle

MET.Map.Inst présentée en section 4.6.2, puis la combinaison des deux en une méthode de mapping hybride.

5.4.1 Résultats de *MET.Seg.Simple*

Du fait de l'heuristique simple et adaptée à notre corpus adoptée par *MET.Seg.Simple*[2], les comptes rendus transformés en arbres sont utilisables dans la suite du processus. *MET.Seg.Simple* n'a pas pour but de produire un arbre parfait (le mapping fait à la suite devant affiner ça), cependant, nous pouvons noter deux éléments qui sont faux et potentiellement corrigeables dès cette étape :

— Les marqueurs typographiques pouvant varier d'un document à un autre, nous ne pouvons pas pré-indiquer au système lesquels sont pertinents pour détecter une énumération à tel ou tel niveau de granularité. Il en résulte des cas comme celui en figure 5.4 où des nœuds sont détectés comme frères alors qu'une hiérarchie devrait être mise en place. Ceci pourrait être cependant amélioré avant le mapping en tentant de détecter un *changement* dans les marqueurs typographiques utilisés d'une ligne à l'autre. L'utilisation des niveaux d'indentation (alinéas) reste problématique car ceux-ci ne sont pas constants et peuvent disparaître lors de l'OCR ;
— La détection des titres de section (au *niveau de granularité grossier*) est certainement le point qui est le plus *ad-hoc*, puisqu'il considère comme un titre potentiel toute ligne contenant en majorité des caractères alphabétiques, tous ces caractères étant en majuscule. Le problème est que dans certains comptes rendus, l'intégralité de la conclusion est également en majuscules, ce qui occasionne des faux positifs. Ignorer les lignes contiguës ressemblant à des titres (à l'exception de la première) permettrait d'éviter cela.

Cependant, on peut se rendre compte qu'il s'agit dans les deux cas d'ajouts de règles *ad-hoc* basées sur le contexte de chaque ligne (les lignes qui les précèdent en l'occurrence) et dans notre méthode l'avantage principal attendu de *MET.Seg.Apprentissage* sur *MET.Seg.Simple* est justement cette prise en compte du contexte. C'est pourquoi nous pensons que la poursuite de *MET.Seg.Apprentissage* est une meilleure voie que l'introduction de toujours plus de règles *ad-hoc* dans *MET.Seg.Simple*.

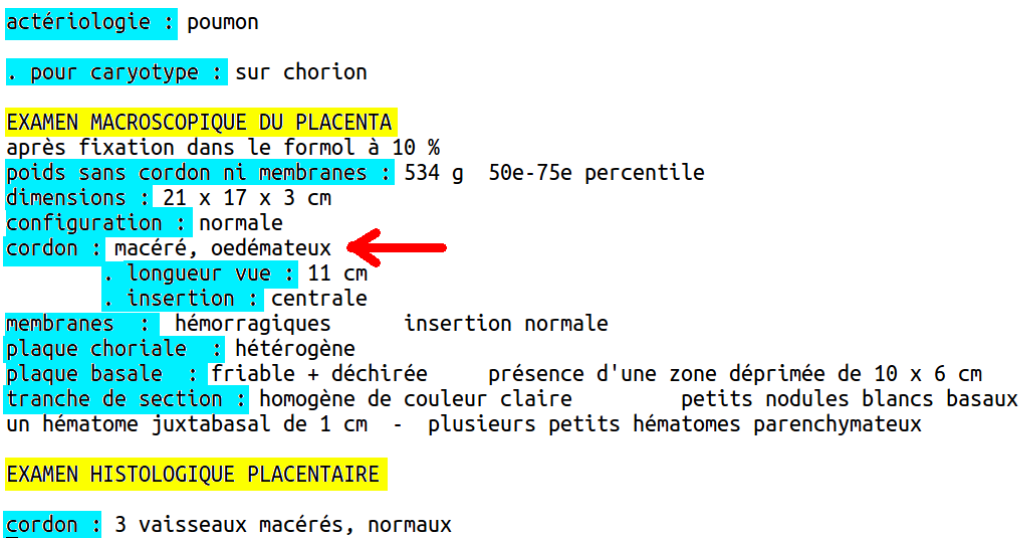
5.4.2 Résultats de *MET.Map.Flexible* utilisée seule

Le *flexible tree matching* (FTM) utilisé seul a rapidement exposé un problème important : le mapping de deux cas est très lent. Sur un processeur bicœur (Intel Core i5-5257U 2.7 GHz), faire correspondre deux arbres de 500 nœuds avec le FTM met environ une heure si on fait faire 100 itérations à l'algorithme de Metropolis-Hastings qui supporte le FTM.

L'ensemble de paramètres (cf. section 4.6.1.4) qui après divers tests a retenu notre attention est le suivant :

2. Il s'agit en effet de la partie la moins générique du processus, puisqu'utilisant des expressions rationnelles faites pour nos besoins, mais cependant facilement adaptables

FIGURE 5.4: Erreur de détection de nœuds au niveau *fin*



On observe que les deux nœuds suivant `cordon` sont clairement des fils de `cordon`. Cependant *`MET.Seg.Simple`* les a considérés comme des frères.

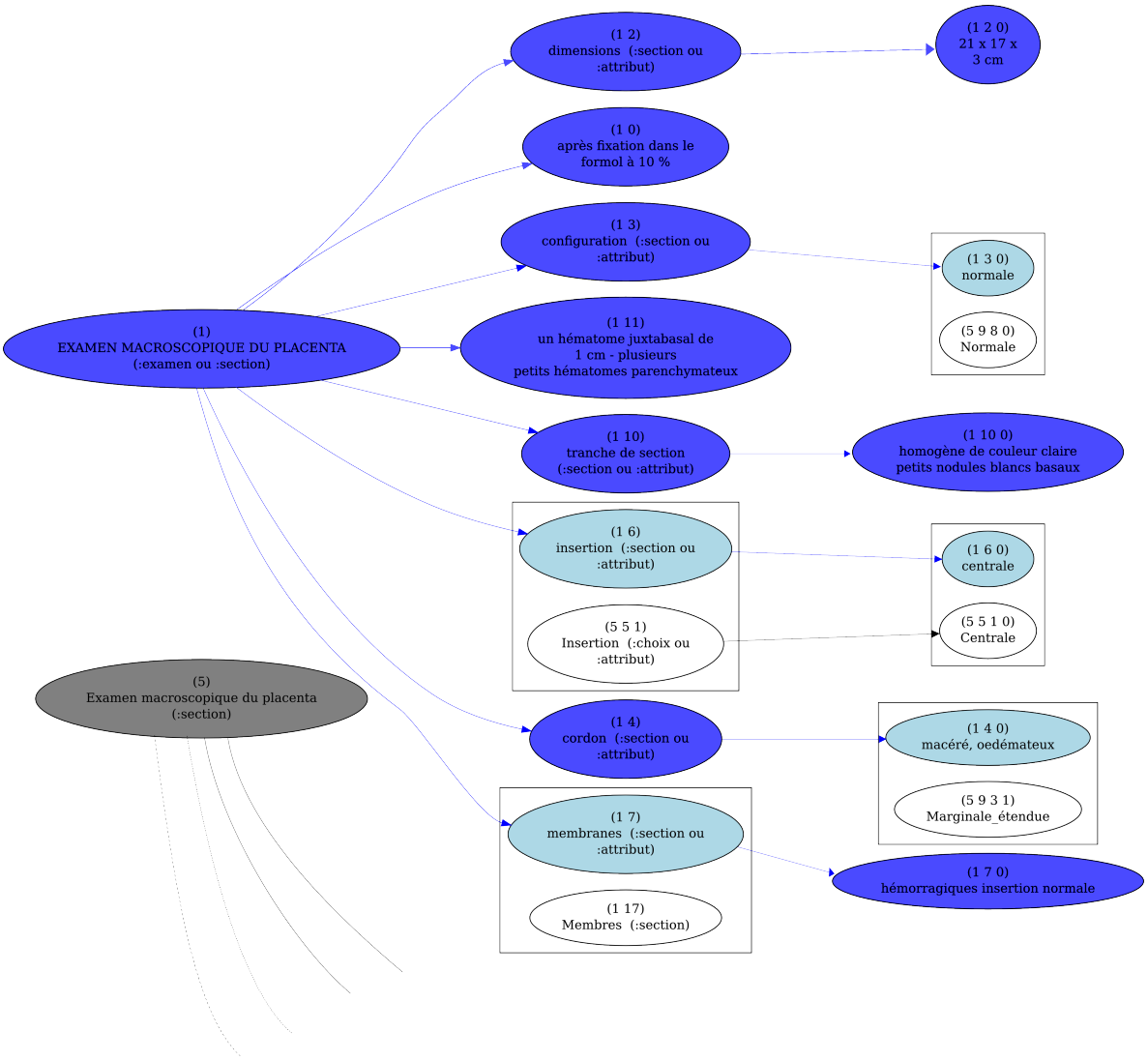
— poids du terme d'ascendance : $w_a = 0,21$;
— poids du terme de fratrie : $w_s = 0,25$;
— coût d'une mise en correspondance avec le nœud vide : $w_n = 1,2$.

Ces valeurs sont bien entendu dépendantes de la fonction choisie pour calculer la distance entre les labels. Pour rappel, la nôtre est exposée en section 4.6.1.2.1. Les autres paramètres ont été laissés comme suggéré par KUMAR, TALTON et al. (2011). Sur une base de 10 comptes rendus (sélectionnés aléatoirement dans notre corpus), ces paramètres nous ont donné en moyenne un bruit (pourcentage de nœuds de l'arbre cas mis en correspondance avec un nœud incorrect de l'arbre modèle, créant ainsi de fausses informations) de 23.7% et un silence (pourcentage de nœuds qui ont été mis en correspondance avec le nœud vide alors qu'un nœud pertinent existait) de 8.6%.

Nous avons remarqué un comportement récurrent dans l'algorithme de flexible tree matching : il tend à commencer par mettre en correspondance les feuilles des deux arbres. Étant donné que les feuilles ne peuvent pas induire de coût d'ascendance elles semblent être les alignements les moins coûteux au début. Le problème est que dans nos arbres cas les feuilles sont les nœuds les moins pertinents à mettre en correspondance : elles correspondent en général aux observations qui sont spécifiques au cas et qui sont celles qui ressemblent le moins souvent à un nœud dans l'arbre modèle. Ceci nous mène à penser que le fait de supprimer les feuilles de l'arbre cas ou de détailler le cas des feuilles dans le calcul de la distance entre les labels pourrait être utile, afin d'éviter d'induire en erreur l'algorithme puisque si des feuilles sont incorrectement mises en correspondance alors la mise en correspondance de leurs pères sera perturbée. Cette perturbation se voit en figure 5.5 au mapping du nœud (1 3 0) : ce nœud est mis en correspondance avec un nœud du modèle qui se trouve avoir le même label, mais qui ne concerne pas la configuration du placenta. Ce mapping non pertinent induira un coût d'ascendance

pour son père, ce qui peut expliquer que le nœud (1 3) ne soit pas mis en correspondance par la suite. Ce problème n'est pas à imputer à l'algorithme de flexible tree matching, il est une conséquence du domaine d'application et des choix de modélisation lors de la construction de l'arbre modèle.

FIGURE 5.5: Mapping obtenu avec le FTM. Les nœuds provenant de l'arbre cas sont en bleu et ceux de l'arbre modèle en blanc/gris)



5.4.3 Résultats de *MET.Map.Inst* utilisée seule

Utilisée seule, *MET.Map.Inst* donne des résultats qui peuvent être intéressants, mais la méthode n'est pas robuste. Un seul nœud mal catégorisé lors de la segmentation qui a précédé et le mapping peut totalement échouer. Seuls les nœuds appartenant au niveau *grossier* et au niveau *intermédiaire* de l'arbre cas peuvent être mappés. Nous fixons le paramètre *seuil* à 6 (pour rappel, la fonction de

distance entre labels utilisée est $distLabels_{Struct}$, cf. 4.6.1.2.1). C'est celui qui nous a semblé offrir le meilleur compromis entre nœuds incorrectement mappés et nœuds non mappés. La figure 5.6 montre un exemple du mieux qu'il peut être obtenu par cette méthode. On y voit la partie histologie du placenta de l'extrait de compte rendu suivant, qui a pu quasi-intégralement être mappée :

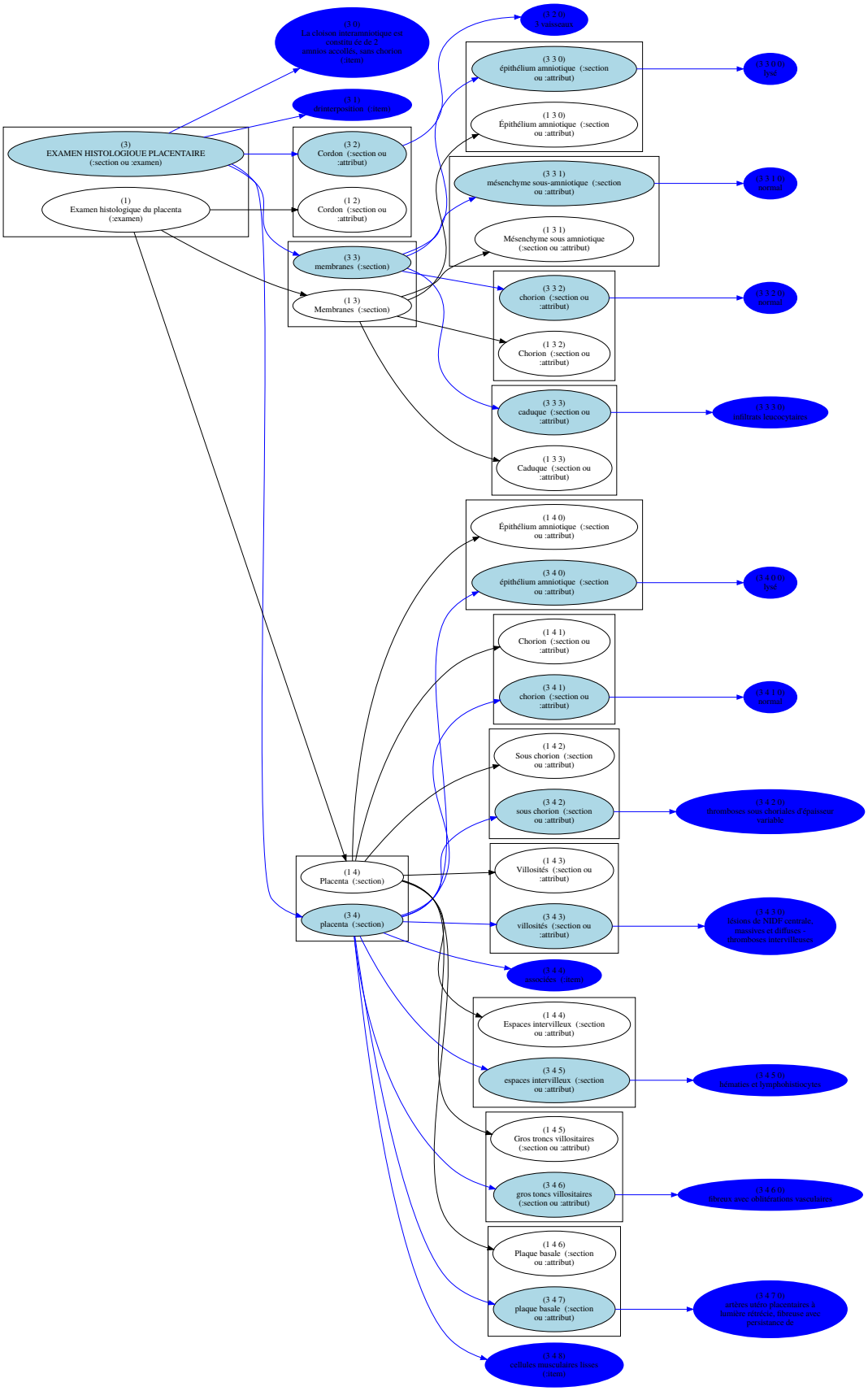
```
EXAMEN HISTOLOGIOUE PLACENTAIRE

La cloison interamniotique est constitu ée de 2 amnios accollés, sans chorion
drinterposition
Cordon : 3 vaisseaux
membranes :
. épithélium amniotique : lysé
. mésenchyme sous-amniotique : normal
. chorion: normal
. caduque : infiltrats leucocytaires
placenta :
. épithélium amniotique : lysé
. chorion: normal
. sous chorion : thromboses sous choriales d'épaisseur variable
. villosités : lésions de NIDF centrale, massives et diffuses - thromboses intervilleuses
associées

. espaces intervilleux : hématies et lymphohistiocytes
. gros toncs villositaires : fibreux avec oblitérations vasculaires
. plaque basale : artères utéro placentaires à lumière rétrécie, fibreuse avec persistance de
cellules musculaires lisses
```

5.4.4 Résultats de *MET.Map.Hybride* et conclusion sur l'homogénéisation de cas

Pour rappel, *MET.Map.Hybride* a pour but de changer la manière dont le processus de *flexible tree matching* débute. Au lieu d'être initialisé aléatoirement, le premier mapping est obtenu par *MET.Map.Inst*. Le temps de calcul est indépendant de l'initialisation, mais notre intuition était qu'il serait possible de réduire le nombre d'itérations du FTM en fournissant un mapping initial plus intelligemment qu'en générant un mapping aléatoire. Nous remarquons tout d'abord que les paramètres que nous avions trouvé expérimentalement pour *MET.Map.Flexible* (cf. section 5.4.2) cessent d'être ceux qui produisent les meilleurs résultats dans le cas de *MET.Map.Hybride*. En effet, avec les paramètres cités en 5.4.2, le FTM a une forte tendance à perdre des liens qui avaient pu être établis dès la phase d'instanciation (par *MET.Map.Inst*).

Pour retrouver des résultats similaires à ceux obtenus à la section 5.4.2 mais avec cette fois-ci seulement 50 itérations, les paramètres sont maintenant :

FIGURE 5.6: Mapping obtenu par *MET.Map.Inst*



— poids du terme d'ascendance : $w_a = 0,3$;
— poids du terme de fratrie : $w_s = 0,1875$;
— coût d'une mise en correspondance avec le nœud vide : $w_n = 1$.

Du fait de l'obtention tardive de ACC[1,3], nous avons malheureusement peu de mesures objectives sur un échantillon représentatif du corpus pour motiver ce qui suit, mais notre intuition à l'heure actuelle est que ce type d'algorithme de Monte-Carlo est viable pour homogénéiser un cas, mais difficile à utiliser en l'état sans avoir au préalable obtenu des arbres cas faisant déjà sens, et qu'il faut donc accorder un grand soin à l'étape de segmentation. Nous reviendrons là-dessus dans les perspectives.

Chapitre 6

Perspectives et conclusion

Dans ce dernier chapitre, nous conclurons quant aux travaux effectués et aux résultats obtenus. Après un rappel des points de la méthode sur lequels nous n'avons pas été en mesure d'expérimenter soit par manque de temps soit à cause de l'obtention trop tardive du corpus, nous reviendrons sur le mapping d'arbres cas et modèles, sur les prétraitements du corpus, sur l'intérêt d'une ontologie de domaine, puis nous terminerons par quelques perspectives et par la conclusion de cette thèse.

6.1 Divergences entre les réalisations et la méthode prévue

Du fait de l'obtention tardive des comptes rendus, certaines parties de la méthode élaborée n'ont pas pu être expérimentées :

- segmentation des cas en arbres par apprentissage de CRF (*MET.Seg.Apprentissage*) : l'annotation manuelle des cas pour permettre au système d'apprendre la meilleure segmentation est une tâche longue qui n'a pas pu être réalisée dans le temps de la thèse. Au vu du temps restant, du fait que *MET.Seg.Simple* fournissait déjà un premier résultat et que la contribution principale de la thèse devait en premier lieu être l'application du mapping flexible d'arbres, il a été choisi de ne pas dépenser de temps sur la réalisation de cette méthode ;
- constitution de [ARB.A.H et mesure de distances entre des arbres de [ARB.A.H via une comparaison des nœuds par mesure de similarité sémantique ;
- intervention des fœtopathologistes et évaluation du système (pour *MET.Sim.Txt* et *MET.Sim.Sem*, afin de faire établir par les experts l'utilisabilité de ces méthodes sans ontologie dédiée à la fœtopathologie) : la disponibilité des praticiens n'ayant pas coincidé avec l'emploi du temps des expérimentations, cette évaluation n'a pas pu être réalisée dans le cadre de la thèse.

Plus généralement, nous nous sommes arrêtés en ce qui concerne les expérimentations à l'homogénéisation des arbres cas. L'expérimentation de la mesure de similarité entre les cas eux-mêmes

qui est l'étape suivante, sera conduite dans le contexte du projet ANR ACCORDYS qui se poursuit jusqu'en 2017.

6.2 Mapping entre arbre cas et arbre modèle

Les méthodes de mapping basées sur des algorithmes de Monte Carlo telles que le *flexible tree matching* nous semblent être une voie intéressante à poursuivre, mais nous pouvons maintenant émettre quelques réserves. Notamment, la lenteur du processus est un problème, et ce pour trois raisons :

— Affiner le mapping nécessite de faire itérer l'algorithme un grand nombre de fois, et ce coût gêne l'augmentation du nombre d'itérations ;
— Obtenir un processus de FTM qui donne des résultats intéressants nécessite de trouver les bons paramètres (poids des terme d'ascendance poids et de fratrie et coût d'une mise en correspondance avec le nœud vide). Cela nécessite des expérimentations, qui sont rendues encore plus fastidieuses s'il faut attendre la complétion de chaque mapping ;
— Réutiliser cette méthode dans d'autres spécialités médicales est rendu clairement plus difficile, car obtenir un mapping en 20 minutes peut être à la rigueur acceptable dans le cas de maladies rares car les comptes rendus sont peu nombreux et arrivent peu fréquemment, mais pas lorsque les examens sont beaucoup plus fréquents.

Dans cette optique, nous voyons deux pistes notables à explorer :

— Faire une passe de profiling et d'optimisation poussée de notre implantation du FTM. Actuellement, notre implantation effectue un certain nombre d'opérations qui pourraient être mises en cache, par exemple elle parcourt de manière systématique les arbres pour trouver les nœuds qui ne sont pas encore touchés par un arc du mapping. C'est une information qui pourrait être cachée. De plus, nous utilisons un certain nombre d'abstractions du langage Clojure (telles que les *zippers*, déjà évoqués, ou les *reducers*, qui permettent facilement de paralléliser le code), et nous observons en utilisant un outil de profiling tel que VisualVM [1] que la majorité du temps de calcul est actuellement passée dans les fonctions relatives à ces abstractions. Il est donc possible que ces abstractions aient un impact sur le temps de calcul qui puisse être corrigé ;
— Limiter l'espace de recherche du FTM. Ceci passera par une limitation du nombre de nœuds dans les arbres avant le mapping : on peut par exemple concevoir que le mapping puisse être d'abord réalisé sur le niveau *grossier* et le niveau *intermédiaire* (sur les deux ou trois premiers niveaux des arbres cas et modèle), et ensuite une fois sur chaque paire de sous-arbres du cas et du modèle dont les racines ont pu être mappées précédemment. Le problème est qu'il est toujours possible dans notre corpus de trouver des cas où une information ne se trouve pas du tout au niveau ou l'on l'attendait. C'est toutefois cette optique qui sera privilégiée lors de l'intégration de notre algorithme au système du projet Accordys.

On pourrait citer une autre piste d'amélioration de la rapidité du calcul de mapping : la distribution

1. http://visualvm.java.net

des calculs du FTM dans le cloud sur un cluster de calcul. Logistiquement, cette option est grandement facilitée par les solutions d'Amazon ou de DigitalOcean par exemple. Aussi, et même sans modifier notre implantation, ceci aurait permis de lancer en parallèle un certain nombre de processus de FTM sur le même compte rendu mais avec des paramètres différents, et aurait facilité les expérimentations. Cependant, dans le cas d'Accordys cette option nous est rendue quasi-inaccessible du fait de l'interdiction de confier les comptes rendus à un tiers, et de l'obligation qu'*in fine* le système tourne dans l'hôpital et que les comptes rendus n'en sortent pas. Il nous est donc impossible de lancer par exemple un ensemble d'instances Amazon EC2 pour distribuer le calcul.

Parmi les méthodes qui n'ont pas été explorées pour évaluer des similarités entre données arborescentes, nous pouvons citer l'analyse en composantes principale d'arbres, telle qu'appliquée par Alfaro et al. (2014) pour étudier les effets de l'âge sur des structures artérielles (représentées sous-forme d'arbres), et les méthodes de machine learning basées sur des *tree kernels* (Moschitti 2006). Ces méthodes pourraient être utiles pour comparer les arbres de ARB.H.

6.3 Prétraitement du corpus et post-traitement des arbres

Les résultats de mappings que nous avons empiriquement observés ne nous permettent pas de dire si l'état de ACC (les erreurs d'OCR notamment) a eu un impact important sur le mapping. En effet, en l'état actuel de notre travail, les nœuds incorrectement mappés à l'issue de *MET.Map.Flexible* ou *MET.Map.Hybride* nous semblaient plus largement être dus à un nombre insuffisant d'itérations plutôt qu'à une distance d'édition trop grande entre les nœuds. Cependant, il est assez clair que ces erreurs d'OCR ont eu un impact sur l'annotation, qui est restée limitée à cause de ces erreurs, une passe de correction de ACC serait donc profitable si l'on souhaite appliquer une méthode de comparaison sémantique.

Une autre étape manquante est la catégorisation des fins de ligne. Nos nœuds sont en effet toujours liés aux lignes, et il pourrait être utile de savoir si une fin de ligne n'a pas de valeur sémantique (est n'est là que pour la mise en page) ou bien si elle est significative, autrement dit si elle est sciemment utilisée par le praticien pour séparer deux éléments de données. La méthode proposée lors de cette thèse suppose que nous sommes toujours de le second cas, ce qui occasionne la création de nœuds non nécessaires dans certains arbres cas (notamment lorsqu'un compte rendu contient tout un paragraphe de texte qui ne peut pas simplement être scindé sur la base des lignes). Ceci devra être corrigé par Accordys grâce à un prétraitement du corpus : Zweigenbaum et al. (2016) présentent la méthode par apprentissage non supervisé qui sera appliquée pour effectuer ce typage des fins de ligne.

Une réalisation qui n'a pas été faite mais pourrait être la suite de ce travail est le post-traitement des feuilles des arbres cas de ARB.H (ou ARB.A.H). Ces feuilles restent en effet toujours du texte libre, et une approche par patrons lexico-syntaxiques pourrait rendre certaines des informations terminales (celles du niveau *détaillé* que nous ne segmentions pas plus) exploitables. Ceci permet de préparer les patrons selon l'endroit où l'on se trouve dans l'arbre, et donc d'être le plus pertinent possible dans la manière dont ces informations terminales sont analysées.

Également, si l'on reconnaît que dans [ARB.H] (ou [ARB.A.H]) ce sont souvent les mêmes nœuds qui sont renseignés par des paragraphes de texte au niveau *détaillé* (c'est le cas par exemple de la conclusion, la plupart du temps), il est possible de construire un modèle vectoriel spécifiquement pour le texte contenu dans ces nœuds-là, et de les comparer par similarité cosinus.

6.4 Ontologie dédiée au domaine

Nous avons fait trois observations importantes lors de l'analyse de notre corpus annoté :

— les annotations ne recouvrent que 41% du texte, qui en principe, au vu de la spécificité du vocabulaire employé, pourrait être annoté en quasi-intégralité avec la bonne RTO,
— les annotations contribuent à changer les similarités produites,
— les annotations les plus fréquentes sont déjà pertinentes, même sans avoir effectué de corrections d'erreurs d'OCR au préalable.

Accordys devra confirmer l'intérêt de ceci via la validation par les fœtopathologistes, mais ces observations nous indiquent que le développement d'une ontologie spécifique à la fœtopathologie serait profitable à Accordys, conjointement avec – comme dit précédemment – un système de correction des erreurs d'OCR ou bien un système d'annotation pouvant fonctionner même en présence de ces erreurs. La construction de cette ontologie a par ailleurs déjà débuté.

6.5 Retour sur le continuum de structuration

Originalement, l'idée d'utiliser une représentation en arbres pour les cas n'était pas motivée par le fait (énoncé en introduction) qu'une telle structuration semblait épouser la structure latente de notre corpus et pourrait donc permettre une informatisation des corpus. La représentation arborescente que nous avons décrite tout au long de ce mémoire était pour nous une représentation nous permettant de progresser sur le continuum de structuration (voir 1.1.1), mais pas une fin en soi. La raison pour laquelle nous nous sommes intéressé au formalisme des logiques $\mathcal{OSF}$ (évoquées en section 2.3.1.2) est que celles-ci nous paraissaient, de par leur structure en treillis (permettant donc d'ajouter des liens transversaux à nos arbres, par exemple lorsque deux données dans un cas se recoupent : l'une dans les observations de l'une des sections et l'autre dans la conclusion du cas), les inférences qu'elles permettent, ainsi que les opérateurs de calcul de similarités qui peuvent s'appuyer dessus (Ontañón et Plaza 2013), très adaptées à être le formalisme de représentation principal de nos cas. Nous voyions donc les arbres cas sur lesquels nous avons travaillé comme des étapes intermédiaires sur le continuum de structuration vers un formalisme logique de représentation. Ce travail d'augmentation de la formalisation ne sera malheureusement pas réalisé dans le cadre d'Accordys, le projet étant actuellement en passe de se terminer.

6.6 Conclusion

Comme nous en avons déjà parlé précédemment, les résultats objectifs que nous avons obtenus – ainsi que le fait que nous n'avons finalement pas eu le temps de comparer les résultats en termes de similarité entre les cas de l'approche structurelle avec ceux des approches textuelles – ne nous autorisent pas à trop d'affirmations, mais ceux-ci nous amènent à penser que :

— les méthodes à base d'algorithmes de Monte-Carlo telles que celle présentée dans cette thèse ont leur utilité dans le cadre du traitement de l'information de comptes rendus médicaux, et sont certainement une bonne réponse à la variabilité des représentations ;
— malgré leur souplesse, ces méthodes ne permettent pas de se passer d'un prétraitement efficace sur le corpus, autrement les mappings produits sont difficiles à exploiter ;
— de façon plus générale, ces méthodes gagnent à être incluses dans une chaîne de traitements plus ad-hoc, plus adaptée au domaine.

D'un point de vue ingénierie des connaissances (IC), nous regrettons de ne pas avoir été en mesure de conclure quant à la pertinence des rapprochements entre cas effectués de façon purement structurelle (*MET.Sim.Struct*, sur ARB) par rapport à celle des cas rapprochés par modèles vectoriels sur des texte annotés et enrichis sémantiquement (*MET.Sim.Sem*, sur ACC.A.TIA). Ceci nécessite en effet une grande implication en temps de la part des praticiens, et nous espérons que ce travail pourra être réalisé dans le cadre d'Accordys. Cependant, nous espérons que les résultats obtenus sur les différentes variantes de *MET.Sim.Sem* (selon la variante de ACC.A utilisée) peuvent déjà donner une bonne indication de l'impact de cet enrichissement sémantique, et donc présager de la contribution positive d'une ontologie dédiée à la fœtopathologie. Nous espérons aussi, dans le domaine de l'IC, avoir ouvert la voie à d'autres flux de travail basés sur la transformation d'arbres, procédé qui nous semble être un pont intéressant entre le texte et les représentations logiques plus communément utilisées en IC, pont que nous avons tenté de formaliser via la notion de continuum de structuration.

D'un point de vue développement informatique, nous pensons être arrivés à un prototype adaptable fournissant divers outils (notamment une implantation générique de l'algorithme de *flexible tree matching*) pour qui souhaite appliquer ces techniques à un autre domaine. Ces outils sont en effet assez directement utilisables si les arbres manipulés ne sont pas trop grands. Les règles de segmentation que nous avons utilisées restent génériques et adaptables, et la création d'un modèle de cas est une étape assez simple pour qui connaît le domaine d'application. De plus, du fait de la généricité du FTM, cette étape de création de modèle peut ne pas être nécessaire pour qui souhaite directement faire correspondre des arbres cas ensemble, pour peu que ces derniers varient moins que les nôtres. Ce prototype devrait sous peu être rendu disponible sous license libre. Nous regrettons toutefois que ce prototype ne sache pas à l'heure actuelle gérer le passage à l'échelle (gestion d'une base de cas complète au lieu de la gestion d'un cas de fœtopathologie particulier).

Pour finir, d'un point de vue gestion des données médicales, nous pensons que ce prototype peut également être utile aux médecins pour l'archivage des cas futurs, leur permettant d'écrire leurs cas selon la méthode qui leur est la plus pratique, et de laisser ensuite l'ordinateur remettre ça sous la forme attendue par le système d'archivage. Il s'agit en effet d'un cas d'utilisation où la lenteur de

l'instanciation du modèle de cas n'est pas un problème, les cas archivés n'ayant pas vocation à être utilisables par le système immédiatement.

Chapitre 7

Annexe

7.1 Extraits de code

7.1.1 Typage des lignes d'un compte rendu

```
(def line-re
  (re-pattern
   (str #"([^\pL\pN]*)"
        #"([^:]+)"
        #"(:)?\s*(.+)?")))

(defn guess-type-and-split-line
  "If second arg is given, compares lines with exam labels in model instead of
  just determining if line _looks_ like an exam line, but divides distance by 2
  if line looks like an exam, so line shape is still taken into account."
  [line & [{:keys [exam-labels dist-max exam-shape-factor]
            :or {dist-max 4, exam-shape-factor 0.5}}]]
  (when (not (empty? line))
    (when-let [[_ init-marker label colon text-after-colon]
               (re-matches line-re line)]
      (let [label (s/trim label)
            num-letters (count (re-seq #"\pL" label))
            mostly-letters? (and (> num-letters 4)
                                 (> (/ num-letters (count label))
                                    0.5))
            looks-like-exam? (and mostly-letters?
                                  (= label (.toUpperCase label))
```

```
                              (not text-after-colon))
            exam-labels (map #(.toLowerCase %) exam-labels)
            type  (when (not-empty label)
                    (or
                     (when (and (empty? exam-labels)
                                looks-like-exam?)
                       #{:section :examen})

                     (when (not-empty exam-labels)
                       (let [[min-dist label]
                             , (apply min-key first
                                      (map #(do [(StringUtils/getLevenshteinDistance
                                                  label %)
                                                 %])
                                           exam-labels))
                             dist (* (double min-dist)
                                     (if looks-like-exam? exam-shape-factor 1))]
                         (when (<= dist dist-max)
                           #{:section :examen [label dist]})))

                     (when (and colon text-after-colon)
                       #{:section :attribut})

                     (when colon
                       #{:section})

                     (when init-marker
                       #{:item}))]
        (when type
          [type label text-after-colon])))))
```

7.1.2 Fonction de mesure de similarité entre locus

```
(defn levenshtein+compat+proportion+alts [loc-cr loc-model]
  (let [ncr    (z/node loc-cr)
        nmodel (z/node loc-model)
        compatible? (or (not (empty? (set/intersection (set (:type ncr))
                                                       (set (:type nmodel)))))
                        (and (or (string? nmodel)
                                 (empty-typed-ltnode? nmodel))
                             (or (string? ncr)
                                 (contains? (:type ncr) :item))))
```

```
        l1  (loc-label loc-cr)
        l2s (cons (loc-label loc-model)
                  (:alt-labels nmodel))]
    (apply min (for [l2 l2s]
                 (* (/ (double (StringUtils/getLevenshteinDistance
                                 (.toLowerCase l1) (.toLowerCase l2)))
                       (min (count l1) (count l2)))
                    (if compatible? 1 2))))))
```

Table des figures

Références

Aamodt, Agnar (2004). « Knowledge-intensive case-based reasoning in creek ». In : *Advances in Case-Based Reasoning*. Springer, p. 1–15.

Aamodt, Agnar et Enric Plaza (1994). « Case-based reasoning: Foundational issues, methodological variations, and system approaches ». In : *AI communications* 7.1, p. 39–59.

Adams, M (2007). *Functional Pearl: Scrap Your Zippers*.

Afantenos, Stergos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Lydia-Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Péry-Woodley, Laurent Prévot et al. (2012). « An empirical resource for discovering cognitive principles of discourse organisation: the ANNODIS corpus ». In : *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA).

Ahmed, Mobyen Uddin, Shahina Begum, Peter Funk et Ning Xiong (2009). « Multi-modal and multipurpose case-based reasoning in the health sciences ». In : *Proc. 8th Int. Conf. Artif. Intell., Knowl. Eng. Data Bases (AIKED)*, p. 378–383.

Aimé, Xavier (2015). « Eléments de réflexion sur l'utilisation de corpus pour la construction d'ontologies ». In : *IC2015*.

Ait-Kaci, Hassan (2007). « Description logic vs. order-sorted feature logic ». In : *Computational Linguistics* 6, p. 19.

Akutsu, Tatsuya, Takeyuki Tamura, Daiji Fukagawa et Atsuhiro Takasu (2014). « Efficient exponential-time algorithms for edit distance between unordered trees ». In : *Journal of Discrete Algorithms* 25, p. 79–93.

Alfaro, Carlos A, Burcu Aydin, Carlos E Valencia, Elizabeth Bullitt et Alim Ladha (2014). « Dimension reduction in principal component analysis for trees ». In : *Computational Statistics & Data Analysis* 74, p. 157–179.

Armengol, Eva et Enric Plaza (2003). « Relational case-based reasoning for carcinogenic activity prediction ». In : *Artificial Intelligence Review* 20.1-2, p. 121–141.

Arnold, Corey W, Suzie M El-Saden, Alex AT Bui et Ricky Taira (2010). « Clinical case-based retrieval using latent topic analysis ». In : *AMIA Annual Symposium Proceedings*. T. 2010. American Medical Informatics Association, p. 26.

Ashburner, Michael, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig et al. (2000). « Gene Ontology: tool for the unification of biology ». In : *Nature genetics* 25.1, p. 25–29.

Augsten, Nikolaus, Michael Böhlen, Curtis Dyreson et Johann Gamper (2008). « Approximate joins for data-centric XML ». In : *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, p. 814–823.

Augsten, Nikolaus, Michael Böhlen et Johann Gamper (2005). « Approximate matching of hierarchical data using pq-grams ». In : *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, p. 301–312.

Baader, Franz (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge university press.

Bach, Kerstin, Christian Severin Sauer, Klaus-Dieter Althoff et Thomas Roth-Berghofer (2014). « Knowledge Modeling with the Open Source Tool myCBR. » In : *KESE@ ECAI*.

Bachimont, Bruno (2000). « Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances ». In : *Ingénierie des connaissances: évolutions récentes et nouveaux défis*, p. 305–323.

Bachimont, Bruno, Antoine Isaac et Raphaël Troncy (2002). « Semantic commitment for designing ontologies: A proposal ». In : *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*. Springer, p. 114–121.

Badra, Fadi, Julien Cojan, Amélie Cordier, Jean Lieber, Thomas Meilender, Alain Mille, Pascal Molli, Emmanuel Nauer, Amedeo Napoli, Hala Skaf-Molli et al. (2009). « Knowledge acquisition and discovery for the textual case-based cooking system WIKITAAABLE ». In : *8th International Conference on Case-Based Reasoning-ICCBR 2009, Workshop Proceedings*, p. 249–258.

Baneyx, Audrey (2007). « Construire une ontologie de la Pneumologie Aspects théoriques, modèles et expérimentations ». Thèse de doct. Université Pierre et Marie Curie-Paris VI.

Begum, Shahina, Mobyen Uddin Ahmed, Peter Funk, Ning Xiong et Mia Folke (2011). « Case-based reasoning systems in the health sciences: a survey of recent trends and developments ». In : *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 41.4, p. 421–434.

Bergmann, Ralph et Martin Schaaf (2003). « Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match? » In : *J. UCS* 9.7, p. 608–626.

Berners-Lee, Tim, James Hendler, Ora Lassila et al. (2001). « The semantic web ». In : *Scientific american* 284.5, p. 28–37.

Bichindaritz, Isabelle (2004). « Mémoire: Case based reasoning meets the semantic web in biology and medicine ». In : *Advances in Case-Based Reasoning*. Springer, p. 47–61.

Biébow, Brigitte, Sylvie Szulman et Nathalie Aussenac-Gilles (2005). « Modélisation du domaine par une méthode fondée sur l'analyse de corpus ». In : *Ingénierie des connaissances*. L'Hamattan, p. 49–71.

Blei, David M, Andrew Y Ng et Michael I Jordan (2003). « Latent dirichlet allocation ». In : *the Journal of machine Learning research* 3, p. 993–1022.

Bouaud, Jacques, Benoît Habert, Adeline Nazarenko et Pierre Zweigenbaum (2000). « Regroupements issus de dépendances syntaxiques sur un corpus de spécialité: catégorisation et confrontation à deux conceptualisations du domaine ». In : *Jean Charlet, Manuel Zacklad, Gilles Kassel, et Didier Bourigault, éditeurs, Ingénierie des connaissances: évolutions récentes et nouveaux défis*, p. 275–290.

Brachman, Ronald J (1978). *A Structural Paradigm for Representing Knowledge*. Rapp. tech. DTIC Document.

Burke, Robin D, Kristian J Hammond, Vladimir Kulyukin, Steven L Lytinen, Noriko Tomuro et Scott Schoenberg (1997). « Question answering from frequently asked question files: Experiences with the faq finder system ». In : *AI magazine* 18.2, p. 57.

Calvanese, Diego, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini et Riccardo Rosati (2007). « Tractable reasoning and efficient query answering in description logics: The DL-Lite family ». In : *Journal of Automated reasoning* 39.3, p. 385–429.

Carpenter, Bob (1992). « The Logic of Typed Feature Structures. Number 32 in Cambridge Tracts in Theoretical Computer Science ». In :

Charlet, Jean (2002). « L'ingénierie des connaissances: développements, résultats et perspectives pour la gestion des connaissances médicales ». In :

— (2013). « Agrégation de contenus et de connaissances pour raisonner a partir de cas de dysmorphologie foetale ». In : *Premier atelier du SIG IMIA francophone*.

Charlet, Jean, Gunnar Declerck, Ferdinand Dhombres, Pierre Gayet, Patrick Miroux et Pierre-Yves Vandenbussche (2012). « Construire une ontologie médicale pour la recherche d'information: problématiques terminologiques et de modélisation ». In : *23es journées francophones d'Ingénierie des connaissances*, p. 33–48.

Chen, Weimin (2001). « New algorithm for ordered tree-to-tree correction problem ». In : *Journal of Algorithms* 40.2, p. 135–158.

Cohen, Sholom et Nerya Or (2014). « A general algorithm for subtree similarity-search ». In : *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, p. 928–939.

Cohen, William W, Pradeep D Ravikumar, Stephen E Fienberg et al. (2003). « A Comparison of String Distance Metrics for Name-Matching Tasks. » In : *IIWeb*. T. 2003, p. 73–78.

Cojan, Julien et Jean Lieber (2011). « An algorithm for adapting cases represented in ALC ». In : *IJCAI*, p. 2582–2589.

Ho-Dac, Lydia-Mai, Marie-Paule PÉry-Woodley et Ludovic Tanguy (2010). « Anatomie des structures énumératives ». In : *Traitement Automatique des Langues Naturelles*, publication–numérique.

Damerau, Fred J (1964). « A technique for computer detection and correction of spelling errors ». In : *Communications of the ACM* 7.3, p. 171–176.

Deerwester, Scott C., Susan T Dumais, Thomas K. Landauer, George W. Furnas et Richard A. Harshman (1990). « Indexing by latent semantic analysis ». In : *JAsIs* 41.6, p. 391–407.

Degoulet, Patrice et Marius Fieschi (2012). *Introduction to clinical informatics*. Springer Science & Business Media. isbn : 978-3790820010.

Demaine, Erik D, Shay Mozes, Benjamin Rossman et Oren Weimann (2007). « An optimal decomposition algorithm for tree edit distance ». In : *Automata, languages and programming*. Springer, p. 146–157.

Dhombres, Ferdinand, Jean-Marie Jouannic, Marie-Christine Jaulent et Jean Charlet (2010). « Choix méthodologiques pour la construction d'une ontologie de domaine en médecine prénatale ». In : *21èmes Journées Francophones d'Ingénierie des Connaissances*. Ecole des Mines d'Alès, p. 171–182.

Diniz-Filho, José Alexandre F, Thannya N Soares, Jacqueline S Lima, Ricardo Dobrovolski, Victor Lemes Landeiro, Mariana Pires de Campos Telles, Thiago F Rangel et Luis Mauricio

Bini (2013). « Mantel test in population genetics ». In : *Genetics and Molecular Biology* 36.4, p. 475–485.

Dubois, J, L Guespin et M Giacomo (1994). « C. et J.-B. Marcellesi, J.-P. Mevel, 1994 ». In : *Dictionnaire de linguistique et des sciences du langage*.

Fauconnier, J, Mouna Kamel, Bernard Rothenburger et Nathalie Aussenac-Gilles (2013). « Apprentissage supervisé pour l'identification de relations sémantiques au sein de structures énumératives parallèles ». In : *Actes de la 20e Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2013)*, p. 132–145.

Fieschi, Marius (1984). *Intelligence artificielle en médecine, des systèmes experts*. Masson. isbn : 978-2-225-80280-5.

Firth, John R (1957). « A synopsis of linguistic theory, 1930-1955 ». In : *In Studies in Linguistic Analysis*, p. 1–32.

Friedman, Carol, Lyudmila Shagina, Yves Lussier et George Hripcsak (2004). « Automated encoding of clinical documents based on natural language processing ». In : *Journal of the American Medical Informatics Association* 11.5, p. 392–402.

Funk, Christopher, William Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, K Bretonnel Cohen, Lawrence E Hunter et Karin Verspoor (2014). « Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters ». In : *BMC bioinformatics* 15.1, p. 1.

Galopin, Alexandre, Jacques Bouaud, Suzanne Pereira et Brigitte Séroussi (2014). « Using an ontological modeling to evaluate the consistency of clinical practice guidelines: application to the comparison of three guidelines on the management of adult hypertension. » In : *MIE*, p. 38–42.

Généreux, Michel, Amália Mendes et Thierry Hamon (2013). « Experiments in synonymy: term extraction and mapping to concepts ». In : *Terminologie et Intelligence artificielle (TIA)*.

Gierl, Lothar, Mathias Bull et Rainer Schmidt (1998). « CBR in Medicine ». In : *Case-Based Reasoning Technology*. Springer, p. 273–297.

Grefenstette, Gregory (1992). « Sextant: Exploring unexplored contexts for semantic extraction from syntactic analysis ». In : *Proceedings of the 30th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 324–326.

Grémy, François (1987). « Informatique médicale ». In : *Flammarion*.

Gruber, Thomas R (1993). « A translation approach to portable ontology specifications ». In : *Knowledge acquisition* 5.2, p. 199–220.

Guarino, Nicola (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. T. 46. IOS press.

Harispe, Sébastien, Sylvie Ranwez, Stefan Janaqi et Jacky Montmain (2013). « Semantic Measures for the Comparison of Units of Language, Concepts or Instances from Text and Knowledge Base Analysis ». In : *arXiv preprint arXiv:1310.1285*.

Harris, Zellig S (1954). « Distributional structure. » In : *Word*.

Hassanpour, Saeed et Curtis P Langlotz (2015). « Information extraction from multi-institutional radiology reports ». In : *Artificial intelligence in medicine*.

Hassena, Anouar Ben et Laurent Miclet (2009). « Dissimilarité analogique et apprentissage d'arbres ». In : *RÀPC-2009*, p. 61.

Hearst, Marti A et Christian Plaunt (1993). « Subtopic structuring for full-length document access ». In : *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, p. 59–68.

Jiang, Tao, Lusheng Wang et Kaizhong Zhang (1994). « Alignment of trees—an alternative to tree edit ». In : *Combinatorial Pattern Matching*. Springer, p. 75–86.

Jimeno Yepes, Antonio et Alan R Aronson (2012). « Knowledge-based and knowledge-lean methods combined in unsupervised word sense disambiguation ». In : *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*. ACM, p. 733–736.

Jimeno-Yepes, Antonio J, Bridget T McInnes et Alan R Aronson (2011). « Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation ». In : *BMC bioinformatics* 12.1, p. 223.

Kamel, Mouna, Mustapha Mojahid et Bernard Rothenburger (2012). « "Quand rédiger c'est décrire"-Mise en forme matérielle des textes et construction d'ontologies à partir de textes ». In : *23es Journées Francophones d'Ingénierie des Connaissances-IC 2012*, p. 133–148.

Kiefer, Christoph et Abraham Bernstein (2008). *The creation and evaluation of isparql strategies for matchmaking*. Springer.

Kifer, Michael, Georg Lausen et James Wu (1995). « Logical foundations of object-oriented and frame-based languages ». In : *Journal of the ACM (JACM)* 42.4, p. 741–843.

Kouylekov, Milen et Bernardo Magnini (2006). « Combining lexical resources with tree edit distance for recognizing textual entailment ». In : *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Springer, p. 217–230.

Kozima, Hideki (1993). « Text segmentation based on similarity between words ». In : *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 286–288.

Kumar, Ranjitha, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R Klemmer et Jerry O Talton (2013). « Webzeitgeist: Design mining the web ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, p. 3083–3092.

Kumar, Ranjitha, Jerry O Talton, Salman Ahmad, Tim Roughgarden et Scott R Klemmer (2011). « Flexible tree matching ». In : *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*. AAAI Press, p. 2674–2679.

Lafferty, John, Andrew McCallum et Fernando CN Pereira (2001). « Conditional random fields: Probabilistic models for segmenting and labeling sequence data ». In : *Proceedings of the International Conference on Machine Learning*. T. 18, p. 282–289.

Lakkaraju, Praveen, Susan Gauch et Mirco Speretta (2008). « Document similarity based on concept tree distance ». In : *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*. ACM, p. 127–132.

Landauer, Thomas K et Susan T Dumais (1997). « A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. » In : *Psychological review* 104.2, p. 211.

Landauer, Thomas K, Danielle S McNamara, Simon Dennis et Walter Kintsch (2013). *Handbook of latent semantic analysis*. Psychology Press.

Lavergne, Thomas, Olivier Cappé et François Yvon (2010). « Practical Very Large Scale CRFs ». In : *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Uppsala, Sweden : Association for Computational Linguistics, p. 504–513. URL : http://www.aclweb.org/anthology/P10-1052.

LEHMANN, Fritz (1992). *Semantic networks in artificial intelligence*. Elsevier Science Inc.

LENZ, Mario (1998). « Defining knowledge layers for textual case-based reasoning ». In : *Advances in Case-Based Reasoning*. Springer, p. 298–309.

LEVENSHTEIN, Vladimir I (1966). « Binary codes capable of correcting deletions, insertions, and reversals ». In : *Soviet physics doklady*. T. 10. 8, p. 707–710.

LI, Fei, Hongzhi WANG, Jianzhong LI et Hong GAO (2014). « A survey on tree edit distance lower bound estimation techniques for similarity join on XML data ». In : *ACM SIGMOD Record* 42.4, p. 29–39.

LI, Guoliang, Xuhui LIU, Jianhua FENG et Lizhu ZHOU (2008). « Efficient similarity search for tree-structured data ». In : *Scientific and Statistical Database Management*. Springer, p. 131–149.

LIEBER, Jean, Mathieu D'AQUIN, Fadi BADRA et Amedeo NAPOLI (2008). « Modeling adaptation of breast cancer treatment decision protocols in the KASIMIR project ». In : *Applied Intelligence* 28.3, p. 261–274.

LIN, Zhiwei, Hui WANG et Sally MCCLEAN (2010). « Measuring tree similarity for natural language processing based information retrieval ». In : *Natural Language Processing and Information Systems*. Springer, p. 13–23.

LIU, Maofu, Li JIANG et Huijun HU (2015). « Automatic extraction and visualization of semantic relations between medical entities from medicine instructions ». In : *Multimedia Tools and Applications*, p. 1–19.

LUC, Christophe (2001). « Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte ». In : *Actes de la 8e Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2001)*, p. 263–272.

MALAISÉ, Véronique, Pierre ZWEIGENBAUM et Bruno BACHIMONT (2004). « Repérage et exploitation d'énoncés définitoires en corpus pour l'aide à la construction d'ontologie ». In : *Actes de TALN*, p. 269–278.

MANTEL, Nathan (1967). « The detection of disease clustering and a generalized regression approach ». In : *Cancer research* 27.2 Part 1, p. 209–220.

MARQUIS, Pierre, Odile PAPINI et Henri PRADE (2014). *Représentation des connaissances et formalisation des raisonnements*. T. 1. Panorama de l'Intelligence Artificielle. Cepaduès Éditions. ISBN : 9782364930414.

MINSKY, Marvin (1974). « A framework for representing knowledge ». In :

MOSCHITTI, Alessandro (2006). « Making Tree Kernels Practical for Natural Language Learning. » In : *EACL*. T. 113. 120, p. 24.

MOTIK, Boris, Peter F PATEL-SCHNEIDER et Bernardo Cuenca GRAU (2009). « Owl 2 web ontology language direct semantics ». In : *W3C Recommendation* 27.

MUGNIER, Marie-Laure et Michel CHEIN (1996). « Représenter des connaissances et raisonner avec des graphes ». In : *Revue d'intelligence artificielle* 10.1, p. 7–56.

NILSSON, Markus et Mikael SOLLENBORN (2004). « Advancements and Trends in Medical Case-Based Reasoning: An Overview of Systems and System Development. » In : *FLAIRS Conference*, p. 178–183.

Ontañón, Santiago et Enric Plaza (2012). « On knowledge transfer in case-based inference ». In : *Case-Based Reasoning Research and Development*. Springer, p. 312–326.

— (2013). « An Approach to Re-Representation in Relational Learning ». In : *CCIA*, p. 11–20.

Pantazi, Stefan V, José F Arocha et Jochen R Moehr (2004). « Case-based medical informatics ». In : *BMC Medical Informatics and Decision Making* 4.1, p. 19.

Patel-Schneider, Peter F, Patrick Hayes, Ian Horrocks et al. (2004). « OWL web ontology language semantics and abstract syntax ». In : *W3C recommendation* 10.

Pawlik, Mateusz et Nikolaus Augsten (2011). « RTED: a robust algorithm for the tree edit distance ». In : *Proceedings of the VLDB Endowment* 5.4, p. 334–345.

Pereira, Suzanne, Saoussen Sakji, Aurélie Névéol, Ivan Kergourlay, Gaétan Kerdelhué, Elisabeth Serrot, Michel Joubert et Stéfan J Darmoni (2009). « Multi-terminology indexing for the assignment of MeSH descriptors to medical abstracts in French ». In : *AMIA Annual Symposium Proceedings*. T. 2009. American Medical Informatics Association, p. 521.

Périnet, Amandine et Thierry Hamon (2014). « Analyse et proposition de paramètres distributionnels adaptés aux corpus de spécialité ». In : *International Conference Journées d'Analyse statistique des Données Textuelles*, p. 507–518.

Plaza, Enric (1995). « Cases as terms: A feature term approach to the structured representation of cases ». In : *Case-Based Reasoning Research and Development*. Springer, p. 265–276.

Prokofyev, Roman, Gianluca Demartini, Alexey Boyarsky, Oleg Ruchayskiy et Philippe Cudré-Mauroux (2013). « Ontology-based word sense disambiguation for scientific literature ». In : *Advances in Information Retrieval*. Springer, p. 594–605.

Recio-García, Juan A, Pedro A González-Calero et Belén Díaz-Agudo (2014). « jcolibri2: A framework for building Case-based reasoning systems ». In : *Science of Computer Programming* 79, p. 126–145.

Recio-Garcia, Juan A, Belén Diaz-Agudo et Pedro A González-Calero (2007). « Textual cbr in jcolibri: From retrieval to reuse ». In : *Proceedings of the ICCBR 2007 Workshop on Textual Case-Based Reasoning: Beyond Retrieval*, p. 217–226.

Reis, Davi de Castro, Paulo Braz Golgher, Altigran Soares Silva et AlbertoF Laender (2004). « Automatic web news extraction using tree edit distance ». In : *Proceedings of the 13th international conference on World Wide Web*. ACM, p. 502–511.

Resnik, Philip (1995). « Using information content to evaluate semantic similarity in a taxonomy ». In : *arXiv preprint cmp-lg/9511007*.

Richter, Michael M. et Rosina Weber (2013a). « Advanced CBR Elements ». In : *Case-Based Reasoning: A Textbook*. Sous la dir. de Springer-Verlag Berlin Heidelberg. Springer-Verlag Berlin Heidelberg. Chap. 12. isbn : 978-3-642-40166-4.

— (2013b). « Textual CBR ». In : *Case-Based Reasoning: A Textbook*. Sous la dir. de Springer-Verlag Berlin Heidelberg. Springer-Verlag Berlin Heidelberg. Chap. 17. isbn : 978-3-642-40166-4.

Rosskopf, Simon (2015). « Zippers and Data Type Derivatives ». In :

Sahlgren, Magnus (2006). « The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces ». Thèse de doct.

Salton, Gerard, Amit Singhal, Mandar Mitra et Chris Buckley (1997). « Automatic text structuring and summarization ». In : *Information Processing & Management* 33.2, p. 193–207.

Shahbazi, Ali et Jason Miller (2014). « Extended Subtree: A New Similarity Function for Tree Structured Data ». In : *Knowledge and Data Engineering, IEEE Transactions on* 26.4, p. 864–877.

Shasha, Dennis, Kaizhong Zhang et FY Shih (1994). « Exact and approximate algorithms for unordered tree matching ». In : *Systems, Man and Cybernetics, IEEE Transactions on* 24.4, p. 668–678.

Sioutos, Nicholas, Sherri de Coronado, Margaret W Haber, Frank W Hartel, Wen-Ling Shaiu et Lawrence W Wright (2007). « NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information ». In : *Journal of biomedical informatics* 40.1, p. 30–43.

Souvignet, Julien, Hadyl Asfari, Jérémy Lardon, Emilie Del Tedesco, Gunnar Declerck et Cédric Bousquet (2016). « MedDRA® Automated Term Groupings using OntoADR: Evaluation with Upper Gastrointestinal Bleedings ». In : *Expert Opinion on Drug Safety* just-accepted.

Sowa, John F (1976). « Conceptual graphs for a data base interface ». In : *IBM Journal of Research and Development* 20.4, p. 336–357.

Steichen, Olivier, Christel Daniel-Le Bozec, Maxime Thieu, Eric Zapletal et Marie-Christine Jaulent (2006). « Computation of semantic similarity within an ontology of breast pathology to assist inter-observer consensus ». In : *Computers in Biology and Medicine* 36.7, p. 768–788.

Tai, Kuo-Chung (1979). « The tree-to-tree correction problem ». In : *Journal of the ACM (JACM)* 26.3, p. 422–433.

Taira, Ricky K, Stephen G Soderland et Rex M Jakobovits (2001). « Automatic Structuring of Radiology Free-Text Reports1 ». In : *RadioGraphics* 21.1, p. 237–245.

Tanaka, Eiichi et Keiko Tanaka (1988). « The tree-to-tree editing problem ». In : *International Journal of pattern recognition and artificial intelligence* 2.02, p. 221–240.

Tao, Cui, Guoqian Jiang, Weiqi Wei, Harold R Solbrig et Christopher G Chute (2011). « Towards semantic-web based representation and harmonization of standard meta-data models for clinical studies ». In : *AMIA summits on translational science proceedings* 2011, p. 59.

Tchechmedjiev, Andon (2012). « État de l'art: mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances ». In : *JEP-TALN-RECITAL 2012*, p. 295.

Tellier, Isabelle (2010). « Introduction au TALN et à l'ingénierie linguistique ». In : *Polycopié de cours: Univer-sité de Lille* 3.

Thessen, Anne E et Cynthia Sims Parr (2014). « Knowledge extraction and semantic annotation of text from the encyclopedia of life ». In : *PloS one* 9.3.

Thieu, Maxime, Olivier Steichen, Eric Zapletal, Marie-Christine Jaulent et Christel Le Bozec (2004). « Mesures de similarité pour l'aide au consensus en anatomie pathologique ». In : *15èmes Journées francophones d'Ingénierie des Connaissances*. Presses universitaires de Grenoble, p. 225–236.

Töpel, Thoralf, Jens Neumann et Ralf Hofestädt (2007). « A medical case-based reasoning component for the rare metabolic diseases database RAMEDIS ». In : *Computer-Based Medical Systems, 2007. CBMS'07. Twentieth IEEE International Symposium on*. IEEE, p. 7–11.

Tversky, Amos (1977). « Features of similarity ». In : *Psychological Review* 84, p. 327–352.

Virbel, J (1999). « Structures textuelles, planches fascicule 1: Enumérations ». In : *Rapport technique, IRIT. Version* 1.

Virbel, J, C Luc, S Schmid, L Carrio, C Dominguez, MP PERYWOODLEY, C Jacquemin, M Mojahid, T Baccino et C GARCIADEBANC (2005). « Approche cognitive de la spatialisation du langage. De la modélisation de structures spatio-linguistiques des textes à l'expérimentation psycholinguistique: le cas d'un objet textuel, l'énumération ». In : *Agir dans l'espace. Paris: Éditions de la Maison des sciences de l'homme*, p. 233–254.

Wagner, Robert A et Roy Lowrance (1975). « An extension of the string-to-string correction problem ». In : *Journal of the ACM (JACM)* 22.2, p. 177–183.

Watson, Ian et Farhi Marir (1994). « Case-based reasoning: A review ». In : *The knowledge engineering review* 9.04, p. 327–354.

Weber, Rosina O, Kevin D Ashley et Stefanie Brüninghaus (2005). « Textual case-based reasoning ». In : *Knowledge Engineering Review* 20.3, p. 255–260.

Weber, Rosina, Alejandro Martins et R Barcia (1998). « On legal texts and cases ». In : *Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop*, p. 40–50.

Wu, D, Rosina Weber et FC Abramson (2004). « A case-based framework for leveraging nutrigenomics knowledge and personalized nutrition counseling ». In : *Proceedings of the case-based reasoning in the health sciences workshop, European conference on case based reasoning (ECCBR), Madrid*, p. 73–82.

Wu, Zhibiao et Martha Palmer (1994). « Verbs semantics and lexical selection ». In : *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 133–138.

Yang, Rui, Panos Kalnis et Anthony KH Tung (2005). « Similarity evaluation on tree-structured data ». In : *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, p. 754–765.

Zhang, Kaizhong (1995). « Algorithms for the constrained editing distance between ordered labeled trees and related problems ». In : *Pattern Recognition* 28.3, p. 463–474.

Zhang, Kaizhong et Dennis Shasha (1989). « Simple fast algorithms for the editing distance between trees and related problems ». In : *SIAM journal on computing* 18.6, p. 1245–1262.

Zhang, Kaizhong, Rick Statman et Dennis Shasha (1992). « On the editing distance between unordered labeled trees ». In : *Information processing letters* 42.3, p. 133–139.

Zweigenbaum, Pierre, Cyril Grouin et Thomas Lavergne (2016). « Une catégorisation de fins de lignes non-supervisée ». In : *Actes de la conférence conjointe JEP-TALN-RECITAL* 2. url : https://jep-taln2016.limsi.fr/actes/Actes%20JTR-2016/Papers/T102.pdf.