Annotation sémantique floue de tableaux guidée par une ontologie

Gaëlle Hignette

N°/__/__/__/__/__/__/__/__/__/__/

THÈSE

pour obtenir le grade de

Docteur

de

l'Institut des Sciences et Industries du Vivant et de l'Environnement (Agro Paris Tech)

Spécialité : Informatique

présentée et soutenue publiquement
par

Gaëlle HIGNETTE

le 17 décembre 2007

ANNOTATION SEMANTIQUE FLOUE DE TABLEAUX GUIDEE PAR UNE ONTOLOGIE

Directeur de thèse : ***Ollivier HAEMMERLE***

Codirecteurs de thèse : ***Patrice BUCHE, Juliette DIBIE-BARTHELEMY***

Travail réalisé : INRA, UMR518 Mathématiques et Informatique Appliquées, F-75231 Paris Cedex 05
et INRA, UR1204 Méthodologies d'analyse de risque alimentaire Mét@risk, F-75231 Paris Cedex 05

Devant le jury :

M. Patrick BOSC, Professeur, **Université de Rennes 1** ..... **Rapporteur**
Mme Marie-Christine ROUSSET, Professeur, **Université de Grenoble** ..... **Rapporteur**
M. Patrice BUCHE, Ingénieur de Recherche, **INRA** ..... **Examinateur**
Mme Juliette DIBIE-BARTHELEMY, Maître de Conférences, **AgroParisTech** ..... **Examinateur**
Mme Christine FROIDEVAUX, Professeur, **Université Paris Sud** ..... **Examinateur**
M. Ollivier HAEMMERLE, Professeur, **Université de Toulouse 2** ..... **Examinateur**

Résumé

Nous présentons dans ce mémoire une méthode d'annotation de tableaux guidée par les connaissances d'un domaine d'application formalisées dans une ontologie. Après avoir présenté le contexte applicatif et une étude bibliographique sur l'annotation sémantique et l'extraction d'information, nous présentons les différentes étapes de notre système : annotation des cellules, des colonnes puis des relations représentées par le tableau. Nous traitons différemment les données selon qu'elles sont numériques ou symboliques. Nous commençons par déterminer si une colonne d'un tableau contient des données numériques ou symboliques. Les données symboliques sont annotées avec les termes de l'ontologie, en utilisant une comparaison mot à mot des termes employés dans le tableau avec ceux définis dans l'ontologie. Les données numériques sont extraites, ainsi que les unités de mesure employées, et comparées avec les unités et intervalles de valeurs possibles définis dans l'ontologie pour les types de données numériques. Le type de données représenté par chaque colonne du tableau est alors déterminé, en utilisant à la fois le contenu de la colonne (deux méthodes différentes sont employées suivant que la colonne contient des données numériques ou symboliques) et le titre de la colonne. Une fois le type des colonnes reconnu, les relations sémantiques représentées par le tableau sont identifiées en utilisant à la fois le titre du tableau et la signature du tableau, qui est comparée avec la signature des relations sémantiques définies dans l'ontologie. Les relations reconnues sont ensuite instanciées pour chaque ligne du tableau. Les annotations que nous manipulons sont floues, c'est-à-dire qu'au lieu de faire un lien direct entre un élément du tableau et un élément de l'ontologie, nous proposons plusieurs valeurs possibles pour l'annotation, en associant à chaque valeur un degré représentant la confiance que l'on accorde à cette valeur. Les différentes étapes de notre méthode d'annotation de tableaux ont été évaluées expérimentalement, en prenant comme domaine d'application la microbiologie alimentaire.

Mots clefs : annotation sémantique, ontologie, ensembles flous

Abstract

This thesis presents a new method for annotating data tables using the knowledge of an application domain described in an ontology. We first present our applicative context and a bibliographic study of other works about semantic annotation and information extraction. Then we present the different steps of our annotation process, in which we annotate the cells, the columns and the relations of a given data table. Data are not annotated in the same way according to whether they are symbolic or numeric. Thus, our first step is to distinguish between columns containing numeric or symbolic data. Symbolic data are annotated with the terms of the ontology, using a word to word comparison between the terms

used in the data table and the terms defined in the ontology. Numeric data are extracted, along with the units in which those data are expressed : they are compared with the units and range defined in the ontology for numeric data types. The data type for each column is then identified using both the column contents (in a different way according to whether the column is symbolic or numeric) and the column title. When the data type of each column has been recognized, the semantic relations represented by the table are found using both the table title and the table signature which is compared to the signature of the relations defined in the ontology. The relations that are recognized in the table are then instanciated for each line in the table. Our annotation is fuzzy, that is, instead of linking a part of the table directly to its correspondant in the ontology, we give several values for the annotation, each with a confidence degree. The different steps of our annotation method have been evaluated during an experiment on the food microbiology domain.

Keywords : semantic annotation, ontology, fuzzy sets

Table des matières

Introduction

La quantité d'information disponible sur internet est aujourd'hui gigantesque et sa croissance est exponentielle. En effet, d'après la synthèse de [Kobayashi & Takeda, 2000], le nombre d'usagers d'internet double chaque année et le nombre de pages accessibles sur le web croît à un rythme encore plus élevé. Le problème est que plus la quantité d'information disponible est importante, plus cette information devient difficile à appréhender pour un être humain. Dans la vision de [Berners-Lee et al., 2001], « le web sémantique apportera une structure au contenu informatif des pages web, créant un environnement dans lequel des agents logiciels sillonnant les pages internet pourront effectuer des tâches sophistiquées pour les utilisateurs ».

Cependant, le web actuel est encore loin de cet idéal. En effet, pour pouvoir utiliser des agents logiciels, il faut s'affranchir des hétérogénéités de structure et de vocabulaire utilisés pour présenter l'information. Il est important de conserver la liberté d'expression sur le web, qui passe par une liberté dans le choix de la présentation et du vocabulaire utilisé. Une solution au problème de l'hétérogénéité dans la façon de représenter les données est l'annotation sémantique : les auteurs continuent à écrire l'information sous la forme de leur choix, mais en rajoutant des métadonnées, compréhensibles par un ordinateur, qui indiquent ce dont parle le document.

L'annotation sémantique est définie par [Kiryakov et al., 2004] comme « une génération de métadonnées spécifiques et d'un schéma d'utilisation, ayant pour but de rendre possibles de nouvelles méthodes d'accès à l'information et d'améliorer les méthodes existantes ». Il s'agit d'assigner à des entités dans le texte un lien vers leur description sémantique. Les entités dans le texte qui peuvent être annotées sont des mots, des suites de mots, des phrases, des paragraphes, voire le document dans son ensemble. Leur description sémantique associée peut être une description textuelle libre, mais en général il s'agit plutôt d'une description formelle, écrite dans un langage du web sémantique et utilisable par des agents logiciels. Cette description sémantique peut être stockée dans le document lui-même, ou dans un autre document regroupant les descriptions sémantiques de plusieurs entités. La solution de l'externalisation est plus intéressante car elle permet de décrire une seule fois une entité présente dans plusieurs documents, et donc de mettre en évidence l'unicité de l'objet du monde

réel auquel chacun des documents fait référence. Par exemple, si la chaîne de caractères « Paris » est annotée comme étant une ville dans deux documents distincts, on sait juste qu'il est fait mention deux fois de villes, qui sont dans les deux cas représentées par la chaîne de caractères « Paris ». Si par contre dans les deux documents la chaîne de caractères « Paris » est annotée comme correspondant à la ville Paris décrite dans un document de référence commun, alors on sait qu'il s'agit effectivement de la même ville.

Il est aujourd'hui encore rare que l'annotation sémantique soit directement réalisée par les auteurs lors de la publication d'information sur internet. Aussi, un enjeu important est l'annotation sémantique de documents déjà publiés : comme il s'agit d'une gigantesque masse d'information, il faut envisager des techniques d'annotation automatique de ces documents. L'hétérogénéité des documents à annoter, qu'elle soit de structure ou de vocabulaire, est la difficulté majeure rencontrée pour l'annotation automatique. Pour limiter les problèmes liés à cette hétérogénéité, la plupart des systèmes d'annotation se concentrent sur un domaine d'application (en général, le domaine est modifiable mais il faut donner des informations spécifiques à un domaine d'application pour le bon fonctionnement du système) ou sur une tâche précise (par exemple, l'annotation des noms de personnes et de pays...). Beaucoup de systèmes sont en outre spécialisés pour un type de document particulier : e-mails, pages personnelles, sites commerciaux, etc.

L'annotation automatique de documents peut être rapprochée de l'extraction d'information. L'extraction d'information est définie par [Chinchor & Marsh, 1998] comme « l'extraction à partir d'un texte de chaînes de caractères (ou de ces chaînes ayant subi une transformation simple) pour remplir des cases prédéfinies donnant un sens aux chaînes extraites ». Par exemple, pour une entreprise, il est possible d'extraire le nom, le pays, le nom du dirigeant... La différence entre annotation sémantique et extraction d'information tient surtout à la façon dont sont traitées les chaînes de caractères reconnues :

- dans le cas de l'annotation sémantique, la chaîne de caractères n'est pas extraite. Elle reste dans son contexte, et un lien est construit depuis la portion de document contenant la chaîne de caractères vers une description de la signification de cette chaîne ;
- dans le cas de l'extraction d'information, la chaîne de caractères reconnue est extraite du document. La chaîne de caractères est donc examinée en dehors de son contexte. La description de la signification de la chaîne est implicite dans le nom du champ extrait.

Si les traitements des chaînes de caractères reconnues diffèrent entre annotation sémantique (chaîne restant dans le texte, sans traitement) et extraction d'information (chaîne extraite du texte avec une transformation possible), les mêmes méthodes sont en revanche applicables dans les deux cas pour reconnaître quelles sont les chaînes de caractères à annoter ou à extraire, et quelle est leur signification.

Comme nous l'avons dit plus haut, l'annotation sémantique consiste à faire un lien entre une portion de document et sa description sémantique. Lorsque l'annotation est faite pour être lue par des humains, cette description sémantique peut être un simple texte explicatif ajoutant des précisions par rapport à ce qui est présenté dans le document. Par contre, lorsque l'annotation est faite en vue d'être utilisable par des agents logiciels, cette description sémantique est représentée de façon formelle dans une ontologie.

Il n'y a pas de définition unique d'une ontologie. La définition la plus communément utilisée est celle de [Gruber, 1993] : « une ontologie est une spécification explicite d'une conceptualisation ». La notion d'ontologie est utilisée dans de nombreux domaines : [Guarino, 1998] distingue notamment les ontologies au sens philosophique des ontologies pour l'intelligence artificielle. Au sens philosophique, une ontologie est un système de catégorisation reflétant une vision du monde, indépendante de tout langage. Les ontologies pour l'intelligence artificielle, et pour l'informatique en général, sont également fondées sur la nécessité de représentation du monde et de catégorisation ; elles sont cependant vouées à être utilisées par des agents logiciels et sont exprimées dans un langage précis. [Neches et al., 1991] définit l'ontologie en informatique de la façon suivante : « une ontologie définit les termes et relations de base constituant le vocabulaire d'un domaine, ainsi que les règles de combinaison des termes et des relations pour la définition d'extensions du vocabulaire ». La notion d'ontologie contient celle de consensus, comme le montre la définition utilisée par [Bouquet et al., 2004] : « Les ontologies sont des modèles *partagés* d'un domaine qui encodent une vue commune à un ensemble de différents groupes ».

Différents types d'ontologies, avec différents niveaux de complexité, sont recensés dans [Smith & Welty, 2001]. Une ontologie peut être

- une simple liste de termes à utiliser pour représenter un domaine ;
- une taxonomie dans laquelle des termes sont mis en relation avec d'autres termes plus généraux ou plus spécifiques qu'eux dans une hiérarchie commune ;
- un cadre plus complexe dans lequel d'autres relations que la simple hiérarchie sont définies entre les termes, avec la possibilité de définir des contraintes, etc.

Toute définition d'un vocabulaire et de contraintes sur ce vocabulaire peut être considérée comme une ontologie, indépendamment du langage utilisé pour la représenter. Cependant des langages spécifiques ont été développés pour représenter les ontologies dans le cadre du web sémantique, tels que les langages RDF, DAML+OIL, ou OWL [Antoniou et al., 2005]. Ces langages introduisent les notions de concepts et d'instances : un concept est l'entité abstraite décrite dans l'ontologie, une instance du concept est un objet réel qui est représenté par ce concept (par exemple, ce document est une instance du concept "mémoire de thèse").

Le but du travail présenté dans ce mémoire est l'annotation sémantique de

documents, et plus exactement l'annotation de tableaux de données issus du web. En effet, les tableaux de données concentrent l'information sous une forme structurée et synthétique, et sont utilisés pour présenter l'information dans de nombreux domaines (sciences, commerce, géopolitique...). Notre travail se rapproche de l'extraction d'information : en effet, nous cherchons à extraire les relations représentées par des tableaux, afin de pouvoir les utiliser lors de requêtes d'utilisateurs, et nous nous autorisons à transformer les chaînes de caractères extraites pour les agréger en éléments plus complexes. Cependant il nous paraît important de conserver le contexte dans lequel ces informations ont été trouvées, car ce contexte peut être important pour leur interprétation par l'utilisateur. En effet, certaines parties du tableau ne sont pas annotées car ne correspondant pas à des données modélisées dans l'ontologie, mais contiennent des informations supplémentaires utiles sur la façon dont ont été produites les données. Nous utilisons donc une approche mixte : les données sont tout d'abord extraites des documents pour être traitées par notre système ; les annotations produites sont ensuite replacées dans le document. Une annotation peut être construites à partir de plusieurs données situées à différents endroits dans le document. Aussi, les annotations que nous replaçons dans le document contiennent des liens vers les données d'origine.

Nombre de travaux portent sur la détection de tableaux à l'intérieur de documents composites, ainsi que sur la détection de l'orientation de tableaux (voir, par exemple, la synthèse réalisée par [Zanibbi et al., 2004]). Cependant, une fois le tableau détecté et ramené à une orientation « standard », les travaux qui s'attachent à comprendre la sémantique des données présentées à l'intérieur des tableaux se font plus rares (une étude bibliographique de tels travaux est présentée dans le chapitre 2 de ce mémoire de thèse). Il s'agit donc d'un champ de recherche ouvert, avec de nombreux domaines d'application possibles. Nous avons été guidés dans notre approche par un domaine d'application précis, la microbiologie alimentaire.

Notre travail s'inscrit dans le cadre du groupement d'intérêt scientifique Sym'Previus, dont le but est la mise en place et l'amélioration d'un outil d'aide à l'expertise en microbiologie prévisionnelle. Nos partenaires jouent à la fois le rôle d'experts (pour la définition de l'ontologie) et d'utilisateurs du système. Nous avons ainsi pu tester notre système sur des tableaux présentant des données définies comme intéressantes par les futurs utilisateurs du système : ceci nous a permis de dégager des problématiques en partant d'exemples concrets. Bien que ce travail soit appliqué, nous nous sommes attachés, tout au long du développement de notre méthode d'annotation de tableaux, à ce que ce travail reste généralisable à d'autres domaines d'application. Ainsi, pour modifier le domaine d'application il suffit de modifier l'ontologie.

Ce travail de thèse a commencé dans le cadre de la fin du projet RNTL[1]

[1]Réseau National de recherche et d'innovation en Technologies Logicielles, mis en place fin 1999 par le Ministère délégué à la Recherche et le Ministère délégué à l'Industrie

e.dot [e.dot, 2005] qui s'est déroulé de 2003 à 2005. Il s'agissait de « construire des entrepôts de données dans des domaines spécifiques en intégrant de manière automatique des informations découvertes sur le web » en s'appuyant sur « le nouveau format du web, XML ». Le but de ce projet était d'extraire du web des documents intéressants pour un domaine d'application et de les annoter. Les documents sont alors stockés dans un entrepôt de données au format XML, chaque document XML contenant à la fois le document initial et les annotations ajoutées.

Ce travail de thèse s'est poursuivi dans le cadre du projet RNTL WebContent [WebContent, 2007], qui a commencé en 2006 et se poursuivra jusqu'en 2009. Le but de ce projet est de « produire une plate-forme flexible et générique pour la gestion de contenus et l'intégration des technologies du Web Sémantique dans le but de montrer leur utilité sur des applications réelles à fort impact économique ou sociétal ». Il s'agit de construire et de mettre en interaction différents services web permettant de rechercher des documents, de les annoter et de les interroger, notamment pour des applications de veille (scientifique, stratégique ou économique). Dans le cadre de ce projet, nous avons notamment dû reconsidérer le format de nos données pour nous conformer aux standards du web sémantique, le langage RDF pour les annotations et le langage OWL pour la représentation des ontologies.

Le sujet de ce mémoire de thèse est l'annotation de tableaux à l'aide d'une ontologie. Ce travail est appliqué au domaine de la microbiologie alimentaire et a été réalisé dans le cadre deux projets nationaux, e.dot et WebContent. Nous présentons en chapitre 1 le contexte applicatif de notre système d'annotation de tableaux ainsi que les entrées de notre système (tableaux de données et ontologie). Une revue des travaux existants dans le domaine de l'annotation sémantique et de l'extraction de données est proposée en chapitre 2 : nous y présentons des méthodes générales d'annotation ainsi que des méthodes spécifiques à l'annotation de tableaux. Les chapitres suivants sont consacrés à la méthode d'annotation de tableaux à l'aide d'une ontologie mise au point au cours de cette thèse. On considère d'une part qu'un tableau est un ensemble de cellules organisées en colonnes, l'ensemble des colonnes d'un tableau permettant de représenter des relations, et d'autre part que l'ontologie est un ensemble de termes organisés en types, ces types étant liés entre eux par des relations. Notre but est de retrouver quelles sont les relations de l'ontologie représentées par un tableau. Nous procédons par agrégation d'éléments de plus en plus complexes. Pour ce faire, nous annotons tout d'abord les cellules à l'aide des termes présents dans l'ontologie, comme présenté en chapitre 3. Nous étudions ensuite en chapitre 4 l'annotation des colonnes du tableau en fonction des types de l'ontologie. Enfin, le chapitre 5 présente comment nous annotons le tableau avec les relations de l'ontologie. Ce travail de thèse est sous-tendu par la construction d'une application réelle. Nous avons tout d'abord construit un prototype afin de tester la validité de notre méthode d'annotation, mais nous avons pour but de mettre à disposition, via le projet RNTL

WebContent, un service d'annotation sémantique de tableaux pouvant réellement être utilisé par des partenaires extérieurs. Nous consacrons donc le chapitre 6 à la question de l'implémentation de notre méthode d'annotation de tableaux. Enfin, après avoir dressé un bilan de notre méthode d'annotation de tableaux, nous ouvrons sur les perspectives de notre travail en termes d'interrogation des tableaux annotés.

Chapitre 1

Contexte de l'étude

Le système d'annotation de tableaux que nous proposons dans cette thèse est utilisé dans le cadre de la construction d'un entrepôt de données ouvert sur le web. Cet entrepôt participe à l'extension d'un système existant, le système MIEL, que nous présentons en section 1.1. La section 1.2 présente ensuite plus en détails la construction de l'entrepôt de données ouvert sur le web. Les entrées de notre système d'annotation, respectivement les tableaux de données et l'ontologie, sont présentées en sections 1.3 et 1.4.

1.1 Le système MIEL

Le système MIEL (Moteur d'Interrogation ELargie) [Buche & Haemmerlé, 2000, Buche et al., 2003] permet d'effectuer des requêtes, définies graphiquement à l'aide d'une ontologie, simultanément sur une base de données relationnelle et une base de graphes conceptuels. L'interrogation des deux bases se fait de façon transparente pour l'utilisateur, et les résultats sont retournés sous la même forme tabulaire quelle que soit l'origine des données. Les réponses sont ordonnées selon un calcul d'adéquation à la requête, en tenant compte des préférences de l'utilisateur.

Les deux bases existantes du système MIEL sont remplies manuellement par des experts. Le coût d'acquisition manuelle des données est très important : en conséquence, les bases de données sont mises à jour à une fréquence fortement inférieure à la production de nouvelles connaissances. Ce problème de mise à jour de l'information, rencontré dans de nombreux domaines, peut être pallié par une ouverture sur le web, où les informations sont mises à jour de façon beaucoup plus fluide. La solution proposée est de construire un entrepôt de données alimenté automatiquement à partir d'informations récoltées sur le web : cet entrepôt pourrait être ainsi régulièrement mis à jour en lançant de nouvelles requêtes sur le web. L'entrepôt de données construit devra pouvoir être interrogé de la même manière que les deux autres bases, sans changements majeurs pour les utilisateurs. Le

système MIEL, étendu à l'interrogation de l'entrepôt de données XML alimenté par le web, est appelé MIEL++. Un schéma de fonctionnement de ce système est présenté en figure 1.1.

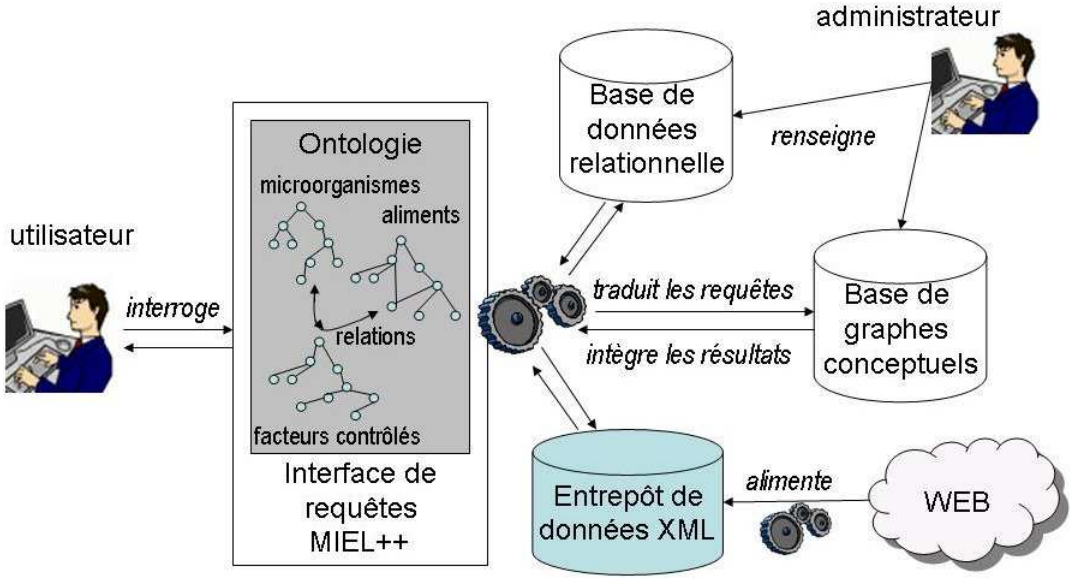

FIG. 1.1 – Le système MIEL++

Le but dans la mise en place du système MIEL++ est de conserver une interface de requêtes commune entre l'entrepôt de données construit à partir du web et les deux autres bases remplies manuellement. L'interface de requêtes actuelle du système MIEL, dans laquelle l'utilisateur sélectionne graphiquement dans l'ontologie les éléments qui l'intéressent, est un point fort de l'application car elle permet une interrogation facilitée pour des utilisateurs non informaticiens. De plus, l'ontologie étant structurée avec des relations de type subsomption entre les différents aliments et les différents microorganismes, il est possible d'exploiter ces hiérarchies pour apporter à l'utilisateur des résultats approchés lorsque qu'il n'y a pas suffisamment de réponses exactes à une requête.

Pour pouvoir interroger l'entrepôt de données en utilisant la même ontologie que celle déjà utilisée pour l'interrogation conjointe de la base de données relationnelle et de la base de graphes conceptuels, il faut donc que les données de l'entrepôt soient préalablement annotées en utilisant cette ontologie. C'est le but de ce travail de thèse, qui consiste en l'annotation sémantique des données provenant du web, avant leur intégration à l'entrepôt de données.

1.2 Les étapes de la construction de l'entrepôt de données

La figure 1.2 illustre la façon dont est construit l'entrepôt de données, selon le schéma défini lors du projet e.dot.

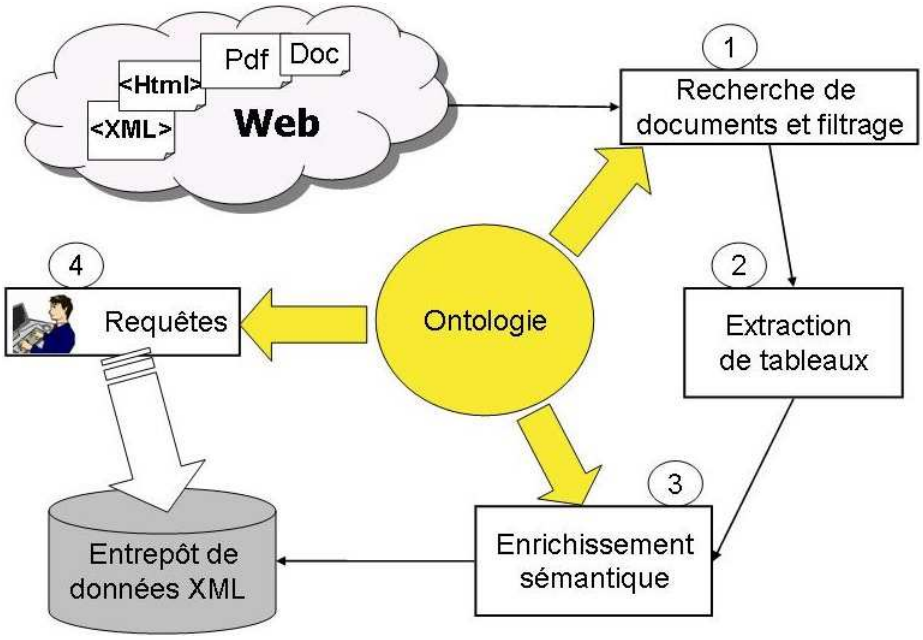

FIG. 1.2 – Mise en place et interrogation de l'entrepôt de données XML

1. Il s'agit tout d'abord de récupérer sur le web les informations intéressantes pour le domaine d'application. Pour cela, un moteur de recherche sur le web est utilisé, avec deux types de critères de recherche :
   - des critères heuristiques permettant de trouver des publications contenant des tableaux : la présence de mot-clefs tels que *"abstract"* et *"references"* et le format du fichier en PDF sont de bons indices pour trouver des publications ; des mot-clefs tels que *"table"* ou *"tab"* permettent d'identifier les documents contenant des tableaux ;
   - des critères liés aux types de données à récupérer : des combinaisons de termes de l'ontologie du domaine sont utilisées pour découvrir automatiquement des documents en lien avec le domaine représenté par cette ontologie. Il est également possible d'utiliser des mot-clefs fournis par un utilisateur dans le cadre d'une recherche précise.

   Les documents ainsi récupérés sont ensuite validés manuellement afin de s'assurer que seuls les documents pertinents pour le domaine d'application étudié sont pris en compte.

2. Les publications sous forme de documents PDF sont traitées par un logiciel de reconnaissance des caractères, puis les tableaux sont extraits semi-automatiquement des documents.
3. Les tableaux sont ensuite enrichis sémantiquement : l'ontologie du domaine est utilisée pour annoter les tableaux. C'est ce travail d'enrichissement sémantique des tableaux de données qui fait l'objet du travail présenté dans ce mémoire de thèse.
4. Une fois les données présentes dans l'entrepôt, il faut pouvoir les interroger avec la même interface de requêtes que pour le système MIEL existant. Cette interrogation fait partie des perspectives qui font suite à notre travail.

1.3 Les tableaux à annoter

Les données intégrées dans l'entrepôt proviennent de tableaux de données publiés sur internet dans différents types de documents tels que rapports de projets, supports de cours ou publications scientifiques. Nous avons choisi de nous intéresser uniquement aux données présentées sous forme de tableaux car ils correspondent à une représentation synthétique des données dans un document. Dans notre domaine d'application, la microbiologie alimentaire, la plupart des résultats expérimentaux sont donnés sous forme de tableaux dans les publications. L'importance des tableaux est immédiatement visible lorsque l'on observe la façon dont est utilisé le système MIEL actuel : la plupart des données renseignées manuellement dans les deux bases existantes de MIEL proviennent de tableaux ou de graphiques dans les publications d'origine. En outre, d'un point de vue pratique, les données intéressantes à extraire d'un document sont plus faciles à repérer si elles sont regroupées dans un tableau que si elles sont disséminées à l'intérieur du texte.

Nous nous intéressons dans ce travail uniquement à des tableaux sous forme canonique, telle que définie par [Tijerino et al., 2005].

Définition 1.1 *Un* schéma *d'une table canonique est un ensemble fini* $S = \{L_1, \ldots, L_n\}$ *de* labels. *A chaque* label L_i *correspond un* domaine D_i. *Soit* $D = D_1 \cup \ldots \cup D_n$. *Une table canonique* T *ayant comme schéma* S *est un ensemble de fonctions* $T = \{t_1, \ldots, t_k\}$ *de* S *vers* D *avec la restriction que, pour chaque fonction* t_j *avec* $1 \leq j \leq k$, *pour tout* i *tel que* $1 \leq i \leq n$, *on a* $t_j(L_i) \in D_i$. *La représentation graphique d'une table sous format canonique est un tableau, dans lequel les labels forment les titres des colonnes et chaque ligne représente une fonction* t_j *: la valeur de* $t_j(L_i)$ *est ainsi représentée dans la* j^{eme} *ligne (ou* $j + 1^{eme}$ *si l'on considère les titres des colonnes comme la première ligne), à la* i^{eme} *colonne dans le tableau.*

Le tableau 1.1 donne un exemple de représentation graphique d'un tableau sous format canonique.

Microorganism	Water Activity Growth Limit Value
Escherichia coli	0.95
Salmonella spp.	0.95
Listeria monoctyogenes	0.92
Stahphylococcus aureus	0.86

TAB. 1.1 – Meat and poultry pathogens of concern and their water activity growth limits (issu de [Decagon Devices, Note])

1.4 L'ontologie utilisée pour l'annotation

Dans le système MIEL, les données de la base de données relationnelles et de la base de graphes conceptuels sont indexées selon une ontologie de domaine. Au moment de l'interrogation, l'utilisateur a accès aux données via une interface graphique reposant sur l'ontologie. Pour notre domaine d'application, l'utilisateur peut ainsi choisir les aliments, microorganismes et facteurs expérimentaux sur lesquels il veut obtenir des informations. La taxonomie des aliments, celle des microorganismes et la liste des facteurs expérimentaux constituent une ontologie du domaine de la microbiologie alimentaire, dont la structure est déterminée par la base de données. Cette ontologie a été constituée lors du projet Sym'Previus, par plusieurs experts en microbiologie alimentaire. Elle a été construite à partir des données, enrichie au fur et à mesure pour représenter les données rentrées dans la base de données relationnelle du système MIEL. Nous souhaitons annoter les tableaux issus du web avec la même ontologie du domaine de la microbiologie alimentaire, afin de pouvoir ensuite interroger les tableaux annotés en utilisant la même interface graphique que celle déjà en place dans le système MIEL.

L'ontologie est essentiellement composée de *termes* organisés entre eux pour représenter le domaine de la microbiologie alimentaire.

Définition 1.2 *Un terme est une suite de mots ayant une signification propre : on distinguera terme de l'ontologie (nom d'un type ou d'une relation, valeur pour un type symbolique) et terme du web (terme provenant d'un tableau isssu du web : contenu d'une case, titre de colonne ou titre du tableau).*

Nous avons été amenés à redéfinir plus formellement la structure de l'ontologie que nous utilisons dans notre processus d'annotation, par souci de généricité. Notre travail d'annotation de tableaux guidée par une ontologie est alors applicable à d'autres domaines, simplement en changeant d'ontologie, à condition que la nouvelle ontologie respecte la structure que nous allons présenter dans ce chapitre.

L'ontologie utilisée pour notre travail d'annotation de tableaux comporte trois sortes de composants, représentatifs du vocabulaire et des connaissances spécifiques au domaine d'application : des types symboliques (section 1.4.1), des

types numériques (section 1.4.2) et des relations (section 1.4.3). Nous avons choisi cette structure car elle est applicable à de très nombreux domaines, et elle est en outre assez intuitive, car elle s'approche d'un modèle relationnel. L'ontologie comporte, en plus des types symboliques et numériques et des relations, deux listes moins spécifiques au domaine d'application, qui peuvent toutefaois être adaptées s'il existe des habitudes linguistiques particulières au domaine étudié :

- la liste des indicateurs de résultat absent regroupe tous les termes utilisés pour représenter le fait qu'une mesure n'a pas pu être effectuée ou n'a pas donné de résultat : par exemple, "no result" ou "NS" (pour Not Specified) ;
- la *stopword list* est une liste de mots ayant un rôle grammatical sans avoir de contenu sémantique fort, tels que les articles, les conjonctions, les pronoms, etc... Elle permet d'éliminer dans un terme tous les mots n'ayant pas de réelle importance dans la signification de ce terme.

1.4.1 Types symboliques

Les types symboliques sont utilisés pour représenter des données dont les valeurs sont des chaînes de caractères.

Définition 1.3 *Un type symbolique est défini par :*

- *le nom du type : c'est un* terme*, composé d'un ou plusieurs mots ;*
- *la taxonomie du type : c'est l'ensemble des valeurs possibles pour ce type, organisées suivant une hiérarchie de subsomption. Chaque valeur possible est un* terme*, composé d'un ou plusieurs mots.*

Notre ontologie du domaine de la microbiologie alimentaire comprend trois types symboliques :

- "Microorganism" représente les microorganismes du domaine étudié (microorganismes pathogènes, microorganismes qui détériorent la qualité des produits ou microorganismes utilisés dans les processus agro-alimentaires) ;
- "Food products" représente les produits alimentaires dans lesquels les microorganismes sont susceptibles de se développer ;
- "Response" représente la réponse d'un microorganisme à un traitement : croissance, absence de croissance, destruction.

La figure 1.3 présente un extrait de la taxonomie du type symbolique ayant pour nom « Food products ».

1.4.2 Types numériques

Les types numériques sont utilisés pour représenter des données dont les valeurs sont numériques.

Définition 1.4 *Un type numérique est défini par les informations suivantes :*

- *le nom du type : c'est un* terme*, composé d'un ou plusieurs mots ;*

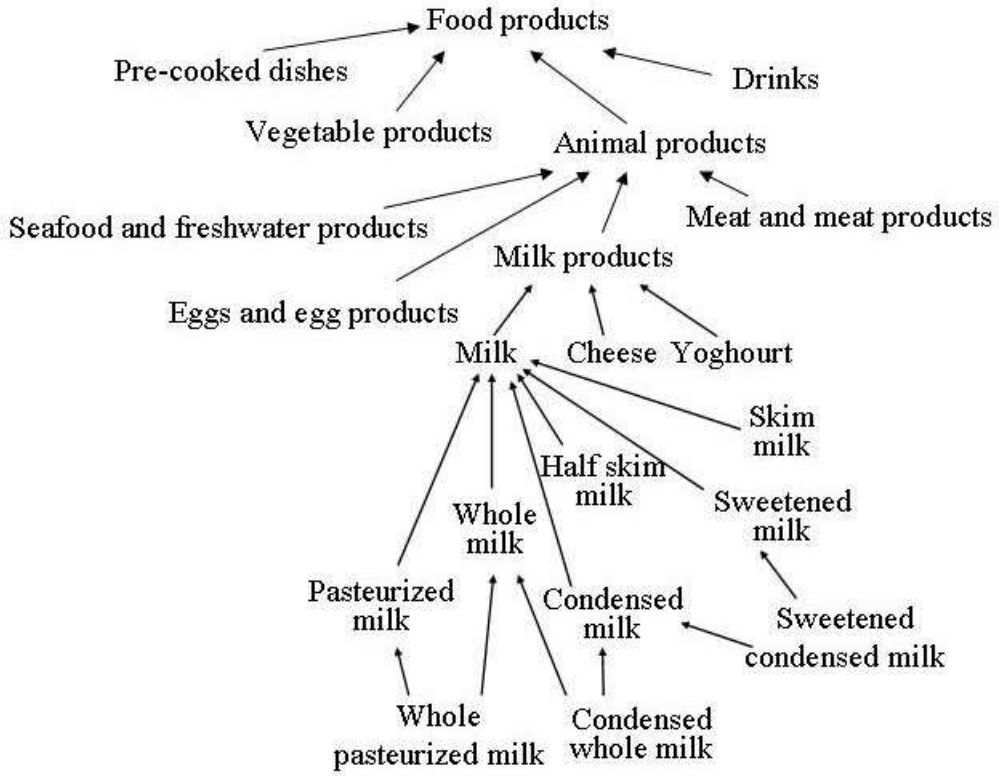

FIG. 1.3 – Extrait de la taxonomie du type « Food products »

- *les unités dans lesquelles peuvent être exprimées les données de ce type, ou #NONE si les données de ce type sont habituellement exprimées sans unité ;*
- *l'intervalle de valeurs possibles pour ce type.*

Dans notre ontologie, les types numériques correspondent soit à des mesures représentant les résultats d'une expérience, soit à des facteurs contrôlés dont la valeur est définie par le cadre expérimental. Nous avons identifié 18 types numériques pour notre ontologie du domaine de la microbiologie alimentaire. Ces types sont listés dans le tableau 1.2.

1.4.3 Relations

Les relations représentent les liens que l'on peut établir entre plusieurs types.

Définition 1.5 *Une relation est définie par les informations suivantes :*
- *le nom de la relation : c'est un* terme*, composé d'un ou plusieurs mots ;*
- *la signature de la relation : c'est l'ensemble des types numériques et/ou symboliques qui sont reliés par cette relation. Cette signature est composée du domaine et du co-domaine de la relation :*

- *le co-domaine de la relation est limité à un seul type ;*
- *le domaine de la relation est composé d'un ou de plusieurs types.*

Dans notre domaine d'application, une instance de relation correspond à une mesure expérimentale ou à la synthèse d'un ensemble de mesures. Pour faciliter la compréhension pour les différents acteurs de notre domaine d'application, nous appelons le co-domaine d'une relation le type résultat de la relation : il s'agit en effet de ce que la relation mesure, en termes de résultat expérimental. Les types de données constitutifs du domaine de la relation sont appelés les types d'accès de la relation. Il s'agit de l'ensemble des données dont la combinaison influe sur la valeur du type résultat. Dans le cadre de notre application, il s'agit du microorganisme et/ou du produit alimentaire impliqué dans la mesure, ainsi que des facteurs expérimentaux contrôlés dans le cadre d'un plan d'expérience. Les relations construites pour notre domaine d'application, avec leur type résultat et leurs types d'accès, sont présentées dans le tableau 1.3.

La notion de type résultat et de types d'accès est facilement généralisable à d'autres domaines. Par exemple, dans le domaine de l'aéronautique, une relation "Poids de l'avion" aura pour type résultat un poids et pour type d'accès un modèle d'avion.

Conclusion du chapitre

Nous avons présenté dans ce chapitre le contexte applicatif dans lequel nous effectuons l'annotation de tableaux. Les tableaux sont annotés en utilisant exclusivement les connaissances du domaine apportées par l'ontologie. Notre but est de reconnaître quelles relations sémantiques de l'ontologie sont représentées par un tableau, et d'instancier ces relations pour chacune de lignes du tableau. Avant de nous attaquer à la description de notre méthode d'annotation de tableaux à l'aide d'une ontologie, nous présentons dans le chapitre suivant une étude bibliographique des travaux d'annotation sémantique et d'extraction d'information.

nom du type	explication	unités	min	max
AW water activity	Activité de l'eau	$\#NONE$	0	1
CO2	Taux de dioxyde de carbone	%	0	100
Colony count concentration	Concentration en microorganismes	cfu	-	-
D reduction	Temps nécessaire pour diminuer la concentration en microorganismes d'un log	mins, secs	0	-
EH redox potential	Potentiel redox	mV	-	-
Growth rate	Paramètre dans les modèles de croissance d'un microorganisme	h^{-1}	-	-
Lag time	Paramètre dans les modèles de croissance d'un microorganisme	h	-	-
N2	Taux d'azote gazeux	%	0	100
NACL	Taux de sel	%	0	100
Number outbreaks deaths	Nombre d'épidémies, de malades ou de morts	$\#NONE$	-	-
O2	Taux de dioxygène	%	0	100
PH	pH	$\#NONE$	0	14
Samples positive	Echantillons dans lesquels le microorganisme est présent	$\#NONE$, %	0	-
Samples tested	Nombre d'échantillons	$\#NONE$	0	-
Temperature	Température	°C,°F	-	-
Time	Temps de conservation	mins,hr, days,weeks	0	-
Year	Année de l'évènement	$\#NONE$	-	-
Ymax	Paramètre dans les modèles de croissance d'un microorganisme	cfu	-	-

TAB. 1.2 – Types numériques de l'ontologie du domaine de la microbiologie alimentaire

Relation	Type résultat	Types d'accès
Prevalence	Samples positive	Food Prod., Micro.
Contamination level	Colony count concentration	Food Prod., Micro.
Growth kinetics	Colony count concentration	Food Prod., Micro., Time, Temp.
D reduction	D reduction	Food Prod.,Micro. Time, Temp.
Growth parameter - AW	AW water activity	Micro.
Growth parameter - pH	PH	Micro.
Growth parameter - Temperature	Temperature	Micro.
Growth parameter - NaCl	NACL	Micro.
Product property - AW	AW water activity	Food Prod.
Product property - pH	PH	Food Prod.
Product property - redox potential	EH redox potential	Food Prod.
outbreaks	Number outbreaks deaths	Food Prod., Micro.
specific growth rate	Growth rate	Food Prod., Micro., Temp.
ymax	Ymax	Food Prod., Micro., Temp.
lag time	Lag time	Food Prod., Micro., Temp.
response	Response	Food Prod., Micro., Time, Temp.

Tab. 1.3 – Relations de l'ontologie du domaine de la microbiologie alimentaire

Chapitre 2

Travaux connexes en annotation sémantique et extraction d'information

Le travail présenté dans ce document consiste en l'annotation sémantique de tableaux de données. Nous présentons donc dans ce chapitre une étude bibliographique portant sur l'annotation sémantique, d'abord sur n'importe quel type de document, puis plus précisément sur des tableaux. Comme cela a été expliqué en introduction, annotation sémantique et extraction d'information partagent les mêmes méthodologies quand il s'agit d'identifier dans un document quelles sont les chaînes de caractères à annoter ou extraire et quelle est la signification de ces chaînes. Ce chapitre présente donc également une étude bibliographique de travaux d'extraction d'information.

Les principaux algorithmes d'extraction d'information sont présentés en section 2.1. Les travaux d'annotation sémantique s'appliquant à divers types de documents plus ou moins structurés sont présentés en section 2.2, tandis que les travaux plus orientés vers l'annotation de tableaux sont présentés en section 2.3. Nous présentons enfin en section 2.4 les premiers travaux d'annotation de tableaux réalisés avant le début de cette thèse par Fatiha Saïs dans le cadre du projet e.dot.

2.1 Algorithmes d'extraction d'information pour des documents non ou faiblement structurés

Cette section présente les algorithmes d'extraction d'information les plus connus, qu'ils soient supervisés (section 2.1.1) ou non supervisés (section 2.1.2).

2.1.1 Algorithmes d'extraction d'information par apprentissage supervisé

L'algorithme $(LP)^2$ [Ciravegna, 2001, Ciravegna, 2003] a été conçu pour l'annotation en XML de pages web faiblement structurées. Il utilise des informations sur la forme grammaticale des mots (nom, verbe, etc., singulier ou pluriel), sur le typage de certains mots reconnus dans un dictionnaire (noms de villes, de pays...) et sur la reconnaissance des entités nommées (noms de personnes, d'organisations, dates...). L'entrée de l'algorithme est un ensemble de pages annotées manuellement, la sortie est un ensemble de règles d'annotation. L'algorithme $(LP)^2$ procède en deux étapes : tout d'abord l'induction de règles d'annotation, qui permettent l'insertion de balises XML (certaines règles permettent l'insertion d'une balise ouvrante, d'autres d'une balise fermante), ensuite l'induction de règles de correction permettant de déplacer les balises afin de corriger les erreurs et imprécisions des règles d'annotation précédentes. Pour chaque balise ouvrante ou fermante de l'annotation manuelle (appelée exemple), un motif comprenant les n mots avant et après la balise est généré. Ce motif est ensuite généralisé : chacun des mots du motif peut être généralisé en le représentant par son lemme, sa classe syntaxique (nom, verbe, chiffre...), sa typographie (majuscules, minuscules...) ou son type (reconnu dans le dictionnaire ou entité nommée), ou encore par un *joker* autorisant n'importe quel mot. Toute combinaison de généralisation (ou mot conservé à l'identique) du motif initial devient un motif candidat pour devenir un règle d'annotation. On conserve uniquement les motifs qui, appliqués à l'ensemble des documents annotés manuellement, permettent de couvrir un nombre minimal d'exemples (i.e. permettent l'insertion d'un certain nombre de balises au même endroit que ce qui a été fait en annotation manuelle), et dont le taux d'erreur (i.e. le nombre de balises insérées au mauvais endroit rapporté au nombre total de balises insérées) est inférieur à un certain seuil. Le nombre minimal d'exemples à couvrir et le seuil sont des paramètres de l'algorithme fournis par l'utilisateur. L'ensemble de motifs conservés est ordonné de façon décroissante par le nombre d'exemples couverts puis de façon croissante par le taux d'erreur ; seuls les k meilleurs motifs générés pour l'exemple en cours sont conservés. Ces k meilleurs motifs sont enregistrés dans la liste des règles d'annotation et appliqués sur le corpus ; tous les exemples couverts par ces motifs sont retirés de la liste des exemples à traiter, puis le processus est répété avec l'exemple suivant, jusqu'à ce qu'il n'y ait plus d'exemple à traiter. Ceci permet d'obtenir une liste de *meilleures règles*, les règles d'annotation ayant une forte précision (d'autant plus grande que le seuil défini pour le taux d'erreur est bas), mais une couverture assez faible. Pour augmenter la couverture, des *règles contextuelles* sont ajoutées à ces *meilleures règles*. Les *règles contextuelles* correspondent aux motifs n'ayant pas été retenus comme *meilleures règles* à cause d'un taux d'erreur trop important, mais que l'on peut rendre plus précis en contraignant leur application dans une certaine fenêtre autour d'une balise insérée dans l'étape précédente par les *meilleures règles*. En-

suite, les *règles de correction* sont générées de la même manière : elles tiennent compte de toutes les balises insérées par les *meilleures règles* et les *règles contextuelles*, mais elles servent à déplacer les balises mal placéees plutôt qu'à en insérer de nouvelles.

Un autre algorithme très utilisé en extraction d'information est l'algorithme BWI (Boosted Wrapper Induction) [Freitag & Kushmerick, 2000]. Là encore, les informations lexico-syntaxiques sont utilisées pour généraliser des motifs permettant la détection de débuts ou fins de chaînes de caractères à extraire (ces chaînes à extraire sont appelées champs). Une différence essentielle avec l'algorithme $(LP)^2$ est que les règles ne sont pas directement appliquées pour insérer des balises, mais pour calculer des scores de début et de fin de champs. Chaque position dans le document (i.e. chaque espace entre mots) se voit ainsi affecter un score de début de champ constitué par la somme des confiances de tous les motifs de détection de début qui s'appliquent à cette position ; de même chaque position se voit attribuer un score de fin de champ. Un histogramme des longueurs de chaînes de caractères à extraire est construit à partir des exemples : il permet de déterminer les probabilités $Pr(x)$, probabilité qu'un champ à extraire ait une longueur x. Un champ entre les positions i et j dans le document est extrait si son score d'extraction est supérieur à un seuil défini par l'utilisateur : le score d'extraction est une combinaison du score de début en position i, du score de fin en position j, et de la probabilité que le champ à extraire ait le même nombre de mots que la chaîne de caractères comprise entre i et j : $score_{extraction}(i,j) = score_{debut}(i) \times score_{fin}(j) \times Pr(j-i)$. L'autre différence de l'algorithme BWI par rapport à l'algorithme $(LP)^2$ est la façon dont les motifs de détection de début ou de fin de champ sont appris. En effet, $(LP)^2$ est un algorithme de couverture : seuls les motifs capables d'extraire les exemples non encore couverts sont recherchés, même si le calcul des meilleures règles se fait sur tout l'ensemble d'exemples. Dans BWI, tous les exemples sont pris en compte pour la génération de motifs de détection. Par contre, les exemples sont pondérés (on part d'un poids uniforme sur tous les exemples, et chaque fois qu'une règle est apprise les poids sont modifiés pour renforcer l'importance des exemples positifs non couverts ou négatifs extraits par erreur) : pour apprendre une nouvelle règle l'algorithme maximise la somme des poids des exemples positifs extraits et minimise celle des exemples négatifs. Cette technique de pondération permet d'augmenter la couverture de l'ensemble de règles en favorisant les règles complémentaires.

L'utilisation de ces techniques d'extraction d'information par apprentissage supervisé nécessite la constitution d'une importante base d'exemples annotés à la main. Dans le cadre de cette thèse, le parti pris est de limiter au maximum le temps d'expertise. En effet, les experts du domaine passent déjà énormément de temps à construire l'ontologie, qui est de toute façon nécessaire quelle que soit la méthode d'annotation sémantique adoptée par la suite. Le but est que le travail manuel reste concentré sur la construction de l'ontologie du domaine d'application, mais que la suite du processus ne nécessite pas de travail humain

mis à part pour la validation des résultats produits par le système. C'est pourquoi nous nous intéressons aux techniques d'extraction d'information non supervisées, s'appuyant uniquement sur une ontologie.

2.1.2 Algorithmes d'extraction d'information par apprentissage non supervisé

NOMEN [Yangarber et al., 2002] est un algorithme d'extraction d'information non supervisé destiné à l'enrichissement de lexique. Il s'agit d'un algorithme itératif d'apprentissage de motifs d'extraction. Il utilise en entrée uniquement le lexique initial et un ensemble de documents non annotés. Le lexique initial consiste en des listes de noms ou groupes nominaux (appelés termes), chaque liste ayant une "classe" associée : noms de gènes, noms de vaccins, noms de maladies, noms de protéines, etc. La première étape consiste à repérer l'ensemble des termes du lexique dans les documents et à les annoter : ces termes sont utilisés comme "graines" permettant la construction de règles d'extraction. Les règles d'extraction, comme pour les algorithmes $(LP)^2$ et BWI, sont des motifs permettant de repérer un début ou une fin de champ à extraire : ces motifs contiennent les n mots avant et après l'insertion de la balise (de début ou de fin de champ). Les motifs peuvent être généralisés en remplaçant un ou plusieurs des mots par un *joker* (n'importe quel mot est accepté dans la reconnaissance du motif). Chaque motif se voit associer la classe de la graine qui a permis de le générer.

Chaque motif (généré à partir d'un exemple d'annotation de "graine", soit avec les mots exacts soit avec une généralisation) est appliqué sur l'ensemble du corpus. A chaque fois qu'un motif est reconnu dans le document, le groupe nominal (obtenu par analyse syntaxique du texte) qui commence ou finit à l'emplacement reconnu, suivant qu'il s'agit d'un motif de reconnaissance de début ou de fin de champ, est examiné. Ce groupe nominal peut appartenir à trois catégories :

- *correct*, si le groupe nominal appartient au lexique avec la classe qui a été associée au motif ;
- *faux*, si le groupe nominal appartient au lexique mais dans une autre classe que celle définie pour le motif (par exemple le groupe nominal est un vaccin alors que le motif a été construit à partir d'une "graine" de type maladie) ;
- *inconnu*, si le groupe nominal n'appartient pas au lexique.

Tous les motifs pour lesquels le nombre de groupes nominaux distincts *corrects* détectés rapporté au nombre de groupes nominaux distincts connus détectés (*corrects* et *faux*) est en-dessous d'un certain seuil sont éliminés. Les motifs restants sont ordonnés selon leur confiance et le nombre de groupe nominaux *corrects* distincts détectés (avec pour confiance le nombre de groupes nominaux *corrects* distincts rapporté au nombre total de groupes nominaux distincts détectés – *corrects*, *faux* et *inconnus*). Les k meilleurs motifs sont retenus et appliqués à

l'ensemble des documents. Chaque groupe nominal reconnu (à un ou plusieurs endroits dans les documents) par ces motifs se voit affecter un score par classe. Soit C une classe (telle que maladie, vaccin, etc.) ; soit GN un groupe nominal reconnu par au moins un motif pour cette classe ; soit $Motifs_C^{GN}$ l'ensemble des motifs de classe C permettant de reconnaître GN à un ou plusieurs endroits dans l'ensemble de documents :

$$score(C, GN) = 1 - \prod_{m \in Motifs_C^{GN}} (1 - confiance(m)) \tag{2.1}$$

Les m meilleurs groupes nominaux suivant ce score sont retenus et ajoutés aux "graines" associées à la classe C. L'apprentissage de motifs est recommencé avec ces nouvelles graines, jusqu'à ce que l'application de l'algorithme ne permette pas de découvrir de nouvelles graines. Cet apprentissage n'est pas supervisé : les nouvelles graines ne sont pas validées au fur et à mesure de leur production, il est donc important que les nombres k et m ne soient pas trop grands, pour éviter qu'une fausse graine génère un grand nombre de motifs et que l'erreur augmente de façon exponentielle au fur et à mesure des itérations.

[Hearst, 1992] propose non pas l'apprentissage de motifs d'extraction à partir d'exemples, mais l'utilisation de motifs linguistiques indépendants du domaine pour l'extraction d'hyponymes. Ces motifs d'extraction permettent d'extraire les instances d'une classe ou les noms de sous-classes de cette classe. Un exemple d'un tel motif est $< GNClass >$ "*such as*" $< GNList >$, avec $< GNClass >$ un groupe nominal dont la tête est le nom de la classe dont les hyponymes sont recherchés, et $< GNList >$ une liste de groupes nominaux (séparés par des virgules, des "and" ou des "or"). Par exemple, la phrase « The characteristics of great cities such as Munich, Berlin, Nuremberg, Leipig, Wittenberg, Weimar, and Dresden are explored. » nous permet de déduire que Munich, Berlin, Nuremberg, Leipig, Wittenberg, Weimar et Dresden sont des villes.

Même lorsque l'apprentissage est non supervisé ou lorsque des règles d'extraction générales sont appliquées sans apprentissage, les techniques d'extraction d'information dans des textes en langue naturelle ne conviennent pas à nos objectifs d'annotation de tableaux issus de publications scientifiques. En effet, les techniques présentées utilisent le contexte linguistique dans lequel apparaissent les instances à annoter. Or les tableaux de données que nous souhaitons traiter ne présentent pas un tel contexte linguistique : chaque case de tableau contient non pas une phrase mais une instance d'un type de données. L'ordre de présentation des types de données appartenant à la signature d'une relation peut varier d'une publication à l'autre (des tableaux contenant les mêmes données peuvent être présentés avec des structures différentes).

2.2 Systèmes d'annotation pour des documents faiblement structurés

De nombreux systèmes d'annotation ont été mis au point dans les dernières années. Les systèmes d'annotation manuelle, qui disposent d'une interface graphique et / ou d'une architecture facilitant le travail manuel d'annotation, sont présentés en section 2.2.1. Les systèmes d'annotation semi-automatique utilisent des exemples ou des règles d'annotation définis manuellement par des experts pour induire de nouvelles règles d'annotation, en utilisant notamment les algorithmes d'extraction d'information présentés en section 2.1. L'expert est alors en général à nouveau impliqué dans la phase de validation des annotations générées. Ces systèmes d'annotation semi-automatique sont présentés en section 2.2.2. Enfin, il existe des systèmes d'annotation entièrement automatiques, qui ne demandent pas de données annotées en entrée de leur système : ils sont présentés en section 2.2.3.

2.2.1 Annotation manuelle

De nombreux systèmes ont été créés pour faciliter l'annotation manuelle de documents. Le système Annotea [Kahan et al., 2002] est un système permettant de partager des annotations, en langue naturelle, effectuées manuellement sur des documents en ligne : l'utilisateur sélectionne la portion de texte à annoter, précise quel est le type de l'annotation (commentaire, question, correction...), et rédige l'annotation. Les annotations sont stockées sur un serveur et sont restituées à chaque fois qu'un utilisateur lit le document en utilisant le système Annotea. Le système Annotea s'appuie sur Amaya [Guetari et al., 1998], qui est à la fois un navigateur web, un éditeur WYSIWYG pour HTML et XML, et un outil de publication sur des serveurs distants. Il s'agit d'un outil du W3C (World Wide web Consortium). Le navigateur Amaya a ensuite « absorbé » le résultat du projet Annotea : dans la littérature, les deux noms Amaya et Annotea sont utilisés indistinctement pour représenter le même système d'annotation partagée de documents.

Mangrove [McDowell et al., 2003] est un autre système permettant l'annotation de documents HTML. Un utilisateur souhaitant annoter un document se voit présenter l'ensemble des propriétés correspondant au type d'annotation qu'il a choisie (personne, événement...), ces propriétés pouvant être insérées à l'endroit souhaité dans le document HTML. Le système Mangrove dispose d'une base de données en RDF qui stocke toutes les annotations : dès publication d'une nouvelle annotation, tous les services sémantiques (i.e. applications qui utilisent les annotations pour produire de l'information) enregistrés dans le système sont notifiés ; l'utilisateur ayant produit l'annotation reçoit un *feedback* des différents services consommateurs de cette annotation, lui permettant à la fois de se rendre

compte de l'utilité de son annotation et éventuellement de la corriger si elle pose problème pour certains services (annotation ambiguë, détail manquant...). Ce *feedback* peut également jouer un rôle de découverte de services sémantiques qui n'étaient pas initialement ceux auxquels l'utilisateur pensait donner accès en produisant une annotation.

Le système CREAM (Creating RElational, Annotation-based Metadata) [Handschuh et al., 2001] est un système permettant l'annotation manuelle de documents à partir d'une ontologie. Dans son implémentation, Ont-O-Mat, l'utilisateur dispose, sur une même fenêtre, de deux vues, l'une permettant de visualiser le document à annoter, l'autre présentant l'ontologie à utiliser pour l'annotation : celle-ci comprend aussi bien les concepts que leurs instances, ce qui permet de marquer une portion de texte comme correpondant à une instance déjà rencontrée dans une autre document (par exemple, une personne). Les annotations sont enregistrées en RDF directement dans les documents ; elles sont traitées dans un serveur d'inférences après avoir été récupérées par un crawler RDF.

Ces systèmes d'annotation manuelle, bien qu'intéressants pour faciliter la production de documents utilisables dans le cadre du web sémantique, présentent un inconvénient majeur : la production d'annotations manuelles est très consommatrice en terme de temps pour les utilisateurs. C'est pourquoi la recherche se tourne de plus en plus vers des systèmes d'annotation semi-automatique, que nous présentons dans la section suivante.

2.2.2 Systèmes d'annotation semi-automatique

Le système CREAM a été enrichi d'outils pour la création semi-automatique d'annotations sémantiques, le système enrichi s'appelant S-CREAM (Semi-automatic CREAtion of Metadata) [Handschuh et al., 2002]. S-CREAM est composé du système CREAM pour l'annotation manuelle, les annotations manuelles étant reprises comme données d'entraînement dans le module Amilcare [Ciravegna, 2004], outil d'extraction d'information par apprentissage. Amilcare commence par appliquer un pré-traitement du texte en utilisant la chaîne de traitement ANNIE du système GATE [Cunningham et al., 2002] : segmentation du texte en mots et en phrases, reconnaissance de la forme grammaticale des mots (nom, verbe, etc.), reconnaissance d'entités nommées (noms de personnes ou d'organisations, dates, etc.). Amilcare applique ensuite l'algorithme $(LP)^2$ (présenté en section 2.1.1) pour l'apprentissage de règles d'extraction d'information. Les balises insérées par Amilcare permettent d'identifier des éléments, mais pas de les mettre en relation les uns avec les autres. Les relations entre éléments sont construites en définissant un type de balise référente, à laquelle se rattachent les autres éléments (par exemple, le type référent est <hotel> : les autres types, par exemple <ville> ou <prix>, sont rattachés à l'hôtel le plus proche ayant été identifié en amont dans le texte). Le système S-CREAM ne dispose d'aucun patron d'extraction de relations : les relations sont reconnues uniquement par

proximité géographique des différentes balises dans le texte.

Le système proposé par [Labský & Svátek, 2006] combine différentes approches d'extraction d'information. L'extraction repose à la fois sur la définition manuelle de règles d'extraction, et sur l'apprentissage de règles d'extraction à partir de documents annotés et /ou de listes de chaînes de caractères correspondant à un champ à extraire. Comme pour le système S-CREAM, l'extraction se fait à partir d'une classe de référence à laquelle se rattachent tous les autres éléments extraits.

Comme nous l'avons déjà expliqué en section 2.1.1, nous souhaitons éviter l'utilisation de techniques demandant la constitution d'une base de documents annotés manuellement en plus du travail manuel déjà important de construction de l'ontologie. De plus, le système S-CREAM comme le système proposé par [Labský & Svátek, 2006] supposent que l'on peut modéliser le domaine avec une classe de référence à laquelle tous les autres éléments extraits peuvent être rattachés, et qu'il existe un seul type de relation entre la classe de référence et chaque autre type d'élément extrait. Ce n'est pas le cas de notre application, dans laquelle tous les types de données ont la même importance. Dans notre travail, la co-occurrence de deux types de données dans un tableau n'est pas suffisante pour savoir quelle est la relation qui existe entre ces deux types (en effet, deux types de données peuvent se trouver associés dans des relations différentes).

2.2.3 Annotation non supervisée

Le système Armadillo [Norton et al., 2005] est un système d'annotation qui combine de nombreuses techniques d'annotation et d'extraction d'information sur le web. Nous classons ce système dans les systèmes d'annotation non supervisée, car le fonctionnement global de la chaîne d'annotation est prévu pour utiliser sans supervision les informations disponibles sur le web. La « non supervision » du système n'est en revanche pas totale : certains des services utilisés dans la chaîne de traitement d'Armadillo sont supervisés ou même construits avec des règles d'extraction définies manuellement.

Le but du système Armadillo est d'annoter des instances d'une classe cible, dont la sur-classe est facilement repérable mais sur laquelle il y a beaucoup d'ambiguïté, en utilisant les relations de ces instances avec des instances de classes sur lesquelles il y a peu d'ambiguïté. Le système Armadillo utilise une classe source, qui sert d'accroche pour trouver les instances de la classe cible, et des classes annexes, qui permettent de valider la classification des instances repérées. Par exemple, une classe cible est *Universitaire* : sa sur-classe *Personne* est facile à repérer car les techniques d'extraction de noms de personnes sont bien développées, mais cette classe est ambiguë car il est difficile de savoir si une personne est un universitaire ou non. Deux classes en relation avec la classe *Universitaire* sont donc utilisées :

- *Université* est la classe source, qui permet de repérer les instances de *Personne* travaillant dans une *Université* ;
- *Article* est la classe annexe. Elle permet de définir un *Universitaire* comme une *Personne* travaillant dans une *Université* et ayant écrit au moins un *Article* (ce qui permet d'éliminer les agents comptables, secrétaires et autres personnels des universités n'étant pas des universitaires).

L'idée dans Armadillo est d'utiliser tout d'abord des « oracles » (données considérées comme certaines), obtenus en examinant des listes d'instances de la classe source (par exemple ici, des listes d'universités). Ces instances sont alors recherchées sur l'ensemble du web et, dans chaque document référençant l'une de ces instances, les instances potentielles de la classe cible sont recherchées (ici, par extraction des noms de personnes). Ensuite, les relations entre les instances potentielles de la classe cible et les instances de la classe source et des classes annexes sont vérifiées, en utilisant la redondance de l'information sur le web :

- par raisonnement contextuel : plus il y a de documents présentant une co-occurrence de l'instance de la classe source et de l'instance de la classe cible, plus la relation entre l'instance de la classe source et de l'instance de la classe cible est probable ;
- par raisonnement relationnel : certaines techniques, dépendantes du domaine, permettent de vérifier la nature d'une relation entre une instance de la classe cible et une instance de la classe source ou d'une classe annexe. C'est par exemple le cas de l'utilisation de règles d'extraction sur le site DBLP, qui permet de trouver des instances d'*Article* et de *Personne* reliées par la relation *auteur*. D'autres techniques permettent d'apporter des indices supplémentaires sur la classification d'une instance de la sur-classe de la classe cible comme étant une instance de la classe cible. Par exemple, la recherche de titres académiques (Docteur, Professeur) apporte un indice supplémentaire sur la classification en tant qu'*Universitaire* d'une *Personne*.

Si les différents indices recueillis sont suffisamment nombreux et/ou sûrs pour reconnaître une instance de la classe cible, cette instance est alors annotée.

Le système Armadillo constitue un véritable progrès vers la réutilisation de services d'extraction d'information spécialisés qui deviennent des briques d'un système d'annotation de plus grande ampleur, et montre un effort vers la construction de systèmes d'annotation non supervisés. Le système Armadillo n'est en revanche pas du tout adapté à notre problème :

- le système Armadillo suppose que l'on dispose de techniques permettant de reconnaître des relations entre instances, et que le but est de pouvoir classifier une instance d'une sur-classe de la classe cible comme une instance de cette classe cible. Notre problématique est différente : non seulement nous cherchons à classifier les instances des différents types (dans le cadre de la reconnaissance du type des colonnes d'un tableau), mais nous cherchons également comment identifier les relations entre les différentes instances ;

– le système Armadillo utilise la redondance de l'information sur le web pour confirmer les relations entre instances. Or, l'un des problèmes majeurs que nous rencontrons dans notre domaine d'application est la rareté de l'information : l'utilisation de la redondance n'est donc pas envisageable pour notre application.

Le système KnowItAll [Etzioni et al., 2005] est construit à partir des motifs linguistiques indépendants du domaine introduits par [Hearst, 1992] et décrits en section 2.1.2. Ainsi, le nom d'une classe est donné en entrée du système et les motifs linguistiques permettent de les instances de cette classe en utilisant un moteur de recherche sur le web. Le système KnowItAll peut également appliquer ces motifs sur une base d'exemples, afin de déterminer la confiance que l'on peut avoir en chacun de ces motifs, et ne retenir une instance que si elle est extraite par plusieurs motifs linguistiques dont la combinaison permet d'avoir une confiance suffisante. Une fois les instances extraites à l'aide des motifs linguistiques indépendants du domaine, la couverture du système KnowItAll est améliorée en utilisant diverses techniques : apprentissage de motifs d'extraction spécifiques au domaine, extraction de sous-classes et extraction de listes. L'apprentissage de motifs d'extraction spécifiques au domaine se fait en repérant dans les documents les différentes occurrences des instances dont les noms ont été trouvés par les motifs linguistiques indépendants du domaine. Les 4 mots précédant et suivant l'occurrence d'une instance dans le texte sont appelés le contexte d'apparition d'une instance : les meilleurs motifs construits comme des sous-chaînes de ces contextes d'apparitions sont utilisés comme motifs d'extraction pour de nouvelles instances. L'extraction de sous-classes est faite en utilisant les motifs linguistiques d'hyponymie indépendants du domaine : l'application une première fois du motif "such as" sur l'expression « scientists, such as mathematicians, physicists and chemists » permet d'identifier *Mathematician*, *Physicist* et *Chemist* comme des sous-classes de la classe *Scientist*. Il devient alors possible de trouver que Leibniz et Lambert sont des *Scientist* grâce à la reconnaissance de la sous-classe *Mathematician* dans la phrase « Attempts to treat the operations of formal logic in a symbolic or algebraic way were made by some of the more philosophical mathematicians, such as Leibniz and Lambert ». Dans [Etzioni et al., 2005], la distinction entre sous-classe et instance n'est pas abordée : les domaines d'application choisis simplifient le problème puisque pour *Scientist*, *City* et *Film*, les instances sont des noms propres (ou du moins la première lettre est capitalisée dans le cas des films) et les sous-classes sont des noms communs. La dernière voie d'amélioration de la couverture du système KnowItAll est l'extraction de listes : en effet, lorsque différentes instances d'une classe ont été reconnues, il est possible de rechercher des documents dans lesquels plusieurs de ces instances apparaissent. Si ces différentes instances apparaissent sous forme de listes ou de tableaux, il est alors possible, par induction de motifs sur la structure du document, de découvrir d'autres instances de la même classe au sein de la liste ou du tableau.

Le système KnowItAll n'est pas adapté à notre problème d'annotation de

tableaux car, d'une part, il utilise le contexte linguistique d'apparition des instances dans des phrases (alors que dans les tableaux on ne dispose pas de contexte linguistique), et d'autre part, il utilise la redondance du web pour confirmer les instances hypothétiques (nous avons déjà signalé que dans notre domaine d'application il y avait très peu de redondance de l'information).

2.3 Annotation de tableaux

Il existe de nombreux travaux sur l'analyse de tableaux, tant pour détecter l'existence d'une structure tabulaire dans un texte que pour déterminer l'orientation d'un tableau [Zanibbi et al., 2004]. Cependant, les travaux d'annotation sémantique ou d'extraction d'information à partir de tableaux sont plus rares. La section 2.3.1 présente des techniques d'extraction d'information basées sur la structure du document, qui sont généralement appliquées à des tableaux. La section 2.3.2 présente les travaux qui s'intéressent aux tableaux de façon spécifique.

2.3.1 Extraction d'information dans des documents fortement structurés

L'extraction d'information est facilitée lorsqu'on travaille sur des documents fortement structurés, et plus exactement de structure homogène, tels que des pages web dynamiques. Dans ce cas, la structure de la page est fixe et les données présentées sont générées à partir d'une base de données. L'annotation ou l'extraction d'information sur de telles pages peut être faite à l'aide de *wrappers* basés sur la structure de la page, c'est-à-dire des règles d'annotation ou d'extraction qui utilisent des informations de mise en forme et/ou d'ordre qui sont constantes entre toutes les pages à annoter : première, dernière ligne d'un tableau, mise en gras ou en italique, champ contenant une destination à la suite d'un horaire (reconnaissable par sa forme caractéristique HH : mm par exemple), etc. Ces *wrappers* ou règles d'extraction sont spécifiques à un format de pages données (à un site web par exemple) et doivent être redéfinis pour chaque nouvel ensemble de pages dont la structure diffère des pages déjà traitées.

Les *wrappers* peuvent être définis à la main dans différents langages de programmation. Cependant, cela demande une importante charge de travail dès que le nombre de sites pour lesquels des *wrappers* sont définis devient grand. De plus, cela demande des connaissances en programmation pour un travail somme toute très répétitif. Il existe également des outils, tels que Lixto [Baumgartner et al., 2001], qui permettent de générer des *wrappers* de façon semi-automatique. Dans l'utilisation de Lixto, l'utilisateur sélectionne simplement dans la page les éléments à annoter, le système génère une règle d'extraction en utilisant une généralisation du chemin de balises HTML permettant d'accéder à l'élément à annoter, et l'utilisateur modifie cette règle pour élargir les critères

d'extraction (OU logique) ou les affiner (ET logique), avec à chaque étape un retour visuel permettant de voir, dans la page, quelles zones seront extraites par la règle d'extraction courante. La force de Lixto est de permettre plusieurs niveaux d'extraction : par exemple si une règle permet d'extraire des descriptions de vol, on pourra définir des règles d'extraction de la destination et de l'horaire à l'intérieur de "l'objet" vol. Ceci permet non seulement d'obtenir une extraction plus pertinente, mais surtout de conserver la notion de relation (au sens bases de données du terme).

La génération semi-automatique de *wrappers* pour l'extraction d'information dans des documents à structure homogène n'est malheureusement pas adaptée à nos besoins : elle nécessite en effet l'intervention d'un humain pour chaque nouvelle structure de présentation des données. Or les données contenues dans les tableaux de publications scientifiques le sont sous des formats extrêmement variables, il faudrait donc une intervention humaine pour chaque nouvelle publication !

D'autres travaux exploitent l'existence d'une structure, sans nécessiter l'homogénéité de structure d'un document à l'autre. C'est le cas du travail de [Tenier et al., 2006], qui propose d'exploiter des structures HTML arborescentes, telles que des tableaux (où c'est l'arborescence ligne-case qui est exploitée) pour reconnaître des relations binaires de type concept-rôle entre instances. Les structures de niveau inférieur (les cases) contiennent des chaînes de caractères identifiées comme des individus (instances de concepts : un nom, un e-mail...). Lorsque l'ontologie décrit l'existence d'un rôle (relation binaire dirigée) entre les concepts de deux instances dans les structures de niveau inférieur, alors ce rôle est instancié. La structure de niveau supérieur (la ligne) se voit alors attribuer une des instances du niveau inférieur si et seulement si cette instance appartient au domaine d'au moins l'un des rôles identifiés, et n'appartient au co-domaine d'aucun des rôles identifiés.

Gaëlle Hignette	hignette@agroparistech.fr	01 44 08 18 89
Patrice Buche	patrice.buche@agroparistech.fr	01 44 08 16 75
...	...	...

Tab. 2.1 – Exemple de tableau exploitable par la méthode d'annotation de [Tenier et al., 2006]

Exemple 2.1 *Prenons le tableau 2.3.1. Un premier travail permet d'identifier que* Gaëlle Hignette *et* Patrice Buche *sont des personnes, que* hignette@agroparistech.fr *et* patrice.buche@agroparistech.fr *sont des emails et que* 01 44 08 18 89 *et* 01 44 08 16 75 *sont des numéros de téléphone. Les rôles hasEmail(Gaëlle Hignette, hignette@agroparistech.fr) et hasPhone(Gaëlle Hignette,*

01 44 08 18 89) sont instanciés sur la première ligne : Gaëlle Hignette *appartenant au domaine des deux rôles identifiés et au co-domaine d'aucun, la ligne est alors annotée avec l'instance* Gaëlle Hignette *(de même pour la deuxième ligne annotée avec* Patrice Buche*).*

Cette technique d'annotation présente deux inconvénients majeurs par rapport à notre problématique :

- elle suppose que les chaînes de caractères représentées dans les cases du tableau ont été préalablement reconnues comme des instances de concepts : dans notre application, la reconnaissance du type de données représenté par une colonne est un problème à part entière, discuté dans le chapitre 4 ;
- elle ne prend en compte que les relations binaires, et suppose que deux concepts ne peuvent être reliés que par une seule relation. Dans l'ontologie que nous utilisons, les relations sont n-aires, et il peut exister plusieurs relations qui relient deux types de données.

2.3.2 Travaux spécifiques sur les tableaux

L'algorithme présenté dans [Pivk et al., 2004] permet de reconnaître les relations représentées par un tableau. L'accent est essentiellement mis sur la reconnaissance de l'organisation du tableau. L'algorithme calcule des distances entre cellules fondées sur le type de contenu des cellules (défini dans une hiérarchie de types : mot, nombre ou date, les nombres étant subdivisés en nombre entier ou décimal, les nombres décimaux étant compris entre 0 et 1 ou pas, etc.). L'orientation générale du tableau est déduite de ces distances : si les colonnes sont semblables deux à deux la lecture de chaque relation est verticale (i.e. une colonne correspond à un enregistrement, au sens bases de données du terme), sinon la lecture est horizontale. Le tableau est ensuite divisé en unités logiques et en régions. Les unités logiques sont déterminées par une rupture horizontale si les enregistrements sont horizontaux, verticale si les enregistrements sont verticaux (rupture entre en-tête et corps du tableau). Les régions sont des ensembles rectangulaires de cellules, contenues dans une même unité logique et ayant le même type fonctionnel : cellules indiquant la nature conceptuelle des éléments de la ligne ou colonne (type des attributs de la relation au sens base de données du terme) et cellules contenant des instances de ces concepts (la valeur des attributs dans une relation au sens base de données). En considérant qu'un tableau se lit soit de droite à gauche, soit de haut en bas, la case en haut à gauche du tableau est nécessairement un en-tête de colonne ou de ligne, i.e. un concept, et la case en bas à droite est nécessairement une donnée, i.e. une instance de concept. Les cellules contenant des chiffres, dates ou unités monétaires ont de plus grandes chances d'être des instances. Les instances sont reconnues car elles forment des blocs de cellules homogènes en bas à droite des unités logiques. Une fois le sens de lecture connu et la différence faite entre concepts et instances, il est

possible d'instancier des relations avec le contenu des cellules d'instance. Dans [Pivk et al., 2004], ce ne sont pas tout à fait des relations qui sont inférées mais des méthodes en F-Logic. Les paramètres de la méthode sont les constituants de la clef primaire (au sens base de données du terme) permettant d'obtenir le résultat de la méthode : ce sont les valeurs des cellules-instances en haut ou à gauche du tableau, dont la combinaison de valeurs ne permet d'obtenir qu'une seule ligne (ou colonne selon le sens de lecture du tableau). Les résultats de la méthode sont constitués par les valeurs des cellules-instances en bas ou à droite dans la ligne ou colonne, qui ne sont pas des clefs. Les noms des attributs de la méthode sont trouvés en recherchant dans WordNet [Fellbaum, 1998] l'hyperonyme (i.e. plus spécifique généralisant commun) de toutes les valeurs d'attribut. Il n'y a pas de recherche de termes similaires mais une simple recherche par égalité : cela suppose que toutes les valeurs d'attribut peuvent être trouvées dans WordNet. Le nom de la méthode est alors donné par la concaténation des hyperonymes des résultats de cette méthode. Ainsi, cet algorithme permet une sémantisation des tableaux sous forme de méthodes en F-Logic. Cependant, cela permet de créer des méthodes et non de reconnaître des méthodes existantes. Dans le cadre de ce travail de thèse, cette technique n'est donc pas appropriée : en effet, nous cherchons à reconnaître des relations prédéfinies dans l'ontologie pour pouvoir ensuite les interroger via cette ontologie. De plus, la méthode de recherche d'hyperonyme dans WordNet est efficace si le domaine traité utilise un vocabulaire de la vie courante, mais lorsque les termes utilisés dans le domaine d'application sont très spécifiques, comme c'est la cas pour notre application en microbiologie alimentaire, les termes du domaine ne se retrouvent pas dans WordNet : il nous faut alors trouver d'autres techniques pour déduire le type des colonnes d'un tableau.

La méthodologie d'extraction d'information à partir de tableaux développée par [Embley et al., 2002] fait appel à la définition d'une *ontologie d'extraction*. L'ontologie d'extraction présente, comme pour une ontologie de domaine, les concepts (objets de références n'ayant pas de représentation textuelle, comme par exemple une voiture) et leurs attributs (segments de texte, nombres ou images pouvant se rattacher au concept, tels que la marque de la voiture ou l'année de construction), ainsi que les valeurs possibles pour ces attributs (définition d'un attribut par son *"set of objects"*, par exemple une liste de marques de voitures). L'ontologie d'extraction présente également, en plus d'une ontologie de domaine, des règles d'extraction et/ou de transformation pour les différents attributs : les règles d'extraction définissent soit un *contexte* d'apparition des valeurs pour l'attribut, soit des *mots clefs* pouvant apparaître comme titre de colonne pour cet attribut ; les règles de transformation permettent de transformer la valeur extraite en la forme de stockage voulue (par exemple, ajout de "19" devant une année en 2 chiffres). La reconnaissance d'un certain nombre d'attributs (d'après les valeurs prédéfinies) dans une table permet l'induction de règles de transformation qui permettent de transformer la table dans le schéma-cible de l'extraction d'information : l'utilisation de ces règles de transformation permet, dans une certaine

mesure, l'extraction d'attributs dont la valeur n'avait pas été initialement définie dans le *"sets of objects"* de l'ontologie (par exemple, lorsque toutes les valeurs d'une colonne sauf une ont été reconnues). Cette technique d'extraction d'information à partir de tableaux est cependant dépendante de la bonne définition des règles d'extraction : elle ressemble en cela à une définition manuelle de wrappers, même si elle gagne en robustesse de par le fait que les règles d'extraction sont définies indépendamment de la structure des documents à analyser. Cette méthode ne convient pas à notre problématique, puisque nous ne souhaitons pas avoir à définir manuellement des règles d'extraction pour les différents types de données de notre ontologie (de telles règles, outre le fait qu'elles demandent beaucoup de temps d'expert pour leur mise en place, sont difficiles à appliquer à notre domaine où le vocabulaire est très variable et où la même unité peut être employée pour différents types numériques).

Dans le système TANGO (Table ANalysis for Generating Ontologies) [Tijerino et al., 2005], la méthode de construction d'ontologies à partir de tableaux est divisée en 4 phases :

1. reconnaissance des informations d'une table et transformation de la table pour la mettre sous forme canonique ;
2. construction de mini-ontologies à partir des tables canoniques ;
3. découverte de mappings entre les différentes mini-ontologies provenant de différentes tables ;
4. fusion des différentes mini-ontologies pour créer une ontologie de domaine plus conséquente.

Les étapes 1 et 2 de cette méthode s'apparentent à notre problématique d'annotation de tableaux. Cependant, dans ce travail la reconnaissance du type des données représentées par le tableau se fait en utilisant des "data frames", qui définissent des formes morphologiques et contextes d'apparition de certains types de données (longitude, latitude, poids, pourcentage...), voire des listes complètes des valeurs possibles (pays, ville...). Si ces data frames ne permettent pas la reconnaissance du type des données, un hyperonyme de toutes les données présentées sur un même plan structurel (donc susceptibles de former une colonne dans la forme canonique de la table) est recherché dans WordNet [Fellbaum, 1998]. Ces techniques ne sont pas utilisables pour notre problème car :

– nous ne disposons pas de "data frames" nous permettant de reconnaître le type des données. En effet, pour les types numériques, nous connaissons les unités possibles mais la présence de ces unités ne nous permet pas de trancher pour un type numérique particulier (plusieurs types numériques s'expriment avec les mêmes unités). De même, pour les types symboliques, nous définissons la hiérarchie du type : ce sont les valeurs que peut prendre le type *dans notre ontologie* mais ne préjuge pas du fait que ce soient les seules valeurs utilisées dans les documents (en effet, les tableaux à analyser présentent une très grande hétérogénéité de vocabulaire).

– nous ne pouvons pas trouver d'hyperonyme aux informations présentées dans la table grâce à WordNet : en effet, lorsque le domaine d'application utilise un vocabulaire trop précis, la plupart des termes utilisés dans les tableaux à annoter ne sont pas présents dans WordNet.

2.4 Premiers travaux sur l'annotation de tableaux dans le cadre du projet e.dot

Les premiers travaux d'annotation de tableaux pour la construction de l'entrepôt de données XML dans le cadre du projet e.dot ont été réalisés par Fatiha Saïs lors de son stage de DEA [Saïs, 2004, Saïs et al., 2005]. Ses idées ont été implémentées dans un logiciel appelé AQWEB, et les annotations générées ont été exploitées dans une première version du moteur d'interrogation élargie MIEL++. La méthode utilisée pour comparer les termes trouvés dans le tableau avec les termes de l'ontologie est présentée en section 2.4.1. La méthode utilisée pour la reconnaissance des relations représentées par le tableau est présentée en section 2.4.2. Nous présentons ensuite le score mis en place pour l'interrogation des tableaux annotés en section 2.4.3. Les limites de cette approche, qui ont été le point de départ de ce travail de thèse, sont discutées en section 2.4.4.

2.4.1 Intersection et inclusion de mots

Afin d'annoter les termes des tableaux issus du web avec des termes de l'ontologie, la solution adoptée consiste en une comparaison de mots. Les termes issus du web et ceux de l'ontologie sont représentés sous forme d'ensembles de mots, d'où sont exclus les mots dits "vides". Les mots vides sont définis dans une liste, fixe pour la langue de travail choisie (au choix : français ou anglais) ; il s'agit de mots trop fréquents et non porteurs de sens, comme les articles, conjonctions ou prépositions. Les mots conservés sont ensuite lemmatisés, opération qui permet de s'abstraire des modifications d'ordre grammatical (pluriel, conjugaison). Les ensembles de mots lemmatisés représentant chacun des termes de l'ontologie peuvent ensuite être comparés avec les ensembles représentant les termes issus du web.

La procédure de comparaison est de plus en plus dégradée, tant qu'aucune correspondance n'a été trouvée :

– tout d'abord, une égalité des ensembles de mots est recherchée : par exemple, le terme *carotte râpée* du web est égal au terme *carottes râpées* de l'ontologie, une fois la lemmatisation effectuée ;
– s'il n'a pas été possible de trouver une égalité, l'inclusion d'un ensemble de mots dans l'autre est recherchée. En effet, un sur-ensemble de mots ajoute de l'information, mais a de fortes chances de représenter un aliment proche de l'ensemble de mots de départ : par exemple *jambon cuit supérieur* est

un sur-ensemble de *jambon*. Les tests d'inclusion sont effectués dans les deux sens (terme du web inclus dans le terme de l'ontologie ou terme de l'ontologie inclus dans le terme du web) ;
– si aucune inclusion n'a été trouvée, une intersection non vide entre l'ensemble de mots du terme du web et l'ensemble de mots d'un terme de l'ontologie est recherchée. Cette technique permet de trouver des rapprochements supplémentaires, par exemple *ragoût de bœuf* trouvé sur le web peut être rapproché de *viande de bœuf* ou alors de *bœuf en sauce*. Par contre il y a plus de risques de rapprochements erronés, par exemple *saucisson sec* avec *fruits secs* ;
– si aucun des mots du terme du web n'apparaît dans un terme de l'ontologie, l'annotation du terme du web est abandonnée.

2.4.2 Reconnaissance des relations représentées par le tableau

Afin de reconnaître les relations représentées par un tableau, la première étape consiste en la reconnaissance des types des colonnes. Au moment où ce travail a été réalisé, les types symboliques de l'ontologie n'étaient pas bien définis en tant que tels. Ainsi, le type d'une colonne pouvait être n'importe quel terme d'une taxonomie (les types numériques étant représentés dans une taxonomie "facteurs expérimentaux", où les noms des différents types étaient reliés à la racine). Un terme est candidat pour être le type d'une colonne s'il subsume plus d'une certaine proportion (définie par l'utilisateur) des termes présents dans la colonne. Parmi tous les termes candidats, on ne conserve que ceux qui sont les plus spécifiques en subsumant le plus de valeurs : soit TC l'ensemble des termes candidats pour être type de la colonne C, soit $desc_C(t)$ la fonction qui associe à un terme t le nombre de termes de la colonne C subsumés par t, alors un terme $t \in TC$ est retenu si $\nexists\ t', t' \in TC\ et\ desc_C(t') > desc_C(t)$ et $\nexists\ t'', t'' \in TC\ et\ desc_C(t') = desc_C(t)\ et\ t'' \prec t$. Si jamais plusieurs termes sont retenus, celui choisi pour le type de la colonne est le premier trouvé. Si aucun type n'a été trouvé pour la colonne de cette manière, alors un terme est choisi comme type de la colonne s'il est égal au titre de la colonne (notamment, un type numérique n'est reconnu que si son nom est égal au titre de la colonne). La colonne est considérée comme de type inconnu si l'on n'a pas réussi à lui attribuer de type de cette manière.

Une relation est reconnue dans le tableau dès qu'au moins deux des types composant sa signature sont reconnus comme étant des types de colonne du tableau (à ce moment-là, la signature d'une relation n'était pas encore subdivisée en type résultat et types d'accès).

2.4.3 Un score pour l'interrogation

Dans le cadre du moteur d'interrogation MIEL++ sont ramenées à l'utilisateur des réponses correspondant à sa requête, ordonnées selon leur adéquation à la requête. Dans un premier temps, il faut donc repérer, pour un terme de l'ontologie utilisé pour l'interrogation, si les réponses sont plus ou moins sûres (i.e. si le terme du web, décrivant la donnée, est plus ou moins ressemblant avec le terme utilisé pour l'interrogation) : une égalité est sûre, une inclusion moins sûre et les intersections ne sont présentées qu'en dernier puisque ce sont les plus susceptibles de contenir des erreurs.

Dans un premier prototype de MIEL++, les résultats d'une requête sur un terme de l'ontologie sont ordonnés en utilisant un score de ressemblance entre les termes du web et le terme de l'ontologie. Soit w le terme du web et o un terme de l'ontologie utilisé pour annoter w, alors la ressemblance $r(w, o)$ entre le terme du web et le terme de l'ontologie est donnée par :

- si le terme o de l'ontologie présente une égalité de mots avec le terme w du web, alors $r(w, o) = 1$;
- si l'ensemble des mots de o est inclus dans l'ensemble des mots de w, alors $r(w, o) = \frac{2 \times min(nbWords(w), nbWords(o))}{nbWords(w) + nbWords(o)}$, avec $nbWords$ la fonction qui associe à un terme le nombre de mots qu'il contient ;
- s'il y a une intersection entre l'ensemble des mots de o et l'ensemble des mots de w, soit $indWordsIntersection$ le nombre de mots contenus dans cette intersection, alors $r(w, o) = \frac{2 \times indWordsIntersection}{nbWords(w) + nbWords(o)}$.

Produit	pH
Fromage de chèvre	6,6
Oignon rouge	5,2
Chou rouge	4,8

Tab. 2.2 – pH des produits utilisés

Exemple 2.2 *Considérons le tableau 2.2, annoté avec la version française de l'ontologie. La relation* AlimentPH *a été reconnue sur chaque ligne. Le terme « Fromage de chèvre » a été annoté, par inclusion de mots, avec le terme de l'ontologie "fromage". Le terme « Oignon rouge » a été annoté, par intersection de mots, avec les termes de l'ontologie "oignon d'Egypte", "oignon de printemps" et "chou rouge". Le terme « Chou rouge » a été annoté, par égalité de mots, avec le terme de l'ontologie "chou rouge".*

Si l'on veut interroger sur le terme de l'ontologie chou rouge*, on a deux relations* AlimentPH *qui correspondent dans ce tableau : l'une pour le terme du web* Oignon rouge*(avec une reconnaissance par intersection), l'autre pour le terme du web* Chou rouge *(avec une reconnaissance par égalité). On obtient les scores de ressemblance suivants :*

– $r(Oignon\ rouge, chou\ rouge) = \frac{2\times1}{2+2} = 0.5$;
– $r(Chou\ rouge, chou\ rouge) = 1$.

Remarque 2.1 *Le calcul est effectué en fonction de la façon dont est réalisée l'annotation, c'est pourquoi on distingue les différentes manières de trouver le rapprochement : égalité, inclusion ou intersection d'ensembles. Cependant, tous les calculs peuvent se ramener à la même formule :*

Soient $W = \{w_1, ..., w_n\}$ *et* $O = \{o_1, ..., o_k\}$ *les ensembles de mots lemmatisés représentant respectivement le terme* w *du web et le terme* o *de l'ontologie. Alors leur score de ressemblance est :*

$$r(w, o) = \frac{2\times \mid W \cap O \mid}{\mid W \mid + \mid O \mid} \tag{2.2}$$

2.4.4 Limites de l'approche

La ressemblance entre deux termes est évaluée uniquement sur la base de la proportion de mots identiques. Avec cette méthode, *Oignon rouge* sera tout aussi proche de *chou rouge* que de *oignon de printemps* : dans les deux cas il y a intersection sur un mot, chaque terme comportant 2 mots non vides. De plus, cette ressemblance est évaluée au moment de l'interrogation, pour connaître l'adéquation d'un terme du web par rapport au terme de l'ontologie utilisé pour la requête. Au niveau de ce qui est stocké dans l'entrepôt de données XML, chaque terme du web est associé à un ensemble de termes de l'ontologie possibles, sans notion d'ordre de pertinence des différents rapprochements.

Le travail présenté dans ce document s'intéresse à la possibilité de représenter, directement au niveau de l'annotation, un ordre de pertinence entre les différents termes de l'ontologie proposés pour annoter un terme du web. Cette pertinence est évaluée en calculant des similarités entre le terme du web, trouvé dans le tableau, et les différents termes de l'ontologie : le chapitre 3 présente de telles mesures de similarité.

Le travail que nous présentons dans ce mémoire présente en outre, par rapport au travail de [Saïs, 2004, Saïs et al., 2005], les améliorations suivantes :

– nous proposons une manière plus fine de gérer les types de colonnes, notamment en distinguant les colonnes numériques des colonnes symboliques ;
– nous gérons des mesures de similarités entre termes du web et termes de l'ontologie pour les données symboliques, mais nous gérons également des données imprécises pour les données numériques ;
– à chacune des étapes de l'annotation, nous combinons différentes sources d'information (titre et contenu des colonnes, titre et colonnes du tableau) afin de rendre nos résultats plus robustes.

Conclusion du chapitre

Nous avons présenté dans ce chapitre différentes méthodes d'annotation de documents ou d'extraction d'information. La construction d'une ontologie, en collaboration avec des experts du domaine d'application, est une étape indispensable pour la mise en place d'un système d'annotations sémantique. Il s'agit malheureusement d'un processus qui demande beaucoup de temps. La réduction du temps d'expertise nécessaire au réglage des systèmes d'annotation nous paraît un enjeu majeur pour l'adoption de ces systèmes dans différents domaines d'application. Aussi, nous avons choisi de construire une méthode d'annotation automatique de tableaux qui ne fasse appel qu'aux connaissances du domaine décrites dans l'ontologie. Cela évite de consommer du temps d'expert supplémentaire, comme cela est le cas dans les systèmes d'annotation supervisés qui demandent, en plus de l'ontologie, la construction d'un corpus de documents annotés manuellement.

Nous avons cherché à comparer notre travail avec des méthodes non supervisées d'annotation de tableaux. Les systèmes existants partent du principe qu'il est possible de trouver le type d'une colonne, soit par recherche d'un hyperonyme de l'ensemble des valeurs de la colonne dans WordNet, soit par définition de règles dans des « data frames ». Ce principe n'est pas vérifié quand le domaine d'application utilise un vocabulaire trop précis pour figurer dans WordNet ou trop variable d'un document à l'autre pour pouvoir être entièrement recensé dans des « data frames ». Dans ce travail de thèse, nous adressons non seulement le problème de la reconnaissance des relations représentées par un tableau, mais également les problèmes amont de la reconnaissance des types de données pour chaque colonne et de la signification des termes dans chaque cellule d'un tableau.

Les chapitres suivants sont consacrés à la présentation de notre méthode d'annotation de tableaux à l'aide d'une ontologie. Nous utilisons une approche d'annotation par agrégation d'éléments de données de plus en plus complexes. L'élément de données au niveau de granularité le plus fin est une cellule du tableau : ainsi, nous commençons par déterminer, pour chacune des cellules d'une colonne, si elle contient des données symbolique ou numériques, et nous agrégeons les résultats pour déterminer si une colonne est numérique ou symbolique. Dans le cas où nous travaillons sur une colonne symbolique, nous commençons par annoter le contenu des cellules : les annotations de toutes les cellules d'une colonne symbolique du tableau sont agrégées pour en déduire le type de cette colonne. Pour les colonnes numériques, on analyse directement la colonne dans son ensemble car les unités, qui sont les éléments majeurs permettant de déterminer le type de la colonne, ne sont pas nécessairement répétées dans toutes les cellules de la colonne. Une fois que chaque colonne (numérique ou symbolique) du tableau a été annotée avec le type correpondant dans l'ontologie, on utilise ces annotations pour en déduire quelles sont les relations représentées par le tableau. Une fois le niveau le plus complexe atteint, on revient au niveau le plus fin, pour instancier les relations, sur chaque ligne du tableau, avec les valeurs contenues dans les cellules du tableau.

Chapitre 3

Annotation des cellules d'un tableau

Ce chapitre s'intéresse uniquement à l'annotation de données symboliques. La façon de déterminer si une colonne contient des données symboliques ou numériques sera présentée au chapitre 4 (plus exactement en section 4.1).

Nous disposons d'une ontologie dans laquelle sont définies des hiérarchies de termes permettant de représenter les données symboliques. Notre but ici est de déterminer quels sont les termes de l'ontologie qui sont les plus proches des termes présentés à l'intérieur des cellules symboliques d'un tableau extrait d'un document trouvé sur le web. Pour cela, nous proposons de calculer des mesures de similarités entre les termes du web et les termes de l'ontologie. Nous avons choisi d'utiliser une comparaison entre termes basée sur des égalités de mots. Nous présentons cette approche en section 3.1. Il existe cependant des mesures de similarité entre termes qui utilisent d'autres méthodes que la comparaison mot à mot : ces mesures de similarité sont présentées en section 3.2. Enfin, il ne faut pas oublier que notre ontologie ne consiste pas en une simple liste de termes : les termes sont organisés entre eux selon une hiérarchie de subsomption. Nous décrivons nos tentatives pour exploiter cette hiérarchie dans la section 3.3.

3.1 Mesures de similarité lexicale par égalité de mots

Nous nous appuyons ici sur le travail de [Saïs, 2004] qui utilise une comparaison mot à mot pour trouver des correspondances entre les termes du web et ceux de l'ontologie : nous améliorons cette technique de comparaison en affectant des poids aux différents mots d'un terme, en fonction de leur importance dans la signification du terme. Les sections 3.1.1 et 3.1.2 présentent la façon dont sont pondérés les différents mots des termes à comparer. Les sections 3.1.3 et 3.1.4 présentent différentes mesures de similarité entre termes utilisant ces

pondérations. Une évaluation expérimentale de l'utilité de ces mesures de similarité dans notre domaine d'application est donnée en section 3.1.5.

3.1.1 Poids des mots dans un terme de l'ontologie

Comme présenté en section 2.4.4, lorsque l'on veut comparer un terme du web avec des termes de l'ontologie, le nombre de mots en commun n'est pas toujours un indicateur suffisant. En effet, il faut tenir compte de l'importance qu'ont ces mots au niveau de la signification des termes. Plus un mot sera porteur de sens, plus le fait que ce mot soit commun au terme du web et au terme de l'ontologie sera un bon indicateur de proximité sémantique entre ces deux termes. Pour tenir compte de cela, il faut pouvoir identifier dans les termes à comparer quels sont les mots porteurs de sens.

Les ontologies sont des vocabulaires structurés destinés à représenter des connaissances sur un domaine. Il paraît donc judicieux d'utiliser l'ontologie pour ajouter de l'information supplémentaire, à savoir l'importance des différents mots dans la signification des termes utilisés dans l'ontologie. En effet, ceci est une connaissance spécifique au domaine : par exemple, selon qu'on s'intéresse à des procédés de l'industrie alimentaire ou à des valeurs énergétiques d'aliments, le mot *"hâché"* dans *poulet hâché* n'aura pas du tout la même importance.

Chaque terme de l'ontologie est représenté par un ensemble de mots lemmatisés, comme présenté en section 2.4.1. Chacun de ces mots se voit associer un poids, compris entre 0 et 1, représentatif de son importance dans la signification du terme. Un poids de 0 correspond à un mot qui n'apporte aucun sens (en pratique, ce poids est réservé aux mots de la *stopword list*). Un poids de 1 correspond à un mot très important dans la signification du terme. Tout terme de l'ontologie doit avoir au moins un mot de poids 1, mais il peut y avoir plusieurs mots de poids 1 dans un terme s'ils sont tout aussi indispensables à la définition du terme. Les poids intermédiaires permettent d'identifier les mots modificateurs de sens, mais non essentiels dans la signification du terme.

Exemple 3.1 *On peut par exemple attribuer les poids suivants, en fonction de l'importance des mots dans la signification des termes :*

- chou rouge *donne* $\{chou : 1; rouge : 0.2\}$ *;*
- oignon de printemps *donne* $\{oignon : 1; printemps : 0.2\}$ *;*
- fruits et légumes *donne* $\{fruit : 1; legume : 1\}$ *;*
- produit laitier *donne* $\{produit : 0.2; laitier : 1\}$.

3.1.2 Poids des mots dans un terme du web

Il n'est pas possible de faire appel au jugement d'experts pour décider de l'importance des mots dans les termes du web, puisqu'il s'agit d'une annotation

automatique. Nous avons envisagé différentes techniques pour assigner automatiquement des poids aux mots des termes issus du web.

Une première technique consiste à tenir compte de la classe grammaticale du mot : les noms auraient un poids fort, les adjectifs un poids faible. Cependant, l'un des problèmes majeurs posé par cette technique est que les analyseurs syntaxiques sont des outils prévus pour être utilisés sur des phrases entières et non sur des termes isolés comme ceux trouvés dans les cases des tableaux (en général, ces termes sont des groupes nominaux). Par exemple, en utilisant l'analyseur syntaxique FrAG (French Annotation Grammar) [Bick, 2004, Kjærsgaard et al., 2007], l'analyse pour le terme *produit laitier* donne le résultat suivant : < *produit : verbe, laitier : adjectif* >. Si par contre le terme est placé dans une phrase (par exemple *"je consomme un produit laitier"*), le mot *produit* est bien reconnu comme un nom. En anglais également, en utilisant l'analyseur ANNIE inclus dans GATE (General Architecture for Text Engineering) [Cunningham et al., 2002, Cunningham et al., 2005], dont l'algorithme d'analyse syntaxique est décrit dans [Hepple, 2000], les résultats sont assez décevants : dans *split pea* (pois cassé), le mot *split* est reconnu comme un nom alors que c'est un adjectif, de même que *smoked* dans *smoked salmon* (saumon fumé) est reconnu comme un verbe conjugué au passé et non un adjectif. En outre, de nombreux contre-exemples trouvés dans l'ontologie (même si nous n'avons pas fait d'étude chiffrée sur ce point) nous montrent que des poids basés sur la nature grammaticale des mots ne représentent sans doute pas une solution idéale : par exemple, dans *produit laitier*, c'est l'adjectif *laitier* qui est porteur de sens.

Voyant que la recherche de la classe grammaticale des mots ne donnait pas de résultats encourageants, nous avons testé une technique plus simple qui utilise l'ordre des mots dans le groupe nominal. En effet, en anglais dans un groupe nominal, le nom est en dernière position, que les mots à rôle d'adjectifs utilisés soient des noms (par exemple dans *cherry tomato*, tomate cerise), de véritables adjectifs (comme dans *fresh cheese*, fromage frais) ou des participes passés (comme dans *split pea*, pois cassé). Nous avons donc essayé, pour des termes en anglais, de donner un poids de 1 au dernier mot du terme, et un poids de 0.5 aux mots précédents. Cependant des expérimentations ont permis de montrer que les annotations faites en utilisant ces poids étaient moins bonnes qu'en donnant simplement à chaque mot dans les termes du web un poids égal à 1, c'est-à-dire quand tous les mots sont considérés comme étant aussi importants (les résultats obtenus en donnant à chaque mot dans les termes du web un poids de 1 sont présentés en section 3.1.5). Cela nous permet d'affirmer qu'une simple analyse de l'ordre des mots dans un groupe nominal ne permet pas de déterminer quel est le mot le plus porteur de sens ; au contraire, comme nous le montrerons en section 3.1.5, les poids déterminés manuellement pour les mots des termes de l'ontologie permettent une amélioration de la qualité de l'annotation car ils sont réellement fondés sur la signification du terme. Dans la suite de notre travail, nous conservons donc les poids déterminés manuellement sur les mots des termes de l'ontologie, et des

poids de 1 sur les mots des termes du web.

3.1.3 Mesure de similarité proposée

Dans la continuité du travail de [Saïs, 2004], nous avons proposé de calculer un degré de similarité entre termes, basé non pas sur le nombre de mots en commun entre deux termes, mais sur l'importance de ces mots en commun dans chacun des deux termes.

Définition 3.1 *Soient* $W = \{w_1 : p_{w_1}; ...; w_n : p_{w_n}\}$ *et* $O = \{o_1 : p_{o_1}, ..., o_k : p_{o_k}\}$ *les ensembles de mots lemmatisés représentant respectivement le terme* w *du web et le terme* o *de l'ontologie avec les poids associés à chaque mot. Soit* C *l'ensemble de paires d'indices* (i, j) *tels que* $w_i = o_j$*. Alors le degré de similarité entre* w *et* o *est :*

$$sim(w, o) = \frac{\sum_{(i,j)\in C}(p_{w_i} + p_{o_j})}{\sum_{m=1}^{n} p_{w_m} + \sum_{m=1}^{k} p_{o_m}} \tag{3.1}$$

Remarque 3.1 *Lorsque les poids de tous les mots sont égaux à* 1 *(c'est-à-dire lorsqu'il n'y a pas de distinction entre les mots importants et les mots moins importants), on retrouve le score présenté en section 2.4.3. Il s'agit en fait du coefficient de similarité de Jaccard [Jaccard, 1912].*

Exemple 3.2 *Prenons le terme du web* Oignon rouge $\{oignon : 1; rouge : 1\}$ *et ses annotations possibles* oignon de printemps $\{oignon : 1; printemps : 0.2\}$ *et* chou rouge$\{chou : 1; rouge : 0.2\}$*. Les degrés de similarité obtenus sont les suivants :*

- $sim(Oignon\ rouge, oignon\ de\ printemps) = \frac{1+1}{2+1.2} = 0.625$ *;*
- $sim(Oignon\ rouge, chou\ rouge) = \frac{1+0.2}{2+1.2} = 0.375$*.*

Ce calcul nous permet de faire figurer dans notre annotation le fait que oignon rouge *est plus proche de* oignon de printemps *que de* chou rouge*. Par comparaison, lorsqu'on utilise le score de ressemblance présenté en section 2.4.1, les deux scores de ressemblance (entre* oignon rouge *et* oignon de printemps *et entre* oignon rouge *et* chou rouge*) sont égaux à* 0.5 *(cf. l'exemple 2.2).*

3.1.4 Mesures classiques de similarité entre vecteurs pondérés

Les termes de l'ontologie et ceux du web ont été jusqu'à présent représentés comme des ensembles de mots lemmatisés, chaque mot dans un terme ayant un poids associé compris entre 0 et 1. Ces termes peuvent également être représentés sous forme de vecteurs pondérés : il y a autant de coordonnées que de mots lemmatisés possibles (tous les mots contenus dans les termes de l'ontologie et dans le terme du web à annoter). Chaque coordonnée a pour valeur le poids,

dans le terme considéré, du mot correspondant à cette coordonnée : 0 si le mot n'appartient pas au terme, le poids du mot dans le terme sinon.

Exemple 3.3 *Le tableau 3.1 montre les coordonnées utilisées pour représenter sous forme de vecteurs pondérés les termes* Oignon rouge *(provenant du web) et* chou rouge *et* oignon de printemps *(provenant de l'ontologie).*

coordonnées / termes	oignon	rouge	chou	printemps
Oignon rouge	1	1	0	0
chou rouge	0	0.2	1	0
oignon de printemps	1	0	0	0.2

TAB. 3.1 – Représentation de termes sous forme de vecteurs pondérés

Avec une telle représentation, il est possible d'appliquer l'une des nombreuses mesures de similarité entre deux vecteurs. Nous étudions ici deux mesures parmi les plus utilisées en recherche d'information : le coefficient de Dice [Lin, 1998] et la mesure cosinus [Van Rijsbergen, 1979].

Soient w un terme du web et o un terme de l'ontologie, représentés sous forme de vecteurs $\vec{w} = (w_1, ..., w_n)$ et $\vec{o} = (o_1, ..., o_n)$. Leur coefficient de similarité de Dice $dice(w, o)$ et leur mesure de similarité par cosinus $cos(w, o)$ sont respectivement :

$$dice(w, o) = \frac{2 \times \sum_{i=1}^{n} w_i o_i}{\sum_{i=1}^{n} w_i^2 + \sum_{i=1}^{n} o_i^2} \tag{3.2}$$

$$cos(w, o) = \frac{\sum_{i=1}^{n} w_i o_i}{\sqrt{\sum_{i=1}^{n} w_i^2 \times \sum_{i=1}^{n} o_i^2}} \tag{3.3}$$

Exemple 3.4 *Reprenons l'exemple du terme du web* Oignon rouge *et ses annotations possibles* oignon de printemps *et* chou rouge *(représentation vectorielle présentée dans le tableau 3.1). Les degrés de similarité obtenus sont présentés dans le tableau 3.2.*

Que la mesure choisie soit le degré de similarité que nous avons proposé, le coefficient de Dice ou la mesure cosinus, les valeurs sont différentes mais l'ordre obtenu, indicatif de pertinence, est le même : le terme de l'ontologie oignon de printemps *est mieux adapté pour représenter le terme du web* Oignon rouge *que ne l'est le terme* chou rouge.

3.1.5 Evaluation expérimentale

L'utilité de définir des poids dans les mots des termes de l'ontologie et de calculer des degrés de similarité pour l'annotation a été évaluée expérimentalement.

mesure	similarité entre *Oignon rouge* et	calculs	résultat
dice	oignon de printemps	$\frac{2\times(1\times1+1\times0+0\times0+0\times0.2)}{(1^2+1^2+0^2+0^2)+(1^2+0^2+0^2+0.2^2)}$	0.658
	chou rouge	$\frac{2\times(1\times0+1\times0.2+0\times1+0\times0)}{(1^2+1^2+0^2+0^2)+(0^2+0.2^2+1^2+0^2)}$	0.132
cosinus	oignon de printemps	$\frac{1\times1+1\times0+0\times0+0\times0.2}{\sqrt{(1^2+1^2+0^2+0^2)\times(1^2+0^2+0^2+0.2^2)}}$	0.693
	chou rouge	$\frac{1\times0+1\times0.2+0\times1+0\times0}{\sqrt{(1^2+1^2+0^2+0^2)\times(0^2+0.2^2+1^2+0^2)}}$	0.139

Tab. 3.2 – Exemple de mesures de similarité par Dice et cosinus

Les noms d'aliments (en langue anglaise), contenus dans les tableaux de publications scientifiques en microbiologie alimentaire recueillies sur internet, ont été collectés pour former une liste de 185 termes distincts. Deux ontologies ont été utilisées pour annoter ces termes : le Codex Alimentarius, taxonomie d'aliments structurée selon trois niveaux de regroupement (1644 termes au total), utilisé par l'Organisation Mondiale de la Santé, et la partie concernant les aliments de l'ontologie de Sym'Previus, à structure plus hétérogène, contenant 507 termes avec une profondeur maximale de 7 (selon la relation de spécialisation des termes).

Chacune de ces deux ontologies a été retravaillée manuellement pour assigner un poids à chaque mot de chaque terme. Pour simplifier le travail d'annotation, seules deux valeurs de poids possibles ont été considérées (outre le poids nul pour les mots vides) : l'annotateur devait choisir pour chaque mot entre *importance mineure* et *importance majeure*. Une première série d'évaluations a permis de déterminer qu'il était plus efficace de choisir un poids de 0.2 pour les mots d'importance mineure plutôt qu'un poids de 0.5 : en effet, cette différence de poids ne joue pas seulement sur les écarts de score mais même sur l'ordre des termes de l'ontologie selon leur score de similarité avec le terme du web. Les mots d'importance majeure, quant à eux, se voient attribuer un poids de 1. Pour chacun des termes récoltés sur le web, le "best match", c'est-à-dire le terme de l'ontologie qui en est le plus proche par sa signification, a été déterminé manuellement dans chacune des deux ontologies.

Chaque terme du web a ensuite été annoté automatiquement, pour chaque ontologie, en utilisant les quatre mesures de similarité présentées précédemment : le score de ressemblance ne prenant pas en compte le poids des mots dans les termes de l'ontologie (cf. formule 2.2), le degré de similarité tenant compte des poids des mots que j'ai proposé (formule 3.1), le coefficient de Dice (formule 3.2), et la mesure cosinus (formule 3.3). Pour chaque terme du web, pour chaque ontologie et pour chaque mesure, toute l'ontologie est parcourue en calculant la similarité entre le terme du web et chacun des termes de l'ontologie ; seuls les termes de l'ontologie ayant un degré de similarité non nul avec le terme du web sont conservés et ordonnés par similarité descendante.

L'évaluation est faite en regardant en quelle position se retrouve le "best match" défini manuellement. Les résultats présentés dans le tableau 3.3 sont évalués « au pire », c'est-à-dire qu'en cas d'égalité de mesures de similarité, le "best match" est considéré comme apparaîssant après les autres annotations de même mesure de similarité. Cette évaluation « au pire » est dictée par la façon dont seraient présentés les choix à un annotateur humain dans le cadre d'une annotation semi-automatique : pour chaque terme à annoter, l'annotateur se voit proposer une liste des n termes ayant les meilleures mesures de similarité, dans laquelle il doit choisir la bonne annotation. Si plusieurs termes de l'ontologie ont une même mesure de similarité avec le terme du web, la liste ne doit pas pour autant être allongée : si le "best match" n'est pas dans les n premiers termes de façon stricte, le fait qu'il sera ou non présenté à l'annotateur n'est pas maîtrisé.

Nous évaluons également la proportion d'aliments pour lesquels le "best match" a une mesure de similarité non nulle avec le terme du web (c'est-à-dire la proportion de termes du web ayant au moins un mot commun avec leur "best match"). Cela nous permet en effet d'évaluer la couverture maximale qui peut être atteinte en utilisant une approche de comparaison de termes par comparaison mot à mot.

ontologie	Codex Alimentarius		Sym' Previus	
termes dont le "best match" a un score non nul	60%		78%	
mesure	termes dont le "best match" est : en première position	dans les 5 premières positions	termes dont le "best match" est : en première position	dans les 5 premières positions
sans poids	30%	46%	45%	61%
mesure proposée	34%	52%	46%	65%
dice	34%	52%	46%	66%
cosinus	34%	52%	46%	66%

Tab. 3.3 – Résultats expérimentaux pour l'annotation de 185 noms d'aliments distincts

L'ontologie Sym'Previus, bien que beaucoup plus petite que le Codex Alimentarius, permet d'annoter plus efficacement les termes du web. En effet cette ontologie a été créée spécialement pour le domaine de la microbiologie alimentaire : elle contient des noms d'aliments pertinents pour le domaine, notamment des produits transformés, alors que le Codex Alimentarius, plutôt orienté sur la représentation des matières premières, est moins bien adapté à l'annotation

de publications en microbiologie alimentaire. Ceci met en avant le fait que la méthodologie proposée doit pouvoir s'appuyer sur une ontologie bien spécifique au domaine traité.

Les trois mesures de similarité utilisant les poids des mots dans les termes de l'ontologie apportent des qualités d'annotation similaires, même si dans le détail, des différences subsistent dans l'ordre des annotations proposées : les annotations sont différentes, mais de qualité comparable. Nous n'avons donc pas pu conclure quant à la meilleure mesure à utiliser : par la suite, les trois mesures sont conservées. Par contre, l'utilisation des poids des mots dans les termes de l'ontologie permet d'obtenir une annotation de meilleure qualité que lorsque le score de ressemblance sans poids est utilisé ; cette différence de qualité, bien qu'assez faible, est systématique.

Enfin, beaucoup plus de "best match" sont retrouvés lorsque les 5 premières annotations sont considérées, plutôt que lorsque seule la première est considérée : ceci justifie le fait que nous conservions pour l'annotation l'ensemble des termes de l'ontologie auxquels est associé leur score de similarité avec le terme du web, plutôt que de choisir simplement le terme de l'ontologie dont le degré de similarité avec le terme du web est le plus élevé. La pertinence du degré de similarité est également validée, puisque la majorité des "best match" se trouvent dans les premières positions selon ce score de similarité (dans l'ontologie Sym'Previus, 46% des "best match" se trouvent en première position selon ce degré de similarité, et 65 à 66% des "best match" se trouvent parmi les 5 premières positions).

3.2 Autres mesures de similarité entre deux termes

Les mesures de similarité entre deux termes sur lesquelles repose notre système d'annotation utilisent des égalités de mots lemmatisés. Cependant il existe d'autres techniques pour mesurer des similarités entre termes. Nous présentons en section 3.2.1 des mesures de similarité utilisant une ontologie tierce comprenant les deux termes à comparer. Une méthode de comparaison de mots utilisant non pas l'égalité des mots une fois lemmatisés, mais l'égalité de petites portions de mots (sur des fenêtres de quelques lettres) est étudiée en section 3.2.2. Enfin nous présentons en section 3.2.3 une mesure, non pas de similarité mais de distance entre termes, basée sur l'exploitation de l'immense quantité d'information disponible sur le web pour s'abstraire de ressources lexicales.

3.2.1 Utilisation d'une ontologie tierce

Les mesures de similarité que j'ai utilisées sont fondées sur une ressemblance lexicale entre les termes. Or ce que l'on cherche à capturer pour l'annotation est une ressemblance sémantique. Il existe de nombreuses mesures de

similarité sémantique entre deux termes qui utilisent une ontologie (ou simplement une taxonomie). Une étude de plusieurs de ces mesures est proposée dans [Blanchard et al., 2005]. Ces mesures sont en réalité des mesures de similarité entre concepts d'une même ontologie : si le nom donné à un concept est considéré comme un terme, alors ces mesures peuvent être appliquées pour calculer une similarité entre termes. Ceci n'est bien évidemment possible qu'à condition que l'ontologie contienne deux concepts dont les noms sont égaux aux deux termes à comparer.

Certaines mesures, fondées sur la théorie de l'information, utilisent les probabilités d'apparition de chacun des deux concepts et/ou de leur généralisant commun. Ainsi, [Resnik, 1999] propose d'utiliser, comme mesure de similarité entre deux concepts, la quantité d'information apportée par leur généralisant commun le plus spécifique. Soient C_1 et C_2 les concepts à comparer, $Commun$ l'ensemble de leurs généralisants communs et Pr la fonction qui associe à un concept la probabilité d'apparition d'une instance de ce concept dans un document d'un corpus :

$$sim_{Resnik}(C_1, C_2) = max_{C \in Commun}(-log(Pr(C))) \tag{3.4}$$

[Lin, 1998] propose une mesure prenant en compte à la fois la quantité d'information apportée par le généralisant commun le plus spécifique (noté C_{comSpe}) et celle apportée par chacun des concepts :

$$sim_{Lin}(C_1, C_2) = \frac{2 \times log(Pr(C_{comSpe}))}{log(Pr(C_1)) + log(Pr(C_2))} \tag{3.5}$$

Une autre mesure, non pas de similarité mais de distance, est proposée par [Jiang & Conrath, 1997] :

$$dist_{Jiang}(C_1, C_2) = -log(Pr(C_1) - log(Pr(C_2) + 2 \times log(Pr(C_{comSpe})) \tag{3.6}$$

L'inconvénient de telles mesures est qu'elles nécessitent l'établissement d'un corpus de documents de référence dans le domaine étudié, afin de calculer les probabilités d'apparition des concepts dans un document. [Seco et al., 2004] propose une mesure utilisant l'information intrinsèquement contenue dans la structure de l'ontologie WordNet [Fellbaum, 1998] pour calculer la quantité d'information apportée par un concept : soit C un concept de l'ontologie, $sub(C)$ l'ensemble des concepts subsumés par C et $nbMax$ le nombre total de concepts dans l'ontologie, alors la quantité d'information apportée par C est :

$$QI(C) = 1 - \frac{log(\mid sub(C) \mid +1)}{log(nbMax)} \tag{3.7}$$

[Seco et al., 2004] propose alors trois mesures de similarité sémantique inspirées des mesures de Resnik, Lin et Jiang & Conrath présentées précédemment :

$$sim^{Seco}_{Resnik}(C_1, C_2) = max_{C \in Commun}(QI(C)) \tag{3.8}$$

$$sim_{Lin}^{Seco}(C_1, C_2) = \frac{2 \times max_{C \in Commun}(QI(C))}{QI(C_1) + QI(C_2)} \quad (3.9)$$

$$sim_{Jiang}^{Seco}(C_1, C_2) = 1 - \frac{QI(C_1) + QI(C_2) - 2 \times max_{C \in Commun}(QI(C))}{2} \quad (3.10)$$

Une méthode à envisager pour annoter les termes du web avec des termes de l'ontologie serait alors d'utiliser une ontologie tierce plus complète, qui contiendrait tous les termes de l'ontologie et les termes du web à annoter et aurait les bonnes propriétés de structure (le nombre de spécialisations d'un terme doit rendre compte du degré de généralité de ce terme). On choisirait pour l'annotation le terme de l'ontologie initiale le plus proche du terme du web dans l'ontologie tierce selon l'une des mesures de similarité sémantique précitées.

Cependant, cette méthode nécessite l'existence d'une ontologie dans laquelle les termes à comparer sont tous représentés, ce qui n'est pas le cas pour notre application. En effet, nous avons effectué des tests systématiques en utilisant l'ontologie WordNet [Fellbaum, 1998] : pour chacun des 507 noms d'aliments de l'ontologie de Sym'Previus (version traduite en anglais), ainsi que pour chacun des 185 noms d'aliments issus de publications en microbiologie alimentaire utilisés pour nos tests (cf. section 3.1.5), nous avons recherché si le terme était présent dans WordNet en lançant la requête directement sur le site de recherche en ligne [Miller & team, 2006]. Seulement 37,3% des aliments de notre ontologie, et 35,7% des aliments issus des publications sont représentés dans WordNet.

Nous avons réalisé les mêmes tests systématiques sur le thesaurus AgroVoc : il s'agit du thesaurus utilisé par la FAO (Food and Agriculture Organization), qui devrait donc contenir de nombreux noms d'aliments. Nous avons réalisé nos tests sur la version téléchargeable en fichier à plat [AgroVoc, 2007], qui donne uniquement accès à la liste des termes existant dans le thesaurus, sans relations entre eux. Il est cependant à noter que le thesaurus AgroVoc est en cours de transformation en une véritable ontologie [Soergel et al., 2004] : ceci rendrait possible l'utilisation de similarités basées sur l'utilisation d'une ontologie tierce telles que décrites plus haut, si les termes à comparer étaient effectivement présents dans AgroVoc. Nos tests nous ont cependant montré que seulement 29,6% des termes de la hiérarchie des aliments de Sym'Previus, et seulement 20% des termes issus des publications en microbiologie alimentaire, se retrouvent dans AgroVoc.

En fait, notre ontologie comme les publications contiennent des aliments très spécialisés (et de plus les noms d'aliments sont très variables d'une publication à l'autre), ce qui limite nos chances de les trouver dans une ontologie tierce : c'est le cas par exemple du terme *Pre-cooked milk-based dishes* (Plats cuisinés à base de lait) provenant de l'ontologie de Sym'Previus, ou du terme *acidified vacuum-packed American processed cheese* (fromage fondu de type américain, acidifié et emballé sous vide) trouvé dans une publication. Ce problème peut se retrouver dans tout domaine d'application qui utilise un vocabulaire à la fois trop spécifique pour être présent dans une ontologie générale, et trop variable

pour être entièrement décrit dans l'ontologie de domaine.

3.2.2 Les n-grammes

Les mesures de similarité que nous avons utilisées reposent sur une égalité entre mots. Cette égalité est évaluée après lemmatisation (un mot au pluriel sera considéré comme égal au mot au singulier, un verbe conjugué comme égal à son infinitif, etc.). Il existe cependant d'autres techniques qui permettent d'évaluer des proximités entre mots, cette proximité pouvant ensuite être utilisée pour calculer une proximité entre termes.

[Lin, 1998] propose ainsi un calcul de similarité entre mots utilisant les n-grammes : chaque mot est représenté par un ensemble de chaînes de caractères de taille n, trouvées en déplaçant caractère par caractère une fenêtre de taille n depuis le début jusqu'à la fin du mot. Par exemple, l'ensemble des trigrammes de *salmon* est $trigr(salmon) = \{sal, alm, lmo, mon\}$. Une mesure de similarité entre deux mots m_1 et m_2, avec $Pr(t)$ la probabilité d'apparition du trigramme t dans un mot, est donnée par :

$$sim(m_1, m_2) = \frac{2 \times \sum_{t \in trigr(m_1) \cap trigr(m_2)} log(Pr(t))}{\sum_{t \in trigr(m_1)} log(Pr(t)) + \sum_{t \in trigr(m_2)} log(Pr(t))} \tag{3.11}$$

Cette mesure de similarité permet de trouver de façon efficace des mots provenant de la même racine. Cependant, elle ne permet pas de distinguer les mots de sens équivalent des mots de sens opposé. [Lin, 1998] donne en effet des résultats expérimentaux de calculs de similarité sur le mot *grandiloquent* : les mots *ineloquent* et *eloquantly* obtiennent un même degré de similarité de 0.55 avec le mot *grandiloquent*, alors que l'un veut dire le contraire de l'autre. Cette mesure, purement lexicale, ne capture pas la sémantique des mots.

De nombreuses autres mesures qui utilisent l'égalité de portions de mots plutôt que l'égalité de mots entiers présentent le même type d'inconvénient. C'est la cas par exemple des distances d'édition [Arslan & Egecioglu, 2000], basées sur le nombre d'insertions, suppressions ou substitutions de caractères nécessaires pour transformer une chaîne de caractères en une autre, ou bien des distances de Jaro et de Jaro-Winkler [Winkler, 1990], basées sur le nombre de caractères communs entre les deux chaînes de caractères à comparer, et l'ordre de ces caractères communs.

3.2.3 La distance Google normalisée

La distance Google normalisée [Cilibrasi & Vitanyi, 2007] est une mesure de distance sémantique entre deux termes fondée sur les fréquences d'apparition de chacun des termes ainsi que leur fréquence de co-occurrence dans un très grand corpus tel que l'ensemble des pages indexées par Google.

Soient n_o le nombre de pages renvoyées par Google lors d'une requête portant sur le terme de l'ontologie o, n_w le nombre de pages renvoyées pour une requête sur le terme du web w, et n_{wo} le nombre de pages renvoyées pour une requête sur les documents contenant à la fois w et o. Soit N un facteur de normalisation (fixé par l'utilisateur). Alors la distance Google normalisée entre les termes w et o est donnée par :

$$dist_{Google}(w, o) = \frac{max(log(n_w), log(n_o)) - log(n_{wo})}{log(N) - min(log(n_w), log(n_o))} \tag{3.12}$$

[Cilibrasi & Vitanyi, 2007] indique que le choix du nombre N a peu d'influence sur les résultats dès lors qu'on choisit N tel qu'il soit supérieur au nombre de pages ramenées pour une requête sur n'importe quel terme.

Nous avons mené une expérience portant sur les 185 termes issus de publications en microbiologie alimentaire et représentant des aliments déjà utilisés pour l'expérience présentée en section 3.1.5. Pour chacun de ces termes, nous avons calculé la distance Google normalisée avec chacun des 507 termes de la hiérarchie des aliments de l'ontologie de Sym'Previus (nous avons choisi $N = 10^{10}$). Nous avons ordonné les termes de l'ontologie selon leur distance croissante avec le terme du web, et évalué en quelle position se trouvait le "best match" (le terme de l'ontologie qui est le plus proche par sa signification du terme du web, déterminé manuellement).

Pour 16% des 185 termes du web à annoter, le "best match" était le terme de l'ontologie ayant la distance la plus faible avec le terme du web. Le "best match" était parmi les 5 plus faibles distances pour 29% des termes du web. Par comparaison, rappelons les résultats présentés en section 3.1.5, en utilisant les mêmes noms d'aliments issus du web, les mêmes termes de la hiérarchie des aliments de l'ontologie de Sym'Previus, avec le score de similarité par cosinus : pour 46% des termes du web, le "best match" avait la plus grande similarité avec le terme du web, et pour 66% des termes le "best match" était parmi les 5 plus grandes similarités. Dans le cadre précis de notre application, la méthode des scores de similarité par cosinus donne donc de meilleurs résultats d'annotation. En effet, si la distance Google normalisée permet de rapprocher des termes qui apparaissent souvent ensemble, cela ne veut pas nécessairement dire que ces termes ont la même signification. Ainsi, la distance est plus faible entre "oeuf" et "farine" qu'entre "oeuf" et "ovoproduit" (on peut ici facilement imaginer que cet artefact est dû aux nombreuses recettes de cuisine impliquant à la fois des œufs et de la farine...)

Nous n'excluons pas complètement d'utiliser la distance Google normalisée comme complément à notre technique, dans des cas où les vocabulaires utilisés dans l'ontologie et dans les documents à traiter seraient trop éloignés et rendraient l'approche par comparaison de mots inefficace. Cependant, l'utilisation de la distance Google normalisée a l'inconvénient majeur qu'elle génère un nombre très conséquent de requêtes sur un moteur de recherches sur le web. Comme il

n'existe à l'heure actuelle aucun moteur de recherche sur le web qui soit non commercial, cette méthode est soumise aux aléas des politiques d'accès à l'information décidées par les moteurs de recherche : par exemple, depuis le 5 décembre 2006, Google ne fournit plus de nouvelle clef d'accès pour accéder à son moteur de recherche en mode Service web : tous les programmes qui ont été construits avant cette date pour accéder au moteur de recherche Google ne peuvent plus être utilisés par de nouveaux utilisateurs.

3.3 Prise en compte de la hiérarchie

Jusqu'à présent, l'ontologie fournie pour l'annotation des termes du web a été considérée uniquement en tant que liste de termes. Dans l'ontologie, ces termes sont en réalité structurés par la relation de subsomption dans les hiérarchies des types symboliques. Ce chapitre discute des gains que peut apporter l'utilisation de cette information supplémentaire, soit en utilisant la hiérarchie pour combiner des scores calculés précédemment (section 3.3.1), soit en utilisant la hiérarchie pour définir automatiquement des poids pour les mots des termes du web (section 3.3.2).

3.3.1 Score de ressemblance par hiérarchie

Nous proposons de définir un score de ressemblance entre un terme du web et un terme de l'ontologie, qui utilise les mesures de similarité lexicales présentées dans la section 3.1, mais qui prenne également en compte les relations hiérarchiques entre les termes de l'ontologie.

Le score de ressemblance entre un terme du web et un terme de l'ontologie que nous avons proposé utilise des égalités de mots. Suite aux expérimentations (cf. section 3.1.5), nous avons pu oserver qu'il arrive souvent que le "best match" n'ait pas de mot en commun avec le terme du web, mais que ses fils (qui sont des spécialisations trop fortes pour bien représenter le terme du web) aient des mots en commun avec le terme du web. Cette situation est représentée dans la figure 3.1.

Nous proposons donc d'utiliser la hiérarchie de l'ontologie et les mesures de similarité calculées précédemment, en ajoutant la considération suivante : si des termes de l'ontologie sont représentatifs d'un terme du web, alors leur généralisant direct est également représentatif de ce terme. Un score de ressemblance par hiérarchie entre le terme du web et chaque terme de l'ontologie est donc calculé.

Définition 3.2 *Soit sim_{lex} une mesure de similarité lexicale (au choix, la mesure que nous avons proposée – formule 3.1, le coefficient de Dice – formule 3.2, ou la mesure cosinus – formule 3.3). Soient w un terme du web, o le terme de l'ontologie avec lequel le terme du web est comparé, et F l'ensemble des fils directs*

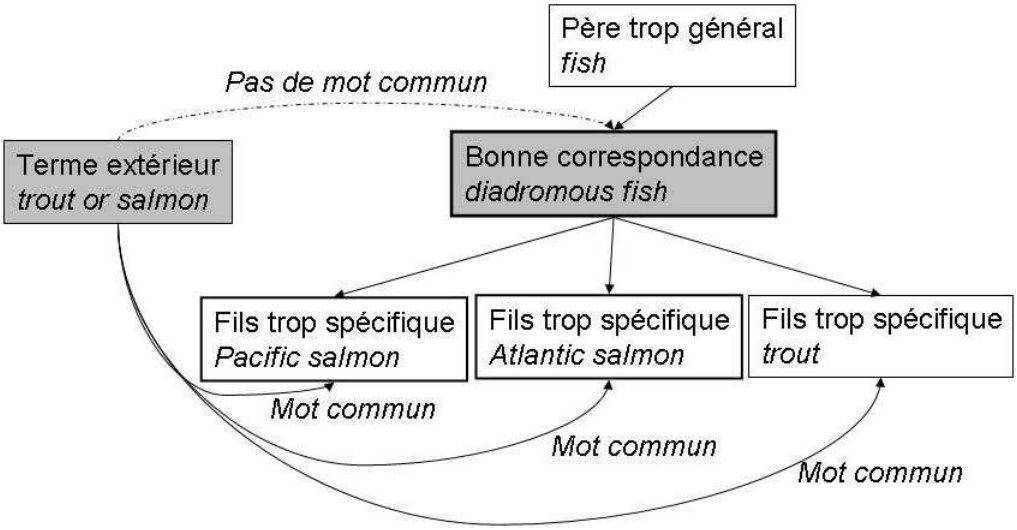

FIG. 3.1 – Terme extérieur n'ayant de mots en commun qu'avec des termes trop spécifiques, exemple du terme *trout or salmon* comparé au Codex Alimentarius

par relation de spécialisation de o dans l'ontologie. Le score de ressemblance par hiérarchie entre w et o est donné par :

$$sim_{hier}(w, o) = 1 - \prod_{f \in F} (1 - sim_{lex}(f)) \tag{3.13}$$

Ce score est inspiré de [Yangarber et al., 2002], où le même type de score est utilisé en extraction d'information, pour combiner les confiances de différents motifs d'extraction d'un même groupe nominal (voir la section 2.1.2). Ici, il s'agit également de combiner un ensemble d'indices en lesquels on a plus ou moins confiance : les différents rapprochements possibles entre le terme du web et chacun des fils directs du terme de l'ontologie sont combinés, la confiance accordée à ces rapprochements étant le degré de similarité lexicale. Plus il y a de fils ayant un degré de similarité lexicale non nul avec le terme du web, plus le score de ressemblance par hiérarchie est élevé ; l'augmentation du score de ressemblance par hiérarchie est d'autant plus forte que les degrés de similarité lexicale entre les fils et le terme du web sont élevés.

Exemple 3.5 *Prenons l'exemple de* trout or salmon*, illustré dans la figure 3.1. Le degré de similarité lexicale entre* salmon *et* diadromous fish *est nul, car il n'y a aucun mot commun (*diadromous fish *représente les poissons qui vivent à la fois en eau douce ou en eau de mer, suivant leur stade de développement). Avec les poids pour le terme du web* trout or salmon *:* $\{trout : 1;\ salmon : 1\}$ *et pour les termes de l'ontologie* Atlantic salmon *:* $\{atlantic : 0.2; salmon : 1\}$*,* Pacific salmon *:* $\{pacific : 0.2;\ salmon : 1\}$ *et*

trout *: {trout : 1}, le calcul de degré de similarité lexicale que nous avons proposé (formule 3.1) donne :* $sim_{lex}(trout\ or\ salmon,\ Atlantic\ salmon) = sim_{lex}(trout\ or\ salmon,\ Pacific\ salmon) = \frac{1+1}{(1+1)+(1+0.2)} = 0.625$ *et* $sim_{lex}(trout\ or\ salmon,\ trout) = \frac{1+1}{(1+1)+(1)} = 0.667$.
Le score de ressemblance par hiérarchie entre salmon *et* diadromous fish *est alors :* $sim_{hier}(trout\ or\ salmon,\ diadromous\ fish) = 1 - (1 - sim_{lex}(trout\ or\ salmon,\ Pacific\ salmon)) \times (1 - sim_{lex}(trout\ or\ salmon,\ Atlantic\ salmon)) \times (1 - sim_{lex}(trout\ or\ salmon,\ trout)) = 1 - (1 - 0.625)(1 - 0.625)(1 - 0.667) = 0.953$

Résultats expérimentaux

Pour chaque terme de l'ontologie proposé pour annoter un terme du web, le score de ressemblance est la valeur maximale entre le degré de similarité lexicale et le score de ressemblance par hiérarchie. La même méthode d'évaluation que pour les mesures de similarité lexicale (cf. section 3.1.5) est utilisée, en ordonnant les annotations selon leur score. Les résultats de cette évaluation sont donnés dans le tableau 3.4.

ontologie	Codex Alimentarius		Sym' Previus	
termes dont le "best match" a un score non nul	68%		81%	
mesure de similarité	termes dont le "best match" est : en première position	dans les 5 premières positions	termes dont le "best match" est : en première position	dans les 5 premières positions
mesure proposée	13%	55%	17%	62%
dice	13%	56%	17%	61%
cosinus	14%	56%	17%	61%

TAB. 3.4 – Résultats expérimentaux pour l'annotation de 185 noms d'aliments distincts avec utilisation du score de ressemblance par hiérarchie

Ces résultats expérimentaux montrent à nouveau que les trois mesures de similarité lexicale utilisant les poids des mots dans les termes de l'ontologie produisent la même qualité d'annotation. Il y a beaucoup moins de "best match" en première position qu'il y en avait en utilisant le degré de similarité lexicale seul. Ceci s'explique facilement par le fait que nous utilisons l'ordre "au pire" ; or, chaque fois que le degré de similarité lexicale permet de trouver le "best match" en première position, le score de ressemblance par hiérarchie de son plus proche

parent dans l'ontologie est égal ou supérieur à cette similarité. Le score de ressemblance par hiérarchie a donc pour effet de "diluer" les résultats. Cependant, l'utilisation du score de ressemblance par hiérarchie permet de récupérer plus de "best match", comme le montre le nombre d'aliments ayant un "best match" avec score non nul.

Sur le Codex Alimentarius, qui est moins bien adapté pour l'annotation des termes du web que l'ontologie de Sym'Previus, l'utilisation du score de ressemblance par hiérarchie permet d'obtenir une meilleure annotation (si le critère de qualité est la présence du "best match" parmi les 5 premières annotations proposées). Par contre, pour l'ontologie de Sym'Previus, l'utilisation du score de ressemblance par hiérarchie n'améliore pas les résultats. L'utilisation du score de ressemblance par hiérarchie peut donc être considérée comme un moyen de pallier une mauvaise adaptation de l'ontologie aux données à annoter.

3.3.2 Gain d'information d'un mot dans l'ontologie

Jusqu'à présent, nous avons choisi d'affecter un poids de 1 à tous les mots des termes du web, et nous avons montré en section 3.1.2 que des techniques issues de la linguistique ne nous permettent pas d'appliquer des poids plus appropriés. Une autre possibilité serait d'appliquer des poids aux mots en fonction du gain d'information qu'ils apportent. Ce gain d'information est une mesure de la façon dont la connaissance d'un mot permet de se concentrer sur une plus petite partie de l'ontologie (sous-arbre ou conjonction de sous-arbres) : plus la zone est petite, plus le mot apporte d'information et donc plus son poids doit être important (c'est-à-dire qu'on donne un poids importants aux mots dans les termes du web qui permettent de focaliser la recherche du terme le plus proche sur une petite partie de l'ontologie). Pour cela, il faut pouvoir mesurer la dispersion d'un mot dans l'ontologie ; cette dispersion est évaluée au moyen de calculs d'entropie.

Mesure d'entropie dans une ontologie pondérée

[Koh & Mui, 2001] propose une mesure d'entropie sur des ontologies dont les termes sont pondérés. Le but de ce travail est la comparaison de deux ontologies représentant des centres d'intérêts de deux utilisateurs différents. Les deux ontologies à comparer sont fondées sur la même taxonomie sous forme d'arbre (i.e. chaque concept n'a qu'un père par la relation de subsomption), représentant les sujets d'intérêt. Seuls diffèrent entre les deux ontologies les poids associés à chaque concept, correspondant à l'intérêt que porte chacun des deux utilisateurs aux différents sujets représentés par les concepts.

Soit une taxonomie sous forme d'arbre, dans laquelle chaque concept C se voit associer un poids $p(C)$. L'ensemble des fils directs de C est noté $f(C)$.
La probabilité conditionnelle d'un sous-sujet k sachant qu'on se situe dans le sujet C est donnée par $Pr(k \mid C) = \frac{p(k)}{\sum_{j \in f(C)} p(j)}$.

La probabilité d'un chemin x constitué dans l'ordre par les concepts $(c_1, c_2, ..., c_n)$ est donnée par $Pr(x) = Pr(c_1) \times \prod_{i=2}^{n} Pr(c_i \mid c_{i-1})$.
La probabilité d'un concept est donnée par la probabilité du chemin qui mène de la racine de l'ontologie à ce concept. La probabilité de la racine de l'ontologie est de 1 (on considère que l'utilisateur est forcément intéressé par quelque chose).

Exemple 3.6 *La figure 3.2 donne une ontologie qui représente les intérêts d'un utilisateur pour différents secteurs de commerce en ligne. Dans cette ontologie, la probabilité de* hôtel-club*, i.e. la probabilité que l'utilisateur s'intéresse à un séjour en hôtel-club plutôt qu'à un autre sujet, est de :*
$Pr(hotelClub) = Pr(commerce) \times Pr(voyages \mid commerce) \times Pr(sejour \mid voyages) \times Pr(hotelClub \mid sejour)$
$= 1 \times \frac{0.8}{0.8+0.1+0.9} \times \frac{0.9}{0.2+0.6+0.2+0.9} \times \frac{0.8}{0.3+0.8+1} = 0.08$

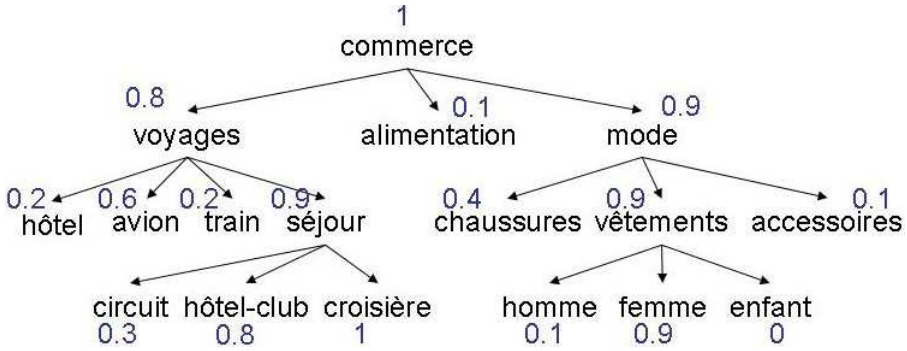

FIG. 3.2 – Intérêts d'un utilisateur de commerce en ligne

Soit $chemin(C)$ l'ensemble des chemins possibles depuis le concept C vers les feuilles du sous-arbre de racine C. L'entropie du sous-arbre de racine C de l'ontologie est donnée par :

$$H(c) = -\sum_{x \in chemin(C))} Pr(x) log_2(Pr(x)) + \sum_{k \in f(C))} Pr(k \mid C) \times H(k) \qquad (3.14)$$

La première partie de cette équation mesure la dispersion dans la distribution des probabilités des différents chemins vers les feuilles, sans se soucier de la forme arborescente de la taxonomie. La deuxième partie de l'équation permet de mesurer la dispersion dans la hiérarchie, en ajoutant l'entropie de chacun des sous-arbres pondérée par la probabilité d'entrer dans ce sous-arbre.

Dans ce travail de thèse, nous nous intéressons également à une ontologie pondérée, mais les poids donnés correspondent à l'importance sémantique des mots dans les termes de l'ontologie. Un mot est choisi, et l'ontologie est pondérée en fonction du poids de ce mot dans chacun des termes. Les poids sur l'ontologie

pour un mot donné sont très souvent nuls (lorsque le mot n'apparaît pas dans un terme), et le fait qu'un poids soit nul sur un terme ne signifie pas qu'il sera nul sur les fils de ce terme. Au contraire, dans le cas des intérêts d'utilisateurs, si un sujet est d'intérêt nul, tous ses sous-sujets le sont également. De plus, alors que dans le cas des intérêts d'utilisateurs seules les feuilles étaient considérées (d'où la définition des chemins), ici tous les termes de l'ontologie sont considérés car ils sont tous susceptibles d'être utilisés pour l'annotation. Pour ces raisons, les calculs précédents ne peuvent pas être appliqués directement.

Nous reprenons donc l'idée présentée dans les calculs précédents, mais en les adaptant à notre problématique : la probabilité d'un terme C dans l'ontologie pondérée selon le poids du mot m ne serait plus la probabilité du chemin de la racine de l'ontologie à ce terme, mais au contraire, les poids des termes fils de C seraient remontés sur leur terme père C. En effet, un terme C peut être atteint de deux manières : soit le terme C est choisi directement, soit l'un de ses termes fils est choisi. Soit $sub(C)$ l'ensemble des termes subsumés (directement ou non) par C, $Onto$ l'ensemble de tous les termes de l'ontologie, et p la fonction qui associe à un terme le poids du mot m dans ce terme. Alors

$$Pr(C) = \frac{p(C) + \sum_{x \in sub(C)} p(x)}{\sum_{x \in Onto} p(x)} \tag{3.15}$$

Comme chaque terme de l'ontologie peut être utilisé pour l'annotation et non pas seulement les feuilles, l'ensemble de tous les chemins vers les feuilles est remplacé dans la formule 3.14 par l'ensemble de tous les termes du sous-arbre, i.e. le terme à la racine du sous-arbre et l'ensemble de tous ses termes fils :

$$H(C) = -Pr(C)log(Pr(C)) - \sum_{x \in sub(C)} Pr(x)log(Pr(x)) + \sum_{k \in f(C)} Pr(k \mid C)H(k) \tag{3.16}$$

avec $Pr(k \mid C) = \frac{Pr(k)}{Pr(C)}$.

L'entropie de l'ontologie pour chacun des mots présents dans l'ontologie peut être calculée par récurrence à partir de la formule 3.16, en appliquant à chaque terme de l'ontologie le poids du mot dans ce terme.

Poids des mots du terme du web selon leur gain d'information

Nous reprenons ici la notion de gain d'information telle qu'utilisée par [Quinlan, 1993].

Définition 3.3 *Le gain d'information apporté par la connaissance d'un attribut a dans un système S est mesuré comme la différence d'entropie du système S sans la connaissance de l'attribut a avec l'entropie du système S lorsqu'on connaît l'attribut a.*

Dans le cadre de ce travail, nous souhaitons calculer le gain d'information d'un mot. Il faut donc calculer :

- l'entropie de l'ontologie (formule 3.16) lorsque tous les concepts sont équivalents, i.e. lorsqu'on attribue un poids de 1 à chacun des termes, y compris ceux qui ne contiennent pas le mot. Cette entropie sert de référence et est notée $H_{ref}(r)$ (r étant le concept à la racine de l'ontologie) ;
- l'entropie de l'ontologie lorsqu'on connaît le mot m, i.e. lorsqu'on attribue à chaque terme un poids égal au poids du mot dans le terme, notée $H_m(r)$.

Alors, le gain d'information apporté par le mot m est donné par :

$$Gain(m) = H_{ref}(r) - H_m(r) \tag{3.17}$$

Ainsi, dans nos calculs de similarité entre termes du web et termes de l'ontologie, il est possible d'attribuer des poids aux mots des termes du web en fonction de leur gain d'information. Pour chacun des mots d'un terme du web, le poids suivant est appliqué :

- si le mot m est présent dans l'ontologie, alors $poids(m) = \frac{Gain(m)}{H_{ref}(r)}$
- si le mot m n'est pas présent dans l'ontologie, son gain d'information ne peut pas être calculé (ou plutôt, son gain d'information est nul), mais il faut tout de même tenir compte de l'existence de ce mot dans les calculs de mesures de similarité ; nous avons donc choisi un poids intermédiaire $poids(m) = 0.5$.

L'utilisation de ces poids n'a été testée que sur l'ontologie du Codex Alimentarius : en effet, la mesure d'entropie sur une ontologie pondérée a été construite pour les taxonomies sous forme d'arbres, ce qui est le cas du Codex Alimentarius, mais n'est pas le cas de l'ontologie de Sym'Previus qui présente des héritages multiples.

Pour l'ontologie du Codex Alimentarius, nous avons refait les expérimentations présentées en section 3.1.5, en appliquant, avant calcul des degrés de similarité, les poids calculés en utilisant le gain d'information pour les mots des termes du web. Le tableau 3.5 présente une comparaison de la qualité d'annotation en utilisant ces poids ou en donnant à tous les mots des termes du web un poids de 1.

Le gain dans la qualité de l'annotation apporté par cette méthode est très faible au vu des contraintes imposées par l'utilisation de cette méthode (en termes de coût de calcul mais surtout de contraintes structurelles sur la hiérarchie, qui doit obligatoirement être sous forme d'arbre). Pour la suite de ce travail, nous avons donc décidé de ne pas retenir l'utilisation du gain d'information pour déterminer le poids des mots dans les termes du web.

poids des mots du terme du web	poids de 1 partout		poids calculés par gain d'information	
mesure	termes dont le "best match" est : en première position	dans les 5 premières positions	termes dont le "best match" est : en première position	dans les 5 premières positions
mesure proposée	34%	52%	35%	52%
dice	34%	52%	35%	52%
cosinus	34%	52%	35%	52%

TAB. 3.5 – Résultats expérimentaux pour l'annotation de 185 noms d'aliments distincts avec le Codex Alimentarius

Conclusion du chapitre

Nous avons proposé dans ce chapitre différentes mesures permettant de comparer un terme à l'intérieur d'une cellule d'un tableau provenant du web avec différents termes de l'ontologie. Nous avons tout d'abord utilisé une approche purement lexicale, par comparaison de mots. Nous avons fait un certain nombre de tentatives pour introduire plus de sémantique dans notre approche, mais nous avons été soit déçus par les résultats (dans les cas de la distance Google normalisée, du score de ressemblance par hiérarchie ou du poids des mots du web par calcul du gain d'information dans l'ontologie), soit confrontés à l'impossibilité d'utiliser ces techniques par manque de ressources sémantiques suffisamment précises (dans le cas des mesures de similarité par utilisation d'une ontologie tierce). Ceci est notamment dû au fait que notre domaine d'application utilise un vocabulaire très précis, bien représenté dans notre ontologie de domaine mais non recensé dans les ontologies plus générales : nous pensons que ce problème peut apparaître dans divers autres domaines d'application.

Le travail présenté dans ce chapitre a donné lieu à diverses publications, dans la communauté informatique comme dans le domaine d'application. [Hignette, 2005] et [Hignette et al., 2005] présentent le score de similarité lexicale entre un terme du web et un terme de l'ontologie défini en section 3.1.3 ainsi que le score de resemblance par hiérarchie (section 3.3.1). [Buche et al., 2006] aborde le score de similarité par cosinus entre un terme du web et un terme de l'ontologie (section 3.1.4) ainsi que les problèmes de représentation et d'interrogation des tableaux annotés en format XML. [Hignette et al., 2006] aborde le score de similarité lexicale entre un terme du web et un terme de l'ontologie défini en section 3.1.3 du point de vue de la microbiologie alimentaire, pour reconnaître quels sont les aliments analysés dans une publication donnée.

Nous avons abordé dans ce chapitre l'annotation des cellules d'un tableau

lorsqu'elles contiennent des données symboliques. Dans le chapitre suivant, nous nous intéressons à l'annotation des colonnes du tableau.

Chapitre 4

Annotation des colonnes d'un tableau

Dans le chapitre précédent, nous avons montré comment le contenu des cellules des colonnes symboliques des tableaux peut être annoté à partir des termes de l'ontologie. Ce travail, mené au début de la thèse, nous a amenés à nous rendre compte qu'il était indispensable de traiter de façon distincte les colonnes contenant des données numériques et les colonnes contenant des données symboliques. Ainsi, la première étape de notre méthode d'annotation des colonnes d'un tableau, présentée en section 4.1, consiste à différencier les colonnes numériques des colonnes symboliques.

Ensuite, nous déterminons le type de chaque colonne : nous avons particulièrement travaillé sur la robustesse de notre méthode en adoptant une approche multi-critères, qui utilise conjointement le titre de la colonne et son contenu effectif. Nous présentons en section 4.2 comment déterminer le type d'une colonne numérique en fonction des types numériques de l'ontologie, avant de présenter en section 4.3 comment déterminer le type d'une colonne symbolique en fonction des types symboliques de l'ontologie.

4.1 Reconnaissance des colonnes numériques et symboliques

Faire la différence entre les colonnes numériques et les colonnes symboliques dans un tableau n'est pas si simple qu'il y paraît, notamment dans le domaine de la microbiologie alimentaire où de nombreuses données symboliques comportent des chiffres (par exemple la souche de microorganisme "E. coli O 157 : H7") alors que les données numériques comportent souvent des mots (unités, précision d'un intervalle de confiance...).

La solution que nous proposons pour distinguer les colonnes numériques des colonnes symboliques est une méthode de classification par règles qui tient compte

des unités des types numériques définies dans l'ontologie : il suffit alors de modifier l'ontologie pour que cette méthode soit applicable à n'importe quel autre domaine d'application. Nous présentons en section 4.1.1 notre méthode de classification, et en section 4.1.2 sa validation expérimentale.

4.1.1 Classification par règles utilisant les unités définies dans l'ontologie

Nous commençons par identifier tous les nombres présents dans la colonne à analyser. Les nombres sont reconnus selon l'expression régulière `(digit)+((',’|‘.’)(digit)+)*`, avec `digit` correspondant à l'un des dix chiffres `(‘0’|‘1’|...|‘9’)`. Les nombres en notation scientifique sont reconnus selon l'expression régulière `digit ‘.’ (digit)+ ‘x 10 ’ (digit)+`.

Ensuite, nous recherchons dans la colonne à analyser toutes les occurrences des unités définies dans l'ontologie. Enfin, nous identifions les mots, qui sont des chaînes de caractères ne correspondant ni à une unité ni à un *indicateur de résultat absent* (rappelons que les *indicateurs de résultat absent* sont définis dans l'ontologie : voir section 1.4).

On applique alors, sur chaque cellule de la colonne à analyser, les règles de classification suivantes :

- si la cellule contient un nombre en notation scientifique, la cellule est numérique ;
- si la cellule contient un nombre immédiatement suivi d'une unité, la cellule est numérique ;
- si aucun des deux cas précédents ne s'applique, le nombre d'indices de numérique et le nombre d'indices de symbolique sont comptés :
  - *indice de numérique* : une unité ou un nombre ;
  - *indice de symbolique* : un mot. Les *indicateurs de résultat absent* ne sont pas des indices, ils peuvent aussi bien représenter un résultat numérique absent qu'une valeur symbolique absente.

  Une fois que les indices de numérique et les indices de symbolique ont été comptés dans la cellule, les règles suivantes sont appliquées :
  - si la cellule contient plus d'*indices de numérique* que d'*indices de symbolique*, la cellule est numérique ;
  - si la cellule contient plus d'*indices de symbolique* que d'*indices de numérique*, la cellule est symbolique ;
  - si la cellule ne contient pas d'indices ou s'il y a autant d'*indices de symbolique* que d'*indices de numérique*, la cellule est de type inconnu.

Une fois que toutes les cellules de la colonne ont été classifiées, la colonne elle-même est classifiée :

- si une colonne contient plus de cellules classifiées numériques que de cellules classifiées symboliques, elle est classifiée comme étant numérique ;

- si la colonne contient plus de cellules classifiées symboliques que de cellules classifiées numériques, elle est classifiée comme étant symbolique ;
- si la colonne contient autant de cellules classifiées symboliques que de cellules classifiées numériques, la colonne est classifiée comme étant numérique. En effet, à cette étape de notre système d'annotation, nous ne souhaitons pas aboutir à une absence de diagnostic s'il y a autant de cellules classées comme numériques que de cellules classées comme symboliques : nous avons choisi de favoriser les numériques pour trancher ce cas ambigu car nous avons constaté expérimentalement que ce cas était essentiellement lié à un grand nombre de *résultats absents* dans la colonne, ce qui est plus fréquent pour les mesures expérimentales numériques.

Exemple 4.1 *Considérons le tableau 4.1. Ce tableau ne comporte qu'une seule ligne dans le document original : chaque colonne sera donc classifiée comme symbolique ou numérique en fonction de la classification de son unique cellule :*

- *pour la première colonne, la cellule contient deux mots, aucun nombre et aucune unité : la cellule, et donc la colonne, est symbolique ;*
- *pour la deuxième colonne, la cellule contient un nombre et un mot (en effet, la chaîne de caractères "not specified" fait partie des indicateurs de résultat absent, elle n'est donc pas considérée dans le décompte des mots) : la cellule est donc classifiée comme de type inconnu. La colonne contient alors un nombre égal de cellules symboliques et numériques (ce nombre étant égal à 0). La colonne est donc considérée comme numérique.*
- *pour la troisième colonne, la cellule contient un nombre et aucun mot : la cellule, et donc la colonne, est numérique ;*
- *pour la quatrième colonne, la cellule contient deux nombres et aucun mot : la cellule, et donc la colonne, est numérique ;*
- *pour la cinquième colonne, la cellule contient un nombre, aucune unité et trois mots : la cellule, et donc la colonne, est symbolique.*

Products	**Samples tested**	**Positive for Campylobacter (%)**	**Year**	**Reference**
Chicken products	1320 (approx - not specified)	0.07	1992/1994	Campbell and Gilbert, 1995

TAB. 4.1 – Exemple de tableau : Reported prevalence of Campylobacter in ready-to-eat New Zealand Poultry (whole, pieces) (issu de [Lake et al., 2003])

4.1.2 Comparaison avec une méthode naïve

Nous avons cherché à évaluer l'apport de l'utilisation de connaissances liées au domaine (les unités définies dans l'ontologie) pour la classification des colonnes en numérique ou symbolique. Pour cela, nous la comparons à une méthode de classification "naïve", n'utilisant pas de connaissances liées au domaine. Pour notre classifieur naïf, toute cellule contenant un (ou des) chiffre(s) est considérée comme numérique, et une colonne est considérée comme numérique si plus de la moitié de ses cellules sont numériques.

La reconnaissance des colonnes numériques et des colonnes symboliques a été évaluée sur 60 tableaux. Ces tableaux, issus de publications scientifiques, de documents de cours ou de rapports de projets, ont été choisis comme intéressants pour le domaine de la microbiologie alimentaire par Eric Mettler, de la société Soredab, notre partenaire dans les projets e.dot et WebContent.

Une classe *symbolique* ou *numérique* a été manuellement assignée à chacune des colonnes de ces tableaux, résultant en 263 colonnes numériques et 86 colonnes symboliques. Nous avons comparé les résultats apportés par notre classifieur utilisant des connaissances de l'ontologie et le classifieur naïf. Les résultats de cette classification sont donnés dans le tableau 4.2.

	classification utilisant les unités		classification naïve	
prédit / manuel	numérique	symbolique	numérique	symbolique
numérique	262	1	229	34
symbolique	5	81	13	73
précision	98%		87%	

Tab. 4.2 – Résultats de classification numérique/symbolique sur 349 colonnes.

On voit ici que l'utilisation des règles de classification faisant référence aux connaissances du domaine permet une amélioration sensible de la reconnaissance des colonnes numériques et des colonnes symboliques.

Nous avons choisi de n'utiliser comme connaissances du domaine pour la classification en numérique ou symbolique, que les unités de types numériques. Il aurait été envisageable de rechercher également les termes des taxonomies des types symboliques comme indice supplémentaire pour reconnaître les colonnes symboliques. Cependant, cela augmenterait de beaucoup le temps de calcul (notre classification en numérique et symbolique est notamment utilisée pour limiter le nombre de colonnes sur lesquelles on applique une comparaison avec les termes de l'ontologie). De plus, nous avons pu vérifier expérimentalement que les erreurs faites par notre méthode de classification n'auraient pas pu être évitées en utilisant les taxonomies des types symboliques comme indice supplémentaire.

4.2 Reconnaissance du type d'une colonne numérique

Lorsqu'une colonne a été reconnue comme étant numérique, nous proposons une méthode pour identifier quel est son type, en fonction des types numériques présents dans l'ontologie. Notre méthode repose sur le calcul d'un score entre la colonne et chacun des types numériques de l'ontologie.

Le score final d'un type numérique pour une colonne numérique, présenté en section 4.2.3, est une combinaison entre deux scores :

- le score du type pour la colonne d'après les unités présentes dans la colonne (section 4.2.1) ;
- le score du type pour la colonne d'après le titre de la colonne (section 4.2.2).

Le calcul du score final de chaque type numérique pour une colonne numérique nous permet de déterminer le type de la colonne (section 4.2.4). Nous présentons nos résultats expérimentaux en section 4.2.5.

4.2.1 Score d'après les unités présentes dans la colonne

Soit u une unité d'une colonne numérique *col* et T_u l'ensemble de tous les types numériques pouvant s'exprimer dans cette unité. Le score du type *type* pour l'unité u est

- $score(u, type) = \frac{1}{T_u}$ si u est une unité valable pour *type* ;
- $score(u, type) = 0$ si le type *type* ne s'exprime pas dans l'unité u.

Soit U_{col} l'ensemble de toutes les unités présentes dans la colonne. On considère également les unités présentes dans le titre de la colonne, à condition qu'elles ne fassent pas partie d'un couple nombre-unité. En effet, dans notre application, un couple nombre-unité dans le titre de la colonne représente généralement une précision de condition expérimentale (par exemple "at 37°C"). Nous n'avons pas vérifié que cette règle était valable dans d'autres domaines d'application : la prise en compte ou non des unités dans le titre de la colonne, avec d'éventuelles règles d'exclusion, peut être personnalisée en fonction de l'application.

Le score du type *type* pour la colonne *col* d'après les unités présentes dans la colonne est :

$$score_{unit}(col, type) = max_{u \in U_{col}}(score(u, type)) \tag{4.1}$$

Le score d'un type donné pour la colonne d'après les unités présentes dans la colonne est ainsi le score de l'unité présente dans la colonne qui est la plus discriminante pour ce type. Si la colonne ne contient aucune des unités définies dans l'ontologie, alors on considère qu'elle contient l'unité #NONE, et les scores pour les différents types numériques sont calculés de la même manière que pour une unité « classique ».

Exemple 4.2 *Reprenons les colonnes ayant été reconnues comme numériques dans le tableau 4.1. Les colonnes ayant pour titre « Samples tested » et « Year » ne présentent aucune unité. On considère donc qu'elles présentent l'unité #NONE. Les différents types pouvant s'exprimer dans cette unité sont : "AW water activity", "Number outbreaks death", "PH", "Samples positive", "Samples tested" et "Year" (voir la liste des différents types numériques de l'ontologie en chapitre 1). Le score de chacun de ces 6 types numériques pour chacune des deux colonnes selon les unités présentes dans ces colonnes est donc de* $1/6 \approx 0,167$*. Les autres types numériques ont un score de 0 pour ces colonnes.*

La colonne ayant pour titre « Positive for Campylobacter (%) », en revanche, contient l'unité %, présente dans le titre de la colonne et non précédée d'un nombre. Les différents types pouvant s'exprimer dans cette unité sont : "CO2", "N2", "NACL", "O2" et "Samples positive". Le score de chacun de ces types numériques pour la colonne selon les unités présentes dans la colonne est donc de $1/5 = 0,2$*. Les autres types numériques ont un score de 0 pour cette colonne.*

4.2.2 Score d'après le titre de la colonne

Soit *col* une colonne numérique. On considère le titre de cette colonne, en ne conservant que les mots qui ne correspondent ni à une unité décrite dans l'ontologie, ni à un mot de la *stopword list* de l'ontologie. Les mots restants sont lemmatisés, et un poids de 1 est attribué à chaque mot.

On calcule le score de similarité entre le titre de la colonne et chacun des types numériques de l'ontologie, en utilisant au choix l'un des scores de similarité présentés en section 3.1.4, avec :

- comme terme du web, le titre de la colonne, transformé comme décrit précédemment ;
- comme terme de l'ontologie, le nom d'un type numérique de l'ontologie.

Soit t_{titre} le titre de la colonne *col* et t_{type} le nom du type *type*, alors le score du type *type* pour la colonne *col* d'après le titre de la colonne est :

$$score_{titre}(col, type) = sim(t_{titre}, t_{type}) \tag{4.2}$$

avec *sim* l'une des mesures de similarité décrites en section 3.1.4.

Exemple 4.3 *Continuons sur l'exemple des colonnes ayant été reconnues comme numériques dans le tableau 4.1. Nous considérons ici des poids de 1 sur tous les mots (dans les titres des colonnes ainsi que dans les noms des types numériques). Si la mesure cosinus est choisie comme mesure de similarité, les scores suivants sont obtenus :*

- *pour la colonne ayant pour titre « Samples tested », le type numérique "Samples tested" a un score d'après le titre de la colonne de* $\frac{2}{\sqrt{2\times 2}} = 1$*, et le type "Samples positive" un score de* $\frac{1}{\sqrt{2\times 2}} = 0,5$*. Tous les autres types*

numériques, n'ayant pas de mot commun entre leur nom et le titre de la colonne, ont un score selon le titre de la colonne de 0.

- *pour la colonne ayant pour titre « Positive for Campylobacter (%) », le mot* for *étant un stopword, seuls les mots* Positive *et* Campylobacter *sont considérés. Le type numérique "Samples positive" a alors pour cette colonne un score selon le titre de la colonne de* $\frac{1}{\sqrt{2\times 2}} = 0,5$*. Tous les autres types numériques ont un score selon le titre de la colonne de 0.*
- *pour la colonne ayant pour titre « Year », le type numérique "Year" a un score selon le titre de la colonne de* $\frac{1}{\sqrt{1\times 1}} = 1$*. Tous les autres types numériques ont un score selon le titre de la colonne de 0.*

4.2.3 Score final d'un type numérique pour une colonne numérique

Soient *type* un type numérique défini dans l'ontologie et *col* une colonne numérique. On commence par regarder les valeurs numériques contenues dans la colonne *col*. Si la colonne contient une (ou des) valeur(s) en dehors de l'intervalle de valeurs possibles pour le type *type*, alors le score final du type *type* pour la colonne *col* est nul : $score_{final}(col, type) = 0$.

Si toutes les valeurs numériques contenues dans la colonne sont compatibles avec l'intervalle de valeurs possibles associé au type *type*, alors le score de la colonne *col* pour le type *type* est donné par :

$$score_{final}(col, type) = 1 - (1 - score_{titre}(col, type)) \times (1 - score_{unit}(col, type)) \tag{4.3}$$

Ce score est inspiré de [Yangarber et al., 2002], où une mesure similaire est utilisée pour combiner les confiances que l'on a en différentes règles de reconnaissance d'une entité nommée. Les deux scores se renforcent ainsi mutuellement, mais il suffit que l'un des deux scores soit bon pour que le score final soit bon.

Exemple 4.4 *Prenons l'exemple de la colonne ayant pour titre « Samples tested » dans le tableau 4.1. Le calcul des scores des différents types numériques pour cette colonne a été présenté en exemple 4.2 pour le score d'après les unités de la colonne, et en exemple 4.3 pour le score d'après le titre de la colonne : il ne reste plus qu'à combiner ces deux scores.*

- *le type numérique "Samples tested" a pour score final* $1 - (1 - 1) \times (1 - 0,167) = 1$
- *les types numériques "AW water activity", "Number outbreaks death", "PH" et "Year" ont pour score final* $1 - (1 - 0) \times (1 - 0,167) = 0,167$
- *le type numérique "Samples positive" a pour score final* $1 - (1 - 0,5) \times (1 - 0,167) = 0,583$
- *les autres types numériques ont un score final nul.*

4.2.4 Choix d'un type numérique pour une colonne numérique

Soit *col* une colonne numérique, les différents types numériques de l'ontologie sont ordonnés suivant leur valeur de $score_{final}$ pour la colonne *col*. Si tous les scores sont nuls, alors le type de la colonne est considéré comme non reconnu. Sinon, l'avantage proportionnel du type de plus haut score par rapport au type de deuxième meilleur score est calculé.

Soit *best* le type numérique de plus haut score pour la colonne *col*, et *secondBest* le type numérique de deuxième plus haut score pour la colonne *col*, l'avantage proportionnel de *best* est calculé selon la formule suivante :

$$avantage(best, col) = \frac{score_{final}(col, best) - score_{final}(col, secondBest)}{score_{final}(col, best)} \quad (4.4)$$

Si l'avantage proportionnel du meilleur type pour la colonne est supérieur à un certain seuil défini par l'utilisateur, alors la colonne *col* est reconnue comme étant du type *best*. Sinon le type de la colonne *col* est considéré comme non reconnu.

Exemple 4.5 *Reprenons la colonne ayant pour titre « Samples tested » dans le tableau 4.1. Le type numérique ayant le plus haut score final pour cette colonne est "Samples tested", avec un score final de 1. Le type numérique de deuxième plus haut score final est "Samples positive", avec un score de 0,583. L'avantage proportionnel de "Samples tested" est donc de* $\frac{1-0,583}{1} = 0,417$*. Si le seuil défini par l'utilisateur est par exemple de 10%, alors la colonne est reconnue comme étant du type "Samples tested".*

4.2.5 Résultats expérimentaux

Nous réutilisons les 262 colonnes ayant correctement été reconnues comme numériques en section 4.1.2. Pour le calcul du score des différents types numériques pour une colonne numérique d'après le titre de la colonne, nous choisissons comme mesure de similarité entre termes du web et termes de l'ontologie la mesure de similarité par cosinus (formule 3.3).

Sur 262 colonnes numériques, 246 ont été correctement reconnues grâce à notre système, 15 ont été considérées comme de type non reconnu, et une seule a été annotée avec le mauvais type numérique. On obtient donc une précision de 99,6% et une couverture de 93,9%, ce que nous jugeons comme un résultat plutôt satisfaisant.

4.3 Reconnaissance du type d'une colonne symbolique

Comme nous l'avons fait pour les colonnes numériques, nous proposons une méthode pour identifier le type d'une colonne reconnue comme symbolique, en fonction des types symboliques définis dans l'ontologie.

Le score final d'un type symbolique pour une colonne symbolique, présenté en section 4.3.2, est une combinaison entre deux scores :

- le score du type pour la colonne d'après le titre de la colonne, calculé de la même manière que pour les colonnes numériques (voir section 4.2.2) ;
- le score du type pour la colonne d'après le contenu des cellules de la colonne, que nous présentons en section 4.3.1.

Le calcul du score final de chaque type symbolique pour une colonne symbolique nous permet de déterminer le type de la colonne (section 4.3.2). Nous présentons nos résultats expérimentaux en section 4.3.3.

4.3.1 Score d'après le contenu de la colonne

Une fois qu'une colonne est reconnue comme étant de type symbolique, nous proposons de calculer les similarités entre les termes contenus dans les cellules de la colonne et chacun des termes de l'ontologie (comme présenté dans le chapitre 3), ceci afin de déterminer à quel type symbolique de l'ontologie correspond chacun des termes dans la colonne.

Soient *col* une colonne symbolique d'un tableau et *type* un type symbolique défini dans l'ontologie. Soit T_{type} l'ensemble de tous les termes de la taxonomie représentant les valeurs possibles du type *type*. Soit t_{ext} le terme correspondant au contenu d'une cellule de la colonne *col*. Alors le score du type *type* pour le terme t_{ext} est la somme des scores de similarité entre le terme t_{ext} et chacun des termes de la taxonomie du type *type* :

$$score(t_{ext}, type) = \sum_{t \in T_{type}} sim(t_{ext}, t) \tag{4.5}$$

Exemple 4.6 *Dans le tableau 4.3, nous considérons la colonne symbolique ayant pour titre « Food ». La première cellule contient le terme « Canned foods "Neutral" ». Ce terme a une mesure de similarité par cosinus non nulle avec les termes de la hiérarchie du type "Food products" suivants :* Baby foods *(similarité de 0,258),* Deep frozen foods *(similarité de 0,236),* Hospital food *(similarité de 0,516),* Food products *(similarité de 0,516) et* Rice baby food *(similarité de 0,192). Le score du type "Food products" pour le terme du web « Canned foods "Neutral" » est de* $0,258 + 0,236 + 0,516 + 0,516 + 0,192 = 1,718$. *Aucun des termes de la hiérarchie du type "Microorganism" n'a un score de similarité par cosinus non nul avec le terme « Canned foods "Neutral" », donc le score de ce*

type pour ce terme est de 0 (idem pour le troisième type symbolique de l'ontologie, "Response").

En ce qui concerne la deuxième cellule de la colonne, elle contient le terme « Canned foods "Acid" ». Ce terme a une mesure de similarité par cosinus non nulle avec les mêmes termes de la hiérarchie du type "Food products" que pour le terme « Canned foods "Neutral" », et les mesures de similarité sont les mêmes : le score du type "Food products" pour le terme du web « Canned foods "Acid" » est donc de $1,718$*. En revanche, plusieurs termes de la hiérarchie du type "Microorganism" ont également un score de similarité par cosinus non nul avec ce terme :* Lactic acid bacteria *(score de 0,333),* Lactic acid bacteria *(score de 0,333) et* Acidophilic lactic acid microorganisms *(score de 0,289). Le score du type "Microorganism" pour le terme du web « Canned foods "Acid" » est donc de* $0,333 + 0,333 + 0,289 = 0,955$*. Le type "Response" a un score de 0 pour le terme « Canned foods "Acid" ».*

Food	*Eh(mV)*	*pH*
Canned foods "Neutral"	-130 to -550	>4.4
Canned foods "Acid"	-410 to -550	<4.4

TAB. 4.3 – Extrait d'un tableau : Redox potentials on some foods (issu de [U.S. Food and Drug Administration, 2001])

Afin de déterminer le type symbolique correspondant au terme t_{ext}, nous proposons la règle suivante :

- si tous les scores des types symboliques de l'ontologie pour le terme t_{ext} sont nuls, le type de t_{ext} est considéré comme non reconnu ;
- sinon, l'avantage proportionnel du type de plus haut score par rapport au type suivant est calculé, comme cela a été fait lors du choix d'un type numérique pour une colonne numérique (formule 4.4) :
  - si l'avantage proportionnel du meilleur type est supérieur à un seuil défini par l'utilisateur, alors le type du terme t_{ext} est ce meilleur type ;
  - sinon, le type du terme t_{ext} est considéré comme non reconnu.

Exemple 4.7 *Continuons l'exemple 4.6. Pour le terme « Canned foods "Neutral" », le type symbolique ayant le plus haut score est "Food products", avec un score de* $1,718$*. Le deuxième plus haut score est de 0. L'avantage proportionnel de "Food products" est donc de* $\frac{1,718-0}{1,718} = 1$*. De même, pour le terme « Canned foods "Acid" », le type symbolique ayant le plus haut score est "Food products", avec un score de* $1,718$*. Le deuxième plus haut score est obtenu pour le type "Microorganism", avec un score de 0,955. L'avantage proportionnel de "Food products" est donc de* $\frac{1,718-0,955}{1,718} = 0,444$*. Si le seuil choisi est par exemple de 10%, alors les deux cellules sont reconnues comme étant du type "Food products".*

Ayant déterminé le type de chaque terme de la colonne *col*, nous pouvons à présent calculer le score d'un type symbolique pour la colonne *col*. Le score du type *type* pour la colonne *col* d'après le contenu de la colonne est calculé à partir de la proportion de termes de la colonne ayant le type *type*.

Soient T_{col} l'ensemble de tous les termes de la colonne et T_{col}^{type} l'ensemble de tous les termes de la colonne ayant le type *type*, alors le score du type *type* pour la colonne *col* d'après le contenu de la colonne est :

$$score_{contenu}(col, type) = \frac{\mid T_{col}^{type} \mid}{\mid T_{col} \mid} \tag{4.6}$$

Exemple 4.8 *Continuons l'exemple portant sur le tableau 4.3. Les deux cellules de la colonne ayant pour titre « Food » ont été reconnues comme étant du type "Food products". Le score du type "Food products" d'après le contenu de la colonne est donc* $2/2 = 1$*, tandis que les scores des types "Microorganism" et "Response" sont de* $0/2 = 0$*.*

4.3.2 Score final d'un type symbolique pour une colonne symbolique

Comme pour les colonnes numériques, le score final d'un type symbolique pour une colonne symbolique est une combinaison du score obtenu en considérant le titre de la colonne et du score obtenu en considérant le contenu de la colonne.

Le score final du type symbolique *type* pour la colonne symbolique *col* est calculé de la même manière que pour les colonnes numériques, mais sans filtrage par valeur :

$$score_{final}(col, type) = 1 - (1 - score_{titre}(col, type)) \times (1 - score_{contenu}(col, type)) \tag{4.7}$$

Le choix du type symbolique pour une colonne symbolique est fait de la même manière que pour une colonne numérique (cf. section 4.2.4), c'est-à-dire que l'avantage proportionnel du type ayant le meilleur score pour la colonne est calculé : si cet avantage est supérieur à un certain seuil défini par l'utilisateur, alors la colonne est considérée comme étant du type de meilleur score ; sinon la colonne est considérée comme de type non reconnu.

Exemple 4.9 *Le terme* Food*, titre de la colonne symbolique du tableau 4.3, a un score de similarité par cosinus de 0,577 avec "Food products". Le score final du type "Food products" pour la colonne symbolique du tableau 4.3 est donc de* $1 - (1 - 0,577) \times (1 - 1) = 1$*. Pour les types 'Microorganism" et "Response", les deux scores (d'après le contenu de la colonne et d'après le titre de la colonne) sont nuls, donc le score final est également nul. L'avantage proportionnel du type "Food products" pour la colonne est donc de 1. C'est le type "Food products" qui est choisi pour annoter la colonne.*

4.3.3 Résultats expérimentaux

La méthode de classification des colonnes symboliques que nous proposons est non supervisée, elle utilise seulement les connaissances préalablement définies dans l'ontologie. Il existe cependant un très grand nombre d'algorithmes dédiés à la classification par apprentissage de données textuelles (nous avons ici des colonnes symboliques, chaque colonne peut donc être considérée comme un sac de mots). Nous avons voulu comparer les résultats de notre méthode avec une méthode de classification par apprentissage.

Nous avons choisi d'utiliser la méthode SMO proposée par [Platt, 1999] et implémentée dans Weka [Frank et al., 2006]. Cette méthode est une optimisation de la méthode des SVM, actuellement la plus populaire en classification par apprentissage : cela nous permet donc de comparer notre méthode à une méthode considérée comme représentative de l'état de l'art. Pour utiliser cette méthode, nous caractérisons les colonnes par un ensemble d'attributs : chaque mot lemmatisé présent dans une colonne (y compris le titre de la colonne) donne un attribut ; la valeur de cet attribut (comprise entre 0 et 1) est la fréquence du mot dans la colonne.

Nous avons utilisé pour cette expérience les 81 colonnes qui avaient été correctement reconnues comme étant symboliques (section 4.1.2). Ces colonnes ont été classées manuellement comme étant du type "Microorganism", "Food Products", "Response" ou de type non reconnu (type non présent dans l'ontologie, et trop spécifique pour qu'on souhaite le rajouter dans l'ontologie). Nous avons ensuite classifié ces colonnes de façon automatique avec notre méthode, ainsi qu'avec la méthode SMO, en utilisant l'évaluation croisée par *leave one out* (chaque colonne est classifiée en utilisant un classifieur entraîné avec l'ensemble des 80 autres colonnes).

	utilisation de l'ontologie				SMO			
prédit / manuel	food	micro.	respo.	autre	food	micro.	respo.	autre
food	34	0	0	12	45	0	0	1
micro.	0	16	0	0	4	12	0	0
response	0	0	1	0	0	0	0	1
autre	3	3	0	12	7	0	0	11

TAB. 4.4 – Résultats de classification sur 81 colonnes symboliques.

Sur l'ensemble des 3 types symboliques de l'ontologie, notre méthode donne une précision de 89% et une couverture de 81%. Avec la méthode SMO, la précision est de 84% et la couverture de 90%. Notre méthode, qui utilise des connaissances du domaine mais ne nécessite pas d'apprentissage, offre une précision légèrement meilleure au détriment de la couverture. Ceci s'explique en

partie par le fait que nous avons choisi de gérer le doute de façon restrictive (lorsque le type de la colonne n'est pas sûr, il n'est simplement pas reconnu). Comparés à la méthode SMO, représentative de l'état de l'art en classification par apprentissage, nous pouvons considérer les résultats de notre méthode comme de bons résultats, puisque notre méthode ne nécessite pas d'apprentissage. Cela nous paraît important dans le sens où nous travaillons dans un domaine où l'information est rare, et où il est donc difficile de fournir beaucoup d'exemples à des méthodes d'apprentissage.

Conclusion du chapitre

Nous avons proposé dans ce chapitre une méthode pour l'annotation des colonnes d'un tableau permettant de trouver quel est le type d'une colonne en combinant différents scores calculés en utilisant à la fois le contenu des colonnes et leur titre. Cette méthode d'annotation des colonnes d'un tableau a été publiée dans [Hignette et al., 2007a] et [Hignette, 2007] pour la communauté de recherche en informatique, et dans [Hignette et al., 2007d] pour la communauté de recherche en microbiologie alimentaire.

Notre méthode d'annotation des colonnes, testée sur le domaine de la microbiologie alimentaire, est facilement adaptable à d'autres domaines : elle utilise les connaissances du domaine formalisées dans l'ontologie, et un simple changement d'ontologie permet de changer de domaine d'application. Notre méthode de combinaison de deux scores pour obtenir un résultat plus robuste est facilement extensible à l'utilisation d'un nombre plus important de scores calculés à partir de critères différents, éventuellement adaptables en fonction du domaine d'application.

Une fois que les types des différentes colonnes d'un tableau ont été reconnus, nous les utilisons pour reconnaître quelles sont les relations représentées par le tableau. L'annotation des relations dans un tableau fait l'objet du chapitre suivant.

Chapitre 5

Annotation des relations représentées par le tableau

Ce chapitre est consacré à l'annotation des relations dans le tableau. Nous commençons par reconnaître quelles sont les relations de l'ontologie les plus susceptibles de représenter la sémantique du tableau, grâce à une approche multi-critères utilisant à la fois les types de colonnes, reconnus comme indiqué dans le chapitre 4, et le titre du tableau. Cette reconnaissance des relations est présentée en section 5.1. Une fois les différentes relations reconnues, il reste à instancier ces relations pour chacune des lignes du tableau. Il s'agit, à chaque ligne du tableau, pour chacun des types de la signature de chaque relation reconnue, c'est-à-dire pour les différents types d'accès ainsi que pour le type résultat de chaque relation, de retrouver la valeur correspondante dans le tableau. Cependant, la valeur à utiliser pour l'instanciation n'est souvent pas unique et sûre. En effet, les données numériques sont souvent représentées sous forme d'intervalles ou assorties d'un écart-type et les données symboliques ne correspondent pas exactement aux termes présents dans notre ontologie. Nous avons choisi de représenter ces valeurs en utilisant le formalisme des sous-ensembles flous. Ce choix permet de conserver une cohérence avec l'application MIEL préexistante, dans laquelle les données imprécises ainsi que les préférences dans les critères de sélection des utilisateurs sont représentées sous forme de sous-ensembles flous. Nous présentons d'abord ce que sont les sous-ensembles flous dans la section 5.2. Nous présentons ensuite l'instanciation des différents types de la signature des relations représentées par le tableau, qui est faite différemment s'il s'agit d'un type numérique (section 5.3) ou d'un type symbolique (section 5.4).

5.1 Reconnaissance des relations représentées par le tableau

Comme pour la reconnaissance du type des colonnes du tableau, la reconnaissance des relations représentées par le tableau fait appel au calcul d'un score de chaque relation de l'ontologie pour le tableau. Le score final d'une relation pour le tableau, présenté en section 5.1.2, est une combinaison entre deux scores :

- le score de la relation pour le tableau d'après le titre du tableau : il s'agit du score de similarité entre le titre du tableau et le nom de la relation, calculé de la même manière que pour les scores des types pour les colonnes d'après le titre de la colonne (voir section 4.2.2) ;
- le score de la relation pour le tableau d'après la signature du tableau, que nous présentons en section 5.1.1.

5.1.1 Score d'une relation pour le tableau d'après la signature du tableau

Pour chaque relation de l'ontologie, nous calculons un score basé sur la comparaison entre la signature de la relation définie dans l'ontologie (voir chapitre 1) et la signature du tableau (types des colonnes, tels que reconnus par la méthode présentée au chapitre 4).

Soit tab un tableau de données, avec $sign(tab)$ l'ensemble des types reconnus pour les colonnes de tab. Soit rel une relation de l'ontologie, $result(rel)$ son type résultat et $acces(rel)$ l'ensemble de ses types d'accès. Alors le score de la relation rel pour le tableau tab d'après la signature du tableau, noté $score_{sign}(rel, tab)$ est calculé comme suit :

- si le type résultat de la relation rel n'a pas été reconnu parmi les colonnes du tableau, c'est-à-dire si $result(rel) \notin sign(tab)$, alors $score_{sign}(rel, tab) = 0$;
- si $result(rel) \in sign(tab)$, alors le score de la relation rel pour le tableau tab est la proportion des types de la signature de la relation rel qui ont été reconnus dans le tableau tab :

$$score_{sign}(rel, tab) = \frac{\mid (acces(rel) \cup \{result(rel)\}) \cap sign(tab) \mid}{\mid acces(rel) \cup \{result(rel)\} \mid} \quad (5.1)$$

Exemple 5.1 *Les étapes précédentes de l'annotation ont permis de reconnaître les types de colonnes "Food Product" et "AW water activity" dans le tableau présenté en figure 5.1.*

Il existe deux relations dans l'ontologie qui ont un score non nul pour le tableau d'après la signature du tableau :

- *la relation "Growth parameter - AW" : un seul type de la signature reconnu sur deux, donc le score est de 0.5 ;*
- *la relation "Product property - AW" : les deux types de la signature sont reconnus, donc le score est de 1.*

Animal products	$\mathbf{a}_w$
fresh meat, poultry, fish	0.99 - 1.00
natural cheeses	0.95 - 1.00

TAB. 5.1 – Extrait d'un tableau : Approximate a_w values of selected food categories (issu de [U.S. Food and Drug Administration, 2001])

5.1.2 Score final d'une relation pour un tableau

Le score final d'une relation pour le tableau est une combinaison du score obtenu en considérant le titre du tableau et du score obtenu en considérant la signature du tableau.

Soit *rel* une relation de l'ontologie et *tab* un tableau, le score final de la relation *rel* pour le tableau *tab*, noté $score_{final}(rel, tab)$, est donné par :

- si $score_{sign}(rel, tab) = 0$, i.e. si le type résultat de la relation n'a pas été trouvé parmi les colonnes du tableau, alors $score_{final}(rel, tab) = 0$.
- si $score_{sign}(rel, tab) > 0$, alors

$$score_{final}(rel, tab) = 1 - (1 - score_{titre}(rel, tab)) \times (1 - score_{sign}(rel, tab)) \quad (5.2)$$

Lorsque le score de chaque relation de l'ontologie pour le tableau a été calculé, il reste à décider quelles relations sont conservées pour annoter le tableau. En effet, il ne s'agit pas simplement de conserver la relation de meilleur score : une relation sémantique de l'ontologie ne représente qu'un seul type de résultat expérimental, un tableau peut donc représenter plusieurs relations sémantiques à la fois (c'est d'ailleurs le cas le plus fréquent dans le jeu de tableaux que nous avons utilisé pour expérimenter notre méthode d'annotation). On considère par contre que, pour un type donné, le tableau ne peut représenter qu'une seule des relations ayant ce type comme résultat.

Pour annoter le tableau, seules les relations ayant un score final pour le tableau non nul sont considérées. Parmi ces relations, si plusieurs ont le même type résultat, seule celle qui a le meilleur score final est conservée (cependant, si plusieurs relations ayant le même type résultat ont le même score, alors toutes ces relations sont conservées pour l'annotation du tableau).

Exemple 5.2 *Reprenons le tableau présenté en figure 5.1. Les seules relations ayant un score selon le titre du tableau non nul sont "Growth parameter - AW" et "Product property - AW" (grâce au mot a_w dans le titre du tableau). En considérant pour chaque mot un poids de 1 (sauf sur "of" dans le titre du tableau qui est un stopword), un score de similarité entre le titre du tableau et le nom de la relation de $\frac{1}{\sqrt{6 \times 3}} \approx 0,236$ est obtenu pour chacune de ces deux relations.*

Pour ces deux relations, le score selon le titre du tableau et le score selon la signature du tableau sont combinés pour calculer le score final :

- $score_{final}(\text{“}Growth\ parameter - AW\text{”}, tableau\ 5.1) \approx 1 - (1 - 0,236) \times (1 - 0,5) \approx 0,618$
- $score_{final}(\text{“}Product\ property - AW\text{”}, tableau\ 5.1) \approx 1 - (1 - 0,236) \times (1 - 1) = 1$

Les deux relations “Growth parameter - AW” et “Product property - AW” étant concurrentes (elles ont toutes les deux “AW water activity” comme type résultat), seule la relation ayant le meilleur score final est conservée, c'est-à-dire “Product property - AW”.

5.1.3 Résultats expérimentaux

Les 60 tableaux de données en microbiologie alimentaire, déjà utilisés pour évaluer la pertinence de notre approche pour l'annotation des colonnes (voir le chapitre 4), ont été réutilisés pour évaluer l'annotation des relations. Ces 60 tableaux ont été manuellement annotés à l'aide des 16 relations sémantiques définies dans l'ontologie : chaque tableau représentait entre 1 et 5 relations de l'ontologie, ce qui donne au total 123 relations avec l'annotation manuelle.

Ces tableaux ont ensuite été annotés en enchaînant toutes les étapes de notre méthode sans validation intermédiaire : reconnaissance numérique/symbolique, annotation des cellules des colonnes symboliques, reconnaissance du type des colonnes, puis reconnaissance des relations représentées par le tableau. Sur les 123 relations de l'annotation manuelle, 119 ont été correctement reconnues par notre processus automatique, 4 n'ont pas été reconnues, et 30 ont été reconnues alors qu'elles ne figuraient pas dans l'annotation manuelle. Notre méthode d'annotation des relations dans un tableau donne donc une précision de 80% pour une couverture de 97%.

Nous avons étudié l'évolution de la précision et de la couverture si un seuil de score est appliqué, c'est-à-dire si seules les relations ayant un score final avec le tableau supérieur à un seuil θ_{rel} sont considérées pour l'annotation des tableaux. La figure 5.1 présente l'évolution de la précision, de la couverture et de la mesure F1 en fonction de la valeur choisie pour le seuil θ_{rel}.

Soit $relations\_manuelles$ le nombre total de relations dans l'annotation manuelle, $vrais\_positifs(\theta_{rel})$ le nombre de relations dont le score est supérieur à θ_{rel}, qui correspondent effectivement aux relations choisies dans l'annotation manuelle et $faux\_positifs(\theta_{rel})$ le nombre de relations dont le score est supérieur à θ_{rel} mais qui ne sont pas des relations représentées par le tableau d'après l'annotation manuelle. Alors la précision, la couverture et la mesure F1 pour le seuil θ_{rel} sont définies comme suit :

$precision(\theta_{rel}) = \frac{vrais\_positifs(\theta_{rel})}{vrais\_positifs(\theta_{rel}) + faux\_positifs(\theta_{rel})}$

$couverture(\theta_{rel}) = \frac{vrais\_positifs(\theta_{rel})}{relations\_manuelles}$

$F1(\theta_{rel}) = \frac{2 \times precision(\theta_{rel}) \times couverture(\theta_{rel})}{precision(\theta_{rel}) + couverture(\theta_{rel})}$

Au seuil $\theta_{rel} = 0,5$, on obtient une inversion entre précision et couverture,

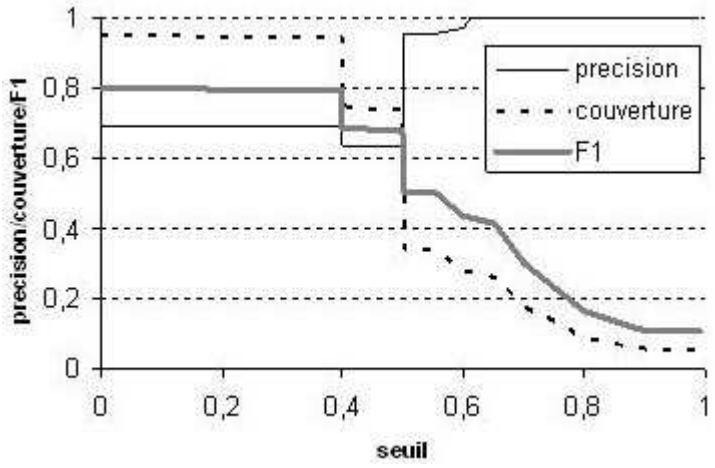

FIG. 5.1 – Precision, couverture et mesure F1 en fonction du seuil θ_{rel}.

avec une précision de 96% pour une couverture de 76%. Le saut pour un seuil de 0,5 s'explique essentiellement par les relations binaires de l'ontologie, ayant un même type résultat et un type d'accès différent (*Growth parameter-AW* et *Product property-AW*, ainsi que *Growth parameter-pH* et *Product property-pH*). En effet, quand on reconnaît dans le tableau uniquement le type résultat de ces relations (score selon le titre du tableau de 0, score selon la signature du tableau de 0,5), soit les deux relations ayant le même type résultat sont utilisées dans l'annotation (quand $\theta_{rel} < 0,5$), l'une des relations étant juste et l'autre fausse, d'où une faible précision mais une bonne couverture, soit aucune des deux relations n'est utilisée (quand $\theta_{rel} \geq 0,5$), d'où une meilleure précision mais une baisse de couverture.

5.2 Les sous-ensembles flous

Comme nous l'avons annoncé en introduction de ce chapitre, les valeurs que nous instancions pour les différents types de la signature des relations reconnues dans le tableau ne sont pas toujours uniques et précises. Nous avons choisi de représenter ces valeurs sous la forme de sous-ensembles flous : cette section présente donc les bases théoriques sur lesquelles nous appuyons notre annotation.

La section 5.2.1 présente une définition des sous-ensembles flous. Les différentes sémantiques rattachées aux sous-ensembles flous sont discutées en section 5.2.2.

5.2.1 Définition d'un sous-ensemble flou

La notion de sous-ensemble flou, apportée par [Zadeh, 1965], est un assouplissement de la notion de sous-ensemble classique d'un ensemble de référence

X. Dans le cas classique, les éléments de X qui possèdent une certaine propriété constituent un sous-ensemble A de X, les éléments de X qui ne possèdent pas cette propriété appartiennent au complémentaire de A dans X. Dans le cas d'un sous-ensemble flou, les éléments peuvent appartenir partiellement à un sous-ensemble, avec un degré d'appartenance compris entre 0 (élément n'appartenant pas au sous-ensemble) et 1 (élément appartenant totalement au sous-ensemble).

Définition 5.1 *Un* sous-ensemble flou *A d'un ensemble de référence X est défini par une fonction d'appartenance μ_A de X dans $[0,1]$ qui associe à chaque élément x de X le degré $\mu_A(x)$ avec lequel x appartient à A.*

Le support *du sous-ensemble flou A est l'ensemble des éléments x de l'ensemble de référence X pour lesquels $\mu_A(x) > 0$. Le* noyau *du sous-ensemble flou A est l'ensemble des éléments x de l'ensemble de référence X pour lesquels $\mu_A(x) = 1$. Un sous-ensemble flou est dit* normalisé *si son noyau est non vide.*

On distingue deux types de sous-ensembles flous, selon qu'ils sont exprimés sur un ensemble de référence à valeurs discrètes ou à valeurs continues. Pour les sous-ensembles flous sur un ensemble de référence à valeurs discrètes, la valeur de la fonction d'appartenance est donnée pour tous les éléments du support de l'ensemble flou. Pour les sous-ensembles flous sur un ensemble de référence à valeurs continues, on se limitera dans ce travail à des distributions sous forme de trapèzes, définies par 4 valeurs : les bornes du support et les bornes du noyau.

Définition 5.2 *Un sous-ensemble flou continu sur $\Re$ de forme trapèze, ayant comme support $sup = [min_{sup}, max_{sup}]$ et comme noyau $ker = [min_{ker}, max_{ker}]$, est défini comme le sous-ensemble flou sur $\Re$ ayant pour fonction d'appartenance la fonction μ telle que, pour tout $x \in \Re$:*

- *si $x \leq min_{sup}$ ou $x \geq max_{sup}$ alors $\mu(x) = 0$;*
- *si $min_{ker} \leq x \leq max_{ker}$ alors $\mu(x) = 1$;*
- *si $min_{sup} \leq x \leq min_{ker}$ alors $\mu(x) = \frac{x - min_{sup}}{min_{ker} - min_{sup}}$;*
- *si $max_{ker} \leq x \leq max_{sup}$ alors $\mu(x) = \frac{x - max_{ker}}{max_{ker} - max_{sup}}$;*

Exemple 5.3 *La figure 5.2 présente deux sous-ensembles flous, le premier défini sur un ensemble de référence à valeurs discrètes, le second sur un ensemble de référence à valeurs continues. La sémantique de ces sous-ensembles flous sera explicitée dans la section suivante. Pour clarifier les choses d'un point de vue applicatif, nous donnons des définitions correspondant aux trois types de laits présentés dans le sous-ensemble flou sur un ensemble de référence à valeurs discrètes :*

- *le lait UHT est un lait qui a été chauffé entre 140 et 150°C, pendant 2 à 5 secondes : la très haute température tue tous les microorganismes, ce qui permet d'obtenir une durée de conservation longue. Bien que le chauffage se fasse sur une durée très courte, le lait prend un léger « goût de cuit ».*

– *le lait pasteurisé est un lait qui est chauffé à 72°C pendant 15 secondes, puis brutalement refroidi. C'est le choc thermique lors du refroidissement qui tue la plupart des microorganismes ; cependant certains pathogènes, notamment sous formes de spores, y survivent, d'où la durée de conservation plus courte que pour les traitements UHT. Comme le lait est relativement peu chauffé, il a moins le « goût de cuit » que le lait UHT.*
– *le lait microfiltré est un lait qui a été filtré à travers une membrane, dont les pores sont assez grands pour laisser passer les protéines du lait, mais pas assez pour laisser passer la plupart des microorganismes. Cependant certains microorganismes de petite taille parviennent à passer, d'où une durée de conservation courte. Cette méthode a l'avantage de conserver intact le goût du lait cru.*

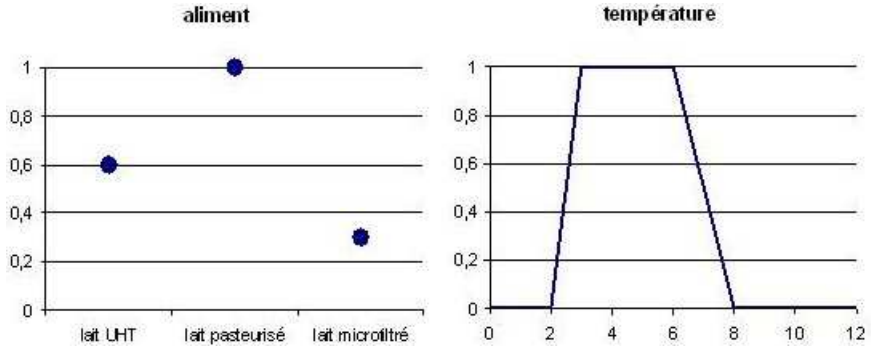

FIG. 5.2 – Deux sous-ensembles flous

5.2.2 Utilisation des sous-ensembles flous

Les trois sémantiques associées aux sous-ensembles flous comme présentées dans [Dubois & Prade, 1997] sont :

– un sous-ensemble flou peut représenter des préférences : les éléments pour lesquels la fonction d'appartenance au sous-ensemble flou est la plus élevée sont les éléments préférés ;
– un sous-ensemble flou peut représenter de l'incertitude ou de l'imprécision : il existe une "vraie" valeur pour une variable y, mais celle-ci n'étant pas connue, elle est représentée par un sous-ensemble flou sur le domaine de définition de y qui donne une disjonction de valeurs possibles pour y. Plus la fonction d'appartenance au sous-ensemble flou est élevée pour un élément, plus cet élément a de "chance" d'être la véritable valeur de y ;
– un sous-ensemble flou peut représenter des similarités : un nouvel objet y est représenté par un sous-ensemble flou sur l'ensemble de référence X des objets connus. Plus la fonction d'appartenance d'un élément x de X est élevée, plus l'objet y est semblable à l'objet x.

Exemple 5.4 *Les sous-ensembles flous de la figure 5.2 peuvent avoir les sémantiques suivantes :*

- *en terme de préférences : un utilisateur d'un moteur de requêtes veut des informations sur la conservation du lait pasteurisé, mais n'exclut pas que des résultats sur le lait UHT ou dans une moindre mesure le lait microfiltré puissent l'intéresser ; il s'intéresse aux températures de conservation entre* 3 *et* 6 *degrés, mais n'exclut pas que des données aux marges (entre* 2 *et* 8 *degrés) puissent l'intéresser ;*
- *en terme d'imprécision : on veut renseigner une base de données, capable de répondre à la requête de l'utilisateur précitée. Un document donne des informations sur la conservation de "lait stérilisé" à "température réfrigérée". Il y a de grandes chances pour que le lait stérilisé soit en fait du lait pasteurisé ; cependant le traitement UHT et la microfiltration sont également des techniques de stérilisation : il est donc possible que ce soit à ces techniques que le document fasse référence, d'où la fonction d'appartenance au sous-ensemble flou non nulle pour les termes "lait UHT" et "lait microfiltré". De même, la réfrigération consiste généralement en un maintien de la température entre* 3 *et* 6 *degrés, mais il est également possible que des températures variant entre* 2 *et* 8 *degrés soient considérées comme de la réfrigération.*
- *en terme de similarité : un document peut faire référence à du lait stérilisé par une technique de stérilisation non répertoriée dans l'ontologie, permettant d'obtenir une conservation longue mais sans « goût de cuit ». Cette technique donne donc des résultats similaires à la pasteurisation et assez similaires au traitement UHT ; elle est moins proche des techniques de microfiltration, bien qu'ayant en commun l'absence de « goût de cuit ». Le sous-ensemble flou représentant des températures n'a en revanche pas de signification en termes de similarité : des similarités peuvent être mesurées sur des objets distincts, mais il n'y a pas de sémantique de la similarité sur un domaine continu.*

Nous utilisons chacune des trois sémantiques que nous venons de décrire :

- la sémantique de préférences est utilisée dans le système MIEL++ pour l'expression des critères de sélection d'un utilisateur lors de l'interrogation ;
- la sémantique d'imprécision est utilisée pour représenter les données numériques. Nous introduisons en outre en section 5.3.1 une autre sémantique propre aux données numériques, la sémantique d'optimalité ;
- la sémantique de similarité est utilisée pour représenter les données symboliques.

5.3 Instanciation des valeurs numériques

Lorsque les relations représentées par un tableau ont été reconnues, il reste à les instancier pour chacune des lignes du tableau. Pour instancier une relation sur une ligne du tableau, nous instancions chacun des types de la signature de la relation pour cette ligne. L'instanciation est différente suivant qu'il s'agit d'un type numérique ou d'un type symbolique. Cette section traite de l'instanciation des types numériques.

Il arrive que l'on trouve, pour une seule ligne d'un tableau, plusieurs valeurs numériques correspondant à un même type numérique que l'on veut instancier. Les différentes sémantiques liées à ces valeurs multiples sont expliquées en section 5.3.1. Dans notre méthode d'annotation, nous commençons par extraire les valeurs numériques correspondant au type numérique à instancier pour la relation (section 5.3.2). Ensuite, les valeurs extraites sont combinées au sein d'un sous-ensemble flou, pour représenter au mieux la signification de ces valeurs dans le tableau d'origine. Nous cherchons tout d'abord à construire un sous-ensemble flou d'optimalité (section 5.3.3). Si aucun sous-ensemble flou d'optimalité ne peut être construit, alors nous cherchons à construire un sous-ensemble flou représentant des données imprécises (section 5.3.4).

La qualité des sous-ensembles flous ainsi obtenus est évaluée en section 5.3.5.

5.3.1 Imprécision et optimalité

Lorsqu'une valeur numérique pour un type numérique est unique, il n'y a pas de problème particulier : le type numérique en question prend cette valeur et c'est tout. Les choses deviennent plus compliquées lorsque plusieurs valeurs numériques sont proposées pour un seul et même type numérique. Outre les éventuelles erreurs lors de la reconnaissance du type numérique d'une colonne, la présence de plusieurs valeurs numériques pour un seul type numérique peut avoir différentes sémantiques : nous expliquons ici les sémantiques d'imprécision et d'optimalité.

Imprécision

La signification la plus courante de la présence de plusieurs valeurs numériques pour une seule donnée est liée à une imprécision des données. Un auteur peut en effet prendre en compte soit une imprécision de mesure soit une variabilité intrinsèque aux données : le tableau présente alors, plutôt qu'une valeur unique, un intervalle de valeurs ou bien une moyenne assortie d'un écart-type. Dans le système MIEL préexistant, ces données imprécises sont représentées sous la forme de sous-ensembles flous. Nous avons décidé de conserver ce formalisme dans l'extension à l'entrepôt de données, d'autant que les outils de comparaison entre

ces données imprécises et les préférences d'un utilisateur sont bien définis (voir section 5.5.1).

Des données imprécises peuvent également être représentées sous forme de valeurs censurées. Une valeur censurée est une donnée imprécise, mais représentée par une seule valeur numérique. Par exemple, lors d'une catastrophe naturelle, le nombre de morts est donné par les médias comme supérieur à un certain nombre : en effet, il y a eu au moins autant de morts que de cadavres retrouvés, mais la limite supérieure n'est pas connue. Il s'agit d'une valeur censurée à droite, c'est-à-dire que si l'on représente la donnée comme un intervalle, la borne de gauche est connue mais pas celle de droite. L'imprécision dans cet exemple est liée à la méthode de mesure. Il existe également des valeurs censurées dont l'imprécision est liée à des contraintes légales : par exemple, lorsque des études sur les polluants dans l'eau sont publiées pour le grand public, seul le fait que la mesure est inférieure à la limite légale est présenté, mais la valeur exacte de la mesure n'est pas donnée. Il s'agit dans ce cas de valeurs censurées à gauche (la borne de droite de l'intervalle est connue, mais pas celle de gauche). Dans notre application, les valeurs censurées sont essentiellement liées aux limites des méthodes de mesure. Il s'agit la plupart du temps de la concentration en microorganismes, qui échappe soit aux limites de détection (aucun microorganisme n'est détecté, ce qui signifie que la concentration est inférieure au seuil de détection de la méthode de mesure), soit aux limites de dénombrement (la méthode ne permet pas d'évaluer la concentration en microorganismes, soit parce qu'elle est trop faible, soit parce qu'elle est trop importante).

Les données imprécises liées à des valeurs censurées, comme les données imprécises représentées par des valeurs numériques multiples, sont représentées dans notre système d'annotation sous forme de sous-ensembles flous, et plus exactement sous forme de distributions de possibilités. La construction de ces sous-ensembles flous est expliquée en section 5.3.4. Cette section présentera également la façon dont sont représentées les données précises, qui sont considérées comme un cas particulier de données imprécises.

Optimalité

Les données imprécises peuvent être représentées par l'auteur d'un tableau sous la forme de valeurs numériques multiples pour un seul type de données (moyenne et écart type ou intervalle de valeurs). L'existence de plusieurs valeurs numériques pour un seul type numérique peut cependant avoir une autre signification : il est possible que la donnée soit intrinsèquement multi-valuée. Ce cas de figure est rencontré lorsque la donnée présente des valeurs minimales, maximales et parfois une plage optimale. A ne pas confondre avec une imprécision des données, les valeurs minimales et maximales traduisent des conditions extrêmes de fonctionnement, les valeurs optimales de bonnes conditions. Par exemple, une bouilloire à résistance chauffante possède une cuve sur laquelle sont indiqués

les remplissages minimum (pour que l'eau recouvre entièrement la résistance) et maximum (pour que l'eau une fois à ébullition ne déborde pas) ; le fonctionnement optimal (faire chauffer un maximum d'eau en un minimum de temps) se situe quelque part entre ces deux bornes. De même, le nombre de personnes constituant l'équipage d'un navire aura une valeur minimale (nombre de personnes nécessaires pour pouvoir physiquement effectuer toutes les manœuvres), une valeur maximale (nombre de personnes pouvant monter à bord), et une valeur optimale (nombre de personnes nécessaires pour que chacun ait une tâche bien définie et ne soit ni sous-employé ni débordé par le travail). Dans notre application en microbiologie alimentaire, les données concernées sont les valeurs cardinales de croissance des microorganismes. En effet, un microorganisme a une température minimale de croissance (en dessous de laquelle le microorganisme ne se développe pas), ainsi qu'une température maximale de croissance (au dessus de laquelle il ne peut pas se développer). Le microorganisme ne se développe pas à la même vitesse quelle que soit la température entre ces deux extrêmes : il exite une température optimale (ou un intervalle de températures optimales) à laquelle le microorganisme se développe le plus rapidement. Cette notion de valeurs minimale, optimale et maximale de croissance est valide pour d'autres types numériques tels que le pH ou l'activité de l'eau.

Ces données intrinsèquement multivaluées, qui, comme nous l'avons vu au travers des quelques exemples présentés, existent dans divers domaines d'application, peuvent être modélisées de différentes manières, selon l'utilisation que l'on souhaite en faire au moment de l'interrogation des tableaux annotés. En effet, les valeurs minimale, maximale et optimale peuvent être considérées comme trois types de données distincts, donnant lieu à trois relations différentes : dans le cas de notre application, les trois relations *condition de croissance : température minimale*, *condition de croissance : température optimale* et *condition de croissance : température maximale* seraient construites. Cette représentation, dans le cadre de notre application, est la bonne si le but est de construire des modèles de croissance des microorganismes. En effet, dans ces modèles, les conditions de température minimales, optimales et maximales sont trois paramètres qui jouent des rôles distincts. Dans d'autres cas d'utilisation des données, on peut également vouloir avoir une réponse d'ensemble, qui prenne en compte à la fois les valeurs minimales, maximales et optimales. Pour reprendre l'exemple de la bouilloire, une question possible est par exemple « pour faire bouillir telle quantité d'eau, quel modèle de bouilloire dois-je choisir ? ». Les réponses à cette question devront être ordonnées suivant l'optimalité de cette quantité d'eau pour le fonctionnement de la bouilloire. Dans le cas de notre application en microbiologie, une autre question de ce type peut être « quels sont les microorganismes capables de se développer dans telles conditions ? ». On voudra alors obtenir une réponse qui donnera, en premier lieu les microorganismes dont les conditions optimales de croissance correspondent à la requête, mais également tous les microorganismes qui peuvent se développer, même de façon suboptimale, dans ces conditions. Pour

répondre à ce type de questions, minimum, maximum et optimum peuvent être représentés sous forme d'un sous-ensemble flou. La fonction d'appartenance à ce sous-ensemble flou ne représentera pas le fait qu'une certaine valeur soit plus ou moins *possible*, mais représentera le fait que cette valeur soit plus ou moins *optimale*.

C'est cette deuxième modélisation sous forme de sous-ensemble flou que nous avons retenue pour l'instanciation des types numériques dans les relations, car elle correspond mieux à nos besoins applicatifs. La constuction de sous-ensembles flous d'optimalité sera présentée dans la section 5.3.3.

5.3.2 Extraction des valeurs numériques

Lorsqu'on veut instancier la valeur d'un type numérique *type* pour une relation donnée, sur une ligne l_i du tableau, il existe plusieurs cas de figure :

- une colonne numérique du type *type* a été reconnue dans le tableau. Dans ce cas, il faut récupérer la valeur numérique de cette colonne à la ligne l_i ; cette valeur peut être une valeur isolée, mais également un intervalle, ou une moyenne assortie d'un écart-type...
- plusieurs colonnes numériques du tableau ont été reconnues comme étant du type *type*. Il faut alors non seulement récupérer les différentes valeurs dans ces colonnes à la ligne l_i, mais également savoir de quelle manière il faut les combiner : ces différentes colonnes représentent-elles un minimum et un maximum (éventuellement un optimum), ou alors une moyenne et un écart-type ?
- aucune colonne du tableau n'a été reconnue comme ayant le type *type*. Dans ce cas, il faut chercher l'information ailleurs (constante dans le titre des colonnes ou dans le titre du tableau).

Pour chaque type numérique *type* de la signature de la relation, nous commençons par rechercher, parmi les colonnes numériques dont le type a été reconnu (voir section 4.2), celles ayant le type *type*. Les valeurs numériques sont extraites, en gardant un pointeur vers la cellule d'où elles sont issues. Chaque valeur numérique extraite reçoit deux marqueurs qui permettent de définir la signification de la valeur par rapport aux autres valeurs extraites. Un premier marqueur est lié à la colonne dans laquelle la valeur numérique a été trouvée : ce marqueur permet de déterminer comment combiner les valeurs extraites dans le cas où plusieurs colonnes du type *type* ont été reconnues dans le tableau. Un deuxième marqueur est lié au rôle tenu par la valeur numérique extraite au sein de la cellule d'où elle a été extraite : ce marqueur permet de déterminer comment combiner les valeurs extraites dans le cas où plusieurs valeurs sont extraites à partir d'une même cellule du tableau. Nous présentons ici comment ces deux marqueurs sont construits :

- marqueur de type de colonne. Ce marqueur est identique pour toutes les valeurs numériques issues d'une même colonne. Ce marqueur est lié à la

présence dans le titre de la colonne de mots-clefs prédéfinis, permettant de marquer la colonne comme représentant un minimum, un maximum, un optimum ou un écart-type. Si aucun de ces mots-clefs n'a été trouvé, le marqueur prend la valeur "inconnu" (il n'y a pas de mot-clef particulier, donc pas de marqueur spécifique, pour désigner une colonne représentant une moyenne : comme nous le présenterons en section 5.3.4 une moyenne est reconnue lorsqu'une valeur ayant comme marqueur de type de colonne "inconnu" est accompagnée d'une valeur ayant comme marqueur de type de colonne "écart-type").

- marqueur de rôle dans la cellule. Lorsqu'une cellule contient plusieurs valeurs numériques, leurs rôles respectifs sont recherchés :
  - si la cellule présente un intervalle (deux nombres séparés par un tiret), alors le premier nombre prend le marqueur "minimum d'intervalle" et le deuxième prend le marqueur "maximum d'intervalle" ;
  - si la cellule présente deux nombres séparés par la chaîne de caractères "+-", le premier nombre prend le marqueur "moyenne" et le deuxième nombre prend le marqueur "écart-type" ;
  - s'il n'y a qu'une valeur dans la cellule, ou si le lien entre les valeurs n'est pas reconnu, la (les) valeur(s) extraite(s) de la cellule prend (prennent) le marqueur "isolé".

Exemple 5.5 *Le tableau 5.2 présente un extrait de tableau dans lequel la relation "Paramètre de croissance - pH" a été reconnue. Pour l'instanciation de cette relation sur la première ligne du tableau, les valeurs numériques correspondant au type numérique "PH" sont extraites : grâce à la méthode décrite au chapitre 4, les trois dernières colonnes du tableau ont été identifiées comme étant du type "PH". Les valeurs extraites pour le type "PH" sur la première ligne de données du tableau 5.2 sont donc :*

- *la valeur 5, avec pour marqueur de type de colonne "minimum" (présence du mot-clef "Min" dans le titre de la colonne), et pour marqueur de rôle dans la cellule "isolé" ;*
- *la valeur 6, avec pour marqueur de type de colonne "optimum" (présence du mot-clef "Opt" dans le titre de la colonne), et pour marqueur de rôle dans la cellule "minimum d'intervalle" ;*
- *la valeur 7, avec pour marqueur de type de colonne "optimum" et pour marqueur de rôle dans la cellule "maximum d'intervalle" ;*
- *la valeur 8.8, avec pour marqueur de type de colonne "maximum" (présence du mot-clef "Max" dans le titre de la colonne), et pour marqueur de rôle dans la cellule "isolé".*

Si aucune colonne du type *type* n'a été reconnue dans le tableau, les occurrences de nombres directement suivis d'une unité sont recherchées dans les titres de colonne ou le titre du tableau. Si l'unité est compatible avec le type *type*, la

Species	pH Min	pH Opt.	pH Max
Bacillus cereus	5	6–7	8.8

TAB. 5.2 – Extrait d'un tableau : Ecologic values for some foodborne bacterial pathogens (issu de [Cliver et al., 2004])

valeur numérique est extraite et se voit assigner le marqueur de type de colonne "inconnu" et le marqueur de rôle dans la cellule "isolé".

5.3.3 Construction d'un sous-ensemble flou d'optimalité

Pour un type numérique donné appartenant à la signature de la relation, pour une ligne donnée du tableau, les valeurs numériques extraites sont recensées. Les marqueurs de type de colonne et de rôle dans la cellule que nous avons ajouté aux valeurs numériques extraites permettent de combiner les différentes valeurs numériques entre elles pour former un sous-ensemble flou. Dans cette section sont traitées les valeurs numériques multiples ayant une sémantique d'optimalité (voir section 5.3.1). La sémantique d'optimalité est traitée en premier car c'est la plus facile à reconnaître : si aucune sémantique d'optimalité n'est reconnue, alors les données seront traitées comme représentant de l'imprécision (ce cas est traité dans la section suivante).

Pour chaque marqueur de type de colonne minimum, maximum, optimum ou inconnu, un intervalle est construit selon la méthode suivante : soit m un marqueur de type de colonne,

- s'il existe une seule valeur numérique val_m de marqueur de colonne m, alors l'intervalle $[val_m, val_m]$ est construit ;
- s'il existe deux valeurs numériques min_m et max_m de marqueur de colonne m, ayant respectivement pour marqueurs de rôle dans la cellule "minimum d'intervalle" et "maximum d'intervalle", alors l'inégalité $min_m < max_m$ est examinée : si l'inégalité est vérifiée, l'intervalle $[min_m, max_m]$ est construit, sinon les valeurs min_m et max_m prennent comme marqueur de type de colonne "inconnu" et comme marqueur de rôle dans la cellule "isolé" ;
- s'il existe deux valeurs numériques moy_m et err_m de marqueur de colonne m, ayant respectivement pour marqueurs de rôle dans la cellule "moyenne" et "écart-type", alors l'intervalle $[moy_m - err_m, moy_m + err_m]$ est construit ;
- si aucun des cas précédents n'est valide, alors on considère que l'intervalle ne peut pas être construit, et toutes les valeurs ayant comme marqueur de type de colonne m se voient affecter le marqueur de type de colonne "inconnu".

Une fois les intervalles créés, le nombre de marqueurs de type de colonne parmi minimum, maximum et optimum qui ont pu se voir assigner un intervalle est compté :

- si un seul des trois marqueurs de type de colonne minimum, maximum ou optimum est présent, ce marqueur est remplacé par le marqueur de type de colonne “inconnu” ;
- si les trois marqueurs se sont vus attribuer un intervalle, alors la compatibilité des valeurs est vérifiée. Soient $[inf_{min}, sup_{min}]$, $[inf_{opt}, sup_{opt}]$ et $[inf_{max}, sup_{max}]$ les intervalles construits respectivement pour les marqueurs de type de colonne minimum, optimum et maximum, on vérifie que l'ordre suivant est respecté : $inf_{min} \leq sup_{min} \leq inf_{opt} \leq sup_{opt} \leq inf_{max} \leq sup_{max}$. Si ce n'est pas le cas, les valeurs numériques constituant ces intervalles se voient affecter le marqueur de type de colonne “inconnu” ;
- si seuls deux des marqueurs minimum, maximum ou optimum se sont vus attribuer un intervalle, et si un intervalle de marqueur “inconnu” a également été construit, la compatibilité des valeurs est évaluée comme expliqué dans le point précédent, en assignant provisoirement à l'intervalle de marqueur “inconnu” le marqueur manquant : si les valeurs sont compatibles, l'intervalle construit pour le marqueur “inconnu” se voit attribuer définitivement le marqueur manquant, sinon l'intervalle garde un marqueur “inconnu” et la transformation se poursuit selon le point suivant ;
- si seuls deux des marqueurs minimum, maximum ou optimum sont présents, et qu'aucun intervalle de marqueur “inconnu” n'a pu être construit (ou que cet intervalle n'est pas compatible avec les intervalles des deux autres marqueurs et n'est donc pas considéré), alors l'intervalle manquant est construit de la façon suivante :
  - si c'est le marqueur minimum qui manque, soit $[inf_{opt}, sup_{opt}]$ l'intervalle de marqueur optimum, alors l'intervalle $[inf_{opt}, inf_{opt}]$ est construit pour le marqueur minimum ;
  - si c'est le marqueur optimum qui manque, soient $[inf_{min}, sup_{min}]$ et $[inf_{max}, sup_{max}]$ les intervalles associés respectivement aux marqueurs minimum et maximum, alors l'intervalle $[sup_{min}, inf_{max}]$ est construit pour le marqueur optimum ;
  - si c'est le marqueur maximum qui manque, soit $[inf_{opt}, sup_{opt}]$ l'intervalle de marqueur optimum, alors l'intervalle $[sup_{opt}, sup_{opt}]$ est construit pour le marqueur maximum.

  On vérifie finalement que les valeurs de l'intervalle construit sont bien compatibles : sinon chaque valeur numérique extraite se voit affecter le marqueur “inconnu”.

Les transformations ci-dessus, une fois appliquées, mènent à deux situations possibles : soit il n'y a plus aucune valeur ayant des marqueurs minimum, maximum ou optimum, soit il y a trois intervalles $[inf_{min}, sup_{min}]$, $[inf_{opt}, sup_{opt}]$ et $[inf_{max}, sup_{max}]$ respectivement marqués minimum, optimum et maximum. Dans ce dernier cas, il est alors possible de construire un sous-ensemble flou *d'optimalité* de forme trapèze, ayant pour support $[inf_{min}, sup_{max}]$ et pour noyau $[inf_{opt}, sup_{opt}]$.

Exemple 5.6 *Reprenons le tableau 5.2 : la façon dont les valeurs numériques sont extraites de ce tableau pour le type numérique "pH" est présentée dans l'exemple 5.5. La figure 5.3 représente le sous-ensemble flou d'optimalité qui est alors associé au type pH pour l'instanciation de la relation "Growth parameter - pH" sur la première ligne du tableau.*

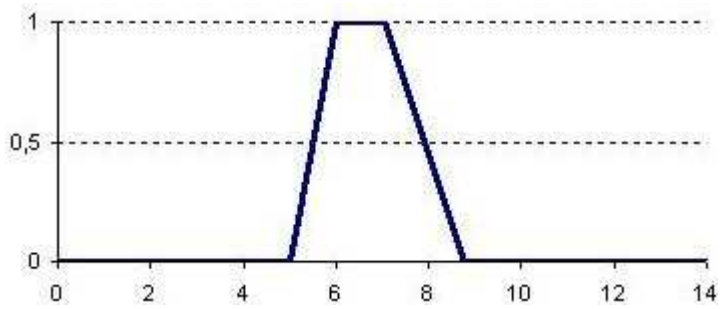

FIG. 5.3 – Ensemble flou pour la valeur du type "pH" pour la première ligne de données du tableau 5.2.

5.3.4 Construction d'un sous-ensemble flou pour des données imprécises

S'il n'a pas été possible de construire un sous-ensemble flou *d'optimalité* à partir des valeurs numériques extraites, alors on considère que les valeurs représentent une donnée imprécise (ou précise, si la valeur est unique). Un sous-ensemble flou représentant une *distribution de possibilité* pour les valeurs numériques extraites est alors construit.

Compte tenu des transformations réalisées au cours de la tentative de construction d'un sous-ensemble flou d'optimalité (voir la section précédente), les valeurs numériques extraites ont comme marqueur de type de colonne soit "inconnu", soit "écart-type".

S'il existe une seule valeur *err* de type "écart-type", accompagnée d'une seule valeur *val* de type "inconnu", alors un sous-ensemble flou de forme trapèze, ayant pour support $[val - err, val + err]$ et pour noyau $[val, val]$ est construit. Sinon, toutes les valeurs numériques extraites se voient affecter le marqueur de type de colonne "inconnu". Dans ce cas, un sous-ensemble flou par cellule est créé :

- si la cellule contient un intervalle, un sous-ensemble flou de forme trapèze, ayant comme support et comme noyau ce même intervalle est créé ;
- si la cellule contient une moyenne *moy* avec un écart-type *err*, un sous-ensemble flou de forme trapèze, ayant comme support $[moy - err, moy + err]$ et comme noyau $[moy, moy]$ est créé. Il s'agit en fait du même cas de figure que celui qui a déjà été traité avec les marqueurs de type de

colonne “inconnu” et “écart-type”, sauf qu'ici les deux valeurs numériques proviennent de la même cellule ;
– si la cellule contient des valeurs isolées (ou si les valeurs numériques proviennent du titre et non d'une cellule du tableau), on construit un sous-ensemble flou égal à l'ensemble (au sens classique du terme) des valeurs extraites, c'est-à-dire un sous-ensemble flou où la fonction d'appartenance est nulle partout sauf pour les valeurs numériques extraites, pour lesquelles la fonction d'appartenance est égale à 1. Les valeurs censurées sont cependant traitées de façon particulière . Les valeurs censurées sont repérées comme étant des valeurs isolées précédées de '>' ou '<' : on construit alors un sous-ensemble flou sous forme d'intervalle classique (i.e. fonction d'appartenance égale à 1 sur toutes les valeurs de l'intervalle, à 0 en dehors de l'intervalle). Soit $[min_{type}, max_{type}]$ l'intervalle de valeurs possibles pour le type numérique dont la valeur est en cours d'instanciation. Si la valeur val trouvée dans la cellule est censurée à gauche (i.e. la cellule du tableau contient “$< val$”), l'intervalle $[min_{type}, val]$ est construit ; si la valeur val est censurée à droite (c'est-à-dire si la cellule du tableau contient “$> val$”), l'intervalle $[val, max_{type}]$ est construit.

Le sous-ensemble flou final pour le type numérique considéré est ensuite construit comme l'union des sous-ensembles flous construits pour toutes les cellules concernées. Ce sous-ensemble flou est une distribution de possibilités : il s'agit en effet d'un sous-ensemble flou représentant des données imprécises, et il est normalisé par construction.

Exemple 5.7 *La figure 5.4 montre le sous-ensemble flou représentant une distribution de possibilités pour le type pH, pour l'instanciation de la relation “Propriétés d'un produit - pH” correspondant à la première ligne de données du tableau 5.3. Il s'agit ici de l'union de deux sous-ensembles flous de forme trapèze représentant chacun une valeur isolée.*

	pH of inoculated food	**pH of uninoculated food**
Ricotta Cheese	6.40	6.46

Tab. 5.3 – Extrait d'un tableau : Results of Challenge Test - S. aureus (issu de [Hogg et al., 2003])

Lors de l'instanciation d'une relation, une valeur d'un type numérique est donc toujours représentée par un sous-ensemble flou : il s'agit soit d'un sous-ensemble flou d'optimalité, de forme trapèze, soit d'un sous-ensemble flou représentant une distribution de possibilités, qui est obtenu par l'union de formes trapèzes et de valeurs isolées.

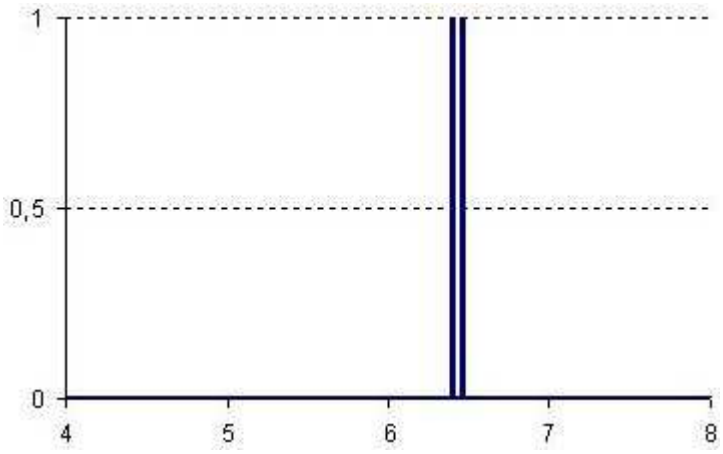

FIG. 5.4 – Sous-ensemble flou pour la valeur du type “pH” pour la première ligne de données du tableau 5.3.

5.3.5 Résultats expérimentaux

On réutilise les 60 tableaux de données qui ont déjà servi à évaluer notre méthode de reconnaissance des types de colonnes et des relations. Seules les 119 relations qui ont été correctement reconnues aux étapes précédentes sont considérées. On s'intéresse à l'instanciation des valeurs numériques pour ces relations : pour limiter le temps de vérification manuelle, seule l'instanciation sur la première ligne de données du tableau est étudiée. En effet, à l'intérieur d'un même tableau, la structure est homogène : si la première ligne est correctement instanciée, on considère en première approximation que les lignes suivantes sont aussi correctement instanciées. Pour chacune des 119 relations correctement reconnues par les étapes précédentes de l'annotation, la qualité de l'instanciation des valeurs numériques pour la première ligne du tableau concerné est évaluée manuellement.

En ce qui concerne l'extraction de valeurs numériques, parmi ces 119 instanciations de relations, on dénombre seulement 2 erreurs (une fois, une valeur numérique a été extraite pour un type alors qu'elle correspondait à un autre type, et une fois le signe négatif n'a pas été reconnu devant le nombre). Par ailleurs, pour 5 tableaux (soit 13 relations concernées), un type numérique (à chaque fois la température) n'a pas pu être instancié : dans chaque cas, le type n'était pas présent dans le tableau ni dans son titre, mais dans le texte environnant, ce qui n'est pas géré par notre approche pour l'instant (nous envisageons dans le futur de tenir compte des phrases dans le document qui font référence au tableau). Toutes les autres valeurs numériques ont été correctement extraites.

En ce qui concerne la construction des sous-ensembles flous, en plus des 2 relations impactées par les erreurs de reconnaissance sur les valeurs numériques

elles-mêmes, on dénombre 4 erreurs de construction des sous-ensembles flous : dans trois cas, un intervalle n'a pas été reconnu en tant que tel, mais comme deux valeurs isolées (correspondant au minimum et au maximum de l'intervalle), et dans un cas, le sous-ensemble flou d'optimalité n'a pas été reconnu en tant que tel mais représenté par des valeurs isolées (il s'agissait d'un cas complexe, dans lequel chaque cellule contenait deux valeurs car la même mesure de température était exprimée à la fois en $°C$ et en $°F$). Toutes les autres relations (i.e. 100 relations) ont vu la totalité de leurs types numériques correctement instanciés. Notre méthode donne donc de bons résultats pour l'instanciation des types numériques dans les tableaux.

5.4 Instanciation des valeurs symboliques

Nous venons de montrer comment les valeurs numériques sont instanciées pour une relation et représentées sous forme de sous-ensembles flous. Nous instancions également les types symboliques de la relation, et ces instanciations sont également représentées sous la forme de sous-ensembles flous. L'ensemble de définition d'un tel sous-ensemble flou pour un type symbolique donné est l'ensemble de tous les termes de la hiérarchie de ce type ; la fonction d'appartenance correspond à la mesure de similarité entre la valeur réelle utilisée dans le tableau et les termes de la hiérarchie du type, telle que présentée dans le chapitre 3.

Pour instancier le type symbolique *type* pour une relation du tableau sur une ligne donnée l_i, on recherche quelles colonnes ont été reconnues comme appartenant au type *type*. Pour chaque colonne *col* du type *type*, on construit sur la ligne l_i un sous-ensemble flou dont le domaine de définition est l'ensemble des termes de la hiérarchie du type *type* et dont la fonction d'appartenance est la mesure de similarité entre chacun des termes de la hiérarchie du type *type* et le terme trouvé dans la cellule de la colonne *col* et de la ligne l_i.

L'instanciation du type *type* pour la relation sur la ligne *li* est alors le sous-ensemble flou résultant de l'union de tous les sous-ensembles flous construits pour les différentes colonnes du tableau ayant été reconnues comme étant du type *type*.

Bien que notre algorithme prévoie la possibilité de reconnaître plusieurs colonnes d'un même type symbolique dans un tableau, cela n'a pas été le cas lors de notre évaluation expérimentale : sur chacun des 60 tableaux utilisés pour l'expérimentation, chaque type symbolique était associé à une seule colonne du tableau (voire aucune). Nous n'avons donc pas eu à évaluer plusieurs méthodes de combinaison des différentes colonnes. Les résultats expérimentaux sont présentés en section 4.3.3 en ce qui concerne la proportion de colonnes symboliques correctement reconnues. La qualité des sous-ensembles flous créés par calcul de similarité peut être évaluée en reprenant les résultats expérimentaux concernant l'annotation des cellules symboliques, présentés en section 3.1.5.

5.5 Annotation floue : impacts sur l'interrogation

Dans le système MIEL++, l'utilisateur peut exprimer ses critères de sélection sous la forme de sous-ensembles flous représentant des préférences. Dans le cadre de l'interrogation des tableaux annotés par notre système, nous devons donc examiner comment les préférences d'un utilisateur peuvent être comparées avec les données, représentées sous forme de sous-ensembles flous ayant pour sémantique l'imprécision, l'optimalité ou la similarité. Afin de pouvoir discuter de ces comparaisons de préférences avec les données, nous faisons quelques rappels de théorie des possibilités en section 5.5.1. Nous analysons ensuite en section 5.5.2 les sémantiques des sous-ensembles flous créés lors de notre annotation pour essayer de déterminer quel est le meilleur moyen de les interroger.

5.5.1 Comparaison de sous-ensembles flous dans la théorie des possibilités

La théorie des possibilités a été introduite par [Zadeh, 1978]. Les données imprécises sont modélisées sous forme de distributions de possibilités. Une distribution de possibilités est une disjonction pondérée de valeurs exclusives dans laquelle au moins une valeur a un degré de possibilité de 1. Elle est représentée sous forme de sous-ensemble flou normalisé ; chaque élément du support du sous-ensemble flou peut être la vraie valeur de la donnée, mais cette valeur est unique (seul un élément du support du sous-ensemble flou est le bon).

[Zadeh, 1978] définit le degré de possibilité d'adéquation entre des sous-ensembles flous représentant des préférences et des sous-ensembles flous représentant des données imprécises.

Définition 5.3 *Le* degré de possibilité d'adéquation *entre deux sous-ensembles flous A et B de X, l'un représentant un critère de sélection flou et l'autre une donnée imprécise, ayant respectivement pour fonction d'appartenance μ_A et μ_B, est $\pi(A, B) = sup_{x \in X}(min(\mu_A(x), \mu_B(x)))$.*

L'inconvénient du degré de possibilité d'adéquation est qu'il ne permet pas de distinguer, entre deux données imprécises répondant aux critères d'interrogation, une donnée représentant une ignorance totale d'une donnée précise (voir l'exemple 5.8). [Dubois & Prade, 1988] définit le degré de nécessité d'adéquation, une mesure d'adéquation entre préférences et données imprécises permettant de faire la distinction entre une ignorance totale et une donnée précise.

Définition 5.4 *Le* degré de nécessité d'adéquation *entre deux sous-ensembles flous A et B de X, l'un représentant un critère de sélection flou et l'autre une donnée imprécise, ayant respectivement pour fonction d'appartenance μ_A et μ_B, est $N(A, B) = 1 - sup_{x \in X}(min(1 - \mu_A(x),\ \mu_B(x)))$.*

Exemple 5.8 *La figure 5.5 représente un critère de sélection flou pour une température ainsi que trois données : une donnée précise (la température exacte a un degré de possibilité de* 1*, toutes les autres températures ont un degré de possibilité de* 0*), une ignorance totale (toutes les températures sont également possibles avec un degré de possibilité de* 1*), et une donnée imprécise « classique » sous forme d'intervalle (degré de possibilité de* 1 *sur l'intervalle, de* 0 *en dehors de l'intervalle).*

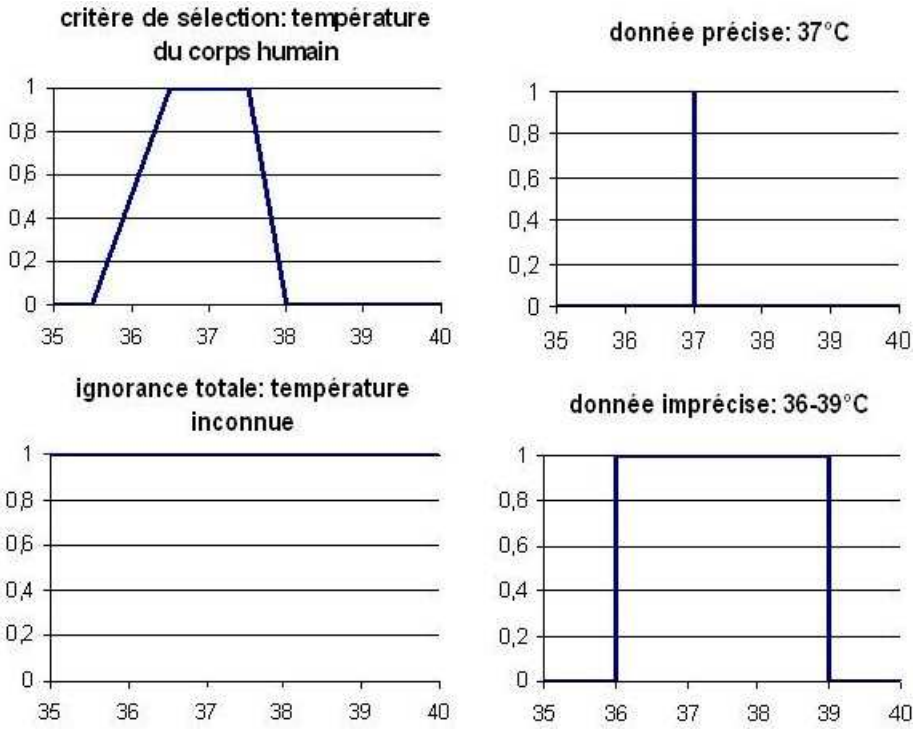

FIG. 5.5 – Un critère de sélection et trois données pour la température

Le degré de possibilité d'adéquation entre le critère de sélection et la donnée précise présentés dans la figure 5.5 est égal à 1, de même qu'entre le critère de sélection et l'ignorance totale, ainsi qu'entre le critère de sélection et la donnée imprécise. Par contre, le degré de nécessité d'adéquation entre le critère de sélection et la donnée précise est de 1, alors que le degré de nécessité d'adéquation entre le critère de sélection et l'ignorance totale est de 0 ; le degré de nécessité d'adéquation entre le critère de sélection et la donnée imprécise est également de 0 (en effet, si la valeur réelle est entre 38 et 39°C, alors elle ne correspond pas du tout au critère de sélection).

5.5.2 Nature des sous-ensembles flous générés dans l'annotation et impacts sur l'interrogation

Les données imprécises ou incertaines représentées dans les bases existantes du système MIEL sont représentées sous forme de distribution de possibilité. On se situe dans le cadre formel bien établi de la théorie des possibilités, et une combinaison des degrés de possibilité et de nécessité est utilisé pour le calcul de l'adéquation entre une requête utilisateur (sous-ensemble flou représentant des préférences) et une réponse (distribution de possibilité) [Buche et al., 2006]. Dans notre travail d'annotation, nous construisons de nouveaux sous-ensembles flous dont la sémantique doit être discutée afin de déterminer comment il convient au mieux de les interroger. Cette problématique est encore à l'état d'exploration et nous présentons ici nos premières pistes de travail.

Cas des sous-ensembles flous pour les données numériques

En section 5.3, nous avons vu que les types numériques d'une relation sont instanciés sous la forme de sous-ensembles flous : il peut s'agir de sous-ensembles flous représentant des distributions de possibilités, ou de ce que nous avons appelé des « sous-ensembles flous d'optimalité ».

En cherchant à mieux caractériser ces sous-ensembles flous, nous avons formulé l'hypothèse qu'ils peuvent également être considérés comme des sous-ensembles flous de possibilités, dans la mesure où la valeur optimale peut être considérée comme unique. Dans ce cas, le sous-ensemble flou représenterait la possibilité qu'une valeur soit la valeur optimale. Ceci se justifie dans le cas des paramètres de croissance des microorganismes : les modèles de croissance communément utilisés sont paramétrés avec *une* valeur optimale de température *précise*. Il est aisément imaginable que la valeur optimale ne soit pas exactement la même selon les individus (i.e. souches de bactéries), ce qui explique une variation et donc une distribution de possibilités.

Si cette hypothèse est retenue, la théorie des possibilités pourra être appliquée pour calculer un degré de possibilité d'adéquation entre une expression de préférences d'un utilisateur et une donnée floue d'optimalité (à condition bien entendu que les deux sous-ensembles flous soient définis sur le même ensemble de définition, i.e. une température, un pH, etc.). Il nous semble par contre que le degré de nécessité d'adéquation n'a pas de signification intéressante dans le cas de sous-ensembles flous d'optimalité, du moins dans notre domaine d'application. En effet, si la croissance d'un microorganisme est optimale aux conditions correspondant à la requête d'un utilisateur, ce microorganisme sera tout aussi intéressant pour l'utilisateur qu'il soit également capable de croître dans d'autres conditions que celles posées dans la requête ou non (dans l'optique où la question posée est « Quels sont les microorganismes qui croissent dans telles conditions ? »).

Cas des sous-ensembles flous pour les données symboliques

En ce qui concerne les types symboliques, nous avons tout d'abord été tentés de considérer également les annotations floues comme des distributions de possibilités. C'est le cas si on considère que le terme de l'ontologie le plus représentatif du terme du web est une valeur qui existe, mais qui n'est pas connue de façon certaine : les degrés de similarité entre termes de l'ontologie et termes du web représentent alors la possibilité qu'un terme de l'ontologie soit effectivement le terme le plus représentatif du terme du web. Ce qui nous intéressait initialement était de pouvoir ordonner les résultats d'une annotation, plutôt que de connaître les valeurs en elle-mêmes des degrés de similarités. On pouvait donc sans problème normaliser les sous-ensembles flous (i.e. appliquer une transformation pour amener la plus haute valeur du degré d'appartenance au sous-ensemble flou à la valeur 1).

Cependant, au cours des expérimentations puis de discussions avec des experts en risque alimentaire travaillant sur des ontologies alimentaires, ce point de vue a été remis en cause. Tout d'abord, le terme de l'ontologie le plus représentatif du terme du web n'est pas forcément unique (par exemple, pour les aliments composés, chacun de leurs composant peuvent être aussi représentatifs). Ensuite, la valeur du degré de similarité est une indication en elle-même de la qualité de l'annotation, et cette information ne doit pas être perdue en normalisant le sous-ensemble flou. Ceci est notamment important pour pouvoir facilement reconnaître une annotation sûre (un degré de similarité égal à 1 signifie que l'on a trouvé exactement la même chaîne de caractères dans l'ontologie, moyennant la suppression des stopwords et une lemmatisation), par rapport à un rapprochement incertain.

L'hypothèse que nous favorisons actuellement est donc que les sous-ensembles flous utilisés pour nos annotations sur les types symboliques représentent des similarités et non une distribution de possibilités correspondant à une imprécision (voir la section 5.2.2 qui décrit les différentes sémantiques des sous-ensembles flous). On ne pourra donc pas appliquer la théorie des possibilités pour calculer l'adéquation d'un terme du web annoté par un sous-ensemble flou sur les termes de l'ontologie, avec des préférences d'utilisateurs exprimées par un sous-ensemble flou sur les termes de l'ontologie.

Un premier élément de réponse nous est fourni par [Baziz et al., 2006], dans lequel les critères de sélection d'un utilisateur, exprimés sur une ontologie de mots clefs, sont comparés avec les mots clefs de chaque document, exprimés sur la même ontologie de mots clefs pondérée selon l'importance de chaque mot clef dans le document. La sémantique du sous-ensemble flou défini sur l'ontologie représentant les mots clefs d'un document peut être considérée comme une sémantique de similarité : en effet, le degré d'appartenance mesure à quel point un mot clef est proche du sujet traité dans le document.

Soit M l'ensemble des mots clefs de l'ontologie, μ_{req} la fonction d'appartenance

au sous-ensemble flou défini sur M désignant les mots clefs pour la requête req et μ_{doc} la fonction d'appartenance au sous-ensemble flou défini sur M désignant les mots clefs du document *doc*. Alors, si la requête est considérée comme disjonctive (c'est à dire qu'il suffit que le document contienne l'un des mots clefs, ce qui correspond à la manière dont un utilisateur exprime ses préférences dans le système MIEL++), le degré de pertinence du document pour la requête est calculé comme :

$$pertinence(doc, req) = max_{m \in M} min(\mu_{doc}(m), \mu_{req}(m)) \tag{5.3}$$

Cette mesure est calculée selon la même formule que le degré de possibilité d'adéquation dans le cadre de la théorie des possibilités.

Il nous semble, pour les sous-ensembles flous à sémantique de similarité, qu'il est intéressant de retenir une mesure de type degré de possibilité d'adéquation. En revanche, il ne nous paraît pas opportun, comme nous l'avons proposé pour les sous-ensembles flous à sémantique d'optimalité, de rejeter les mesures de type degré de nécessité d'adéquation. En effet, lorsqu'un terme du web est proche du terme sur lequel on interroge, le fait que ce terme soit également très proche d'un autre terme de l'ontologie n'ayant rien à voir avec la requête est informatif. A similarité égale du terme du web avec un terme de la requête, on sera plus intéressé par un terme du web étant similaire uniquement avec des termes proches dans l'ontologie du terme sur lequel on interroge, que par un terme du web étant similaire à d'autres termes très éloignés dans l'ontologie. Le calcul d'un tel « degré de nécessité d'adéquation » prenant en compte la hiérarchie des termes dans l'ontologie est un problème auquel nous souhaitons nous atteler dans la continuité de ce travail de thèse.

Conclusion du chapitre

Nous avons montré dans ce chapitre comment sont reconnues les relations représentées par les tableaux, ainsi que la façon dont ces relations sont instanciées pour chaque ligne du tableau. Chaque relation est assortie d'un score qui représente la certitude avec laquelle la relation a été reconnue, et chaque instance de type de données dans une relation est représentée par un sous-ensemble flou. Notre méthode de reconnaissance des relations est abordée dans [Hignette et al., 2007c] et expliquée plus en détails dans [Hignette et al., 2007b].

Nous avons commencé à explorer la façon dont les annotations floues peuvent être prises en compte dans le cadre d'une interrogation de l'entrepôt de données où l'utilisateur peut exprimer des préférences. Cependant, l'interrogation des tableaux annotés reste encore pour nous à l'état de perspectives de notre travail.

Maintenant que nous avons présenté l'ensemble des étapes de notre travail d'annotation de tableaux, nous présentons dans le chapitre suivant la façon dont

nous avons implémenté ce système d'annotation et comment il s'intègre dans le cadre plus large du projet WebContent.

Chapitre 6

Implémentation et intégration dans le projet WebContent

Notre méthode d'annotation de tableaux a été implémentée dans un prototype que nous présentons en section 6.1. Dans le cadre du projet WebContent, il nous faut répondre aux exigences d'interopérabilité avec les autres systèmes de recherche de documents, d'annotation, d'interrogation, etc. développés par les autres partenaires. Pour cela, il a notamment fallu redéfinir les formats d'entrée et de sortie de notre système en utilisant les langages du web sémantique. Nous présentons en section 6.2 comment la structure de notre ontologie peut être représentée en langage OWL. La représentation des tableaux dans le format d'échange de documents WebContent est quant à elle décrite en section 6.3. Nous présentons enfin en section 6.4 la façon dont nous comptons proposer notre système d'annotation sous forme de service web.

6.1 Implémentation d'un prototype d'annotation en Java

Un prototype a été construit pour évaluer expérimentalement la validité de l'approche d'annotation proposée. Ce prototype a été écrit en langage Java, version 1.6. Il prend en entrée des tableaux au format XTab et une ontologie dans un format ad-hoc et donne en sortie des tableaux annotés au format SML étendu à la représentation du flou. Nous présentons tout d'abord les entrées de notre système en section 6.1.1, puis nous expliquons comment les tableaux sont prétraités (section 6.1.2) avant d'être annotés (section 6.1.3). Le format de sortie de notre prototype pour les tableaux annotés est présenté en section 6.1.4.

6.1.1 Format des données en entrée

Notre prototype a été construit pour annoter des tableaux enregistrés au format Xtab. Ce format est un format XML (eXtensible Markup Language) suivant une DTD (Document Type Definition) spéciale définie par [Saïs, 2004] pour la représentation de tableaux sous forme canonique. Nous donnons ici comme exemple la représentation XTab du tableau 6.1.

```
<table>
  <title>
    <tabTitle>Approximate aw values of selected food categories</tabTitle>
    <colTitle>Animal Products</colTitle>
    <colTitle>aw</colTitle>
  </title>
  <content>
    <ligne>
      <case>fresh meat, poultry, fish</case>
      <case>0.99 - 1.00</case>
    </ligne>
    <ligne>
      <case>natural cheeses</case>
      <case>0.95 - 1.00</case>
    </ligne>
  </content>
</table>
```

Animal products	$\mathbf{a}_w$
fresh meat, poultry, fish	0.99 - 1.00
natural cheeses	0.95 - 1.00

TAB. 6.1 – Extrait d'un tableau : Approximate a_w values of selected food categories (issu de [U.S. Food and Drug Administration, 2001])

L'ontologie donnée en entrée de notre prototype, construite à partir de la base de données relationnelles du système MIEL, est représentée dans un format ad-hoc, constitué d'un ensemble de fichiers CSV (Comma-Separated Values) :

- un fichier pour la définition des termes des hiérarchies avec les poids des différents mots ;
- un fichier par hiérarchie définissant le nom du type symbolique et les relations père-fils dans la hiérarchie ;
- un fichier pour définir les types numériques avec leurs unités et leur intervalle de valeurs possibles ;
- un fichier pour définir les relations avec leur type résultat et leurs types d'accès.

6.1.2 Pré-traitement des tableaux

Les tableaux au format XTab sont tout d'abord traités à l'aide de l'application GATE (version 3.1b1)[Cunningham et al., 2005]. Tout d'abord, les fonctionnalités "Tokenizer" (séparation du texte en mots) et "sentence splitter" (séparation

du texte en phrases) de la chaîne de traitement prédéfinie ANNIE sont appliquées. La séparation du texte en phrases n'a pas vraiment de sens pour notre application puisque nous travaillons sur des tableaux et non des phrases entières, mais cette étape est techniquement obligatoire pour pouvoir utiliser la suite des traitements disponibles dans GATE. Ensuite l'application d'un "HashGazetteer" sur l'ensemble du document permet de rechercher les unités qui sont définies dans l'ontologie ainsi que les mots de la stopword list. Un ensemble de règles, écrites dans la grammaire JAPE, permet ensuite de reconnaître les nombres (y compris en format scientifique) et d'annoter les couples nombre-unité. Ces règles permettent également de reconnaître la présence d'un intervalle ou d'une moyenne associée à son écart-type. Grâce à cet ensemble de règles écrites en JAPE, en tenant compte du précédent résultat du HashGazetteer, on annote également les mots qui ne sont ni des unités ni des stopword.

Pour des raisons de simplicité, l'application GATE permettant la pré-annotation des tableaux a été construite et testée à petite échelle en utilisant l'interface graphique du logiciel GATE. L'application en état stable a ensuite été sérialisée dans un fichier : il est alors facile de lancer cette application sans interface graphique, par programmation en langage Java grâce à l'API de GATE. L'opération de pré-annotation des tableaux à l'aide de GATE se fait en traitement batch, indépendant de notre programme d'annotation proprement dit : il suffit de préciser les emplacements du dossier contenant les tableaux au format XTab et du dossier devant contenir les tableaux une fois annotés par GATE. Les fichiers produits sont identiques aux fichiers XTab d'origine, avec l'ajout de nouvelles balises Word, Unit, Number, NumberWithUnit, Interval, etc.

6.1.3 Annotation des tableaux

Une fois les tableaux annotés avec GATE, ils sont repris dans le programme d'annotation proprement dit. Il s'agit là encore d'un fonctionnement en batch, où l'on précise quel répertoire contient les tableaux à annoter. Les tableaux avec les annotations de GATE sont analysés les contenus des balises Word d'une case sont récupérés pour constituer un terme : les mots du terme sont alors lemmatisés. On utilise pour cela un lemmatiseur par substitution (remplaçant une fin de mot par une autre). Les règles de substitution utilisées par notre lemmatiseur sont décrites dans le tableau 6.2.

Une fois qu'une substitution est réalisée, le mot est recherché dans Wordnet version 2.1 [Miller & team, 2005] via l'API JWNL (Java WordNet Library) version 1.3 [Didion & Barton, 2003]. Si le mot lemmatisé existe dans Wordnet, il est ajouté à la liste des lemmes du mot. Toutes les substitutions possibles sont ainsi testées. Le mot lui-même, non transformé, est systématiquement ajouté à l'ensemble des lemmes du mot. Dans le calcul des scores de similarité par égalité mot à mot, deux mots sont considérés comme égaux si l'intersection de leurs deux ensembles de lemmes est non vide.

fin de mot	remplacée par	fin de mot	remplacée par
s	simple troncature	ies	y
es	simple troncature	es	e
ed	simple troncature	ed	e
ing	simple troncature	ing	e
er	simple troncature	er	e
est	simple troncature	est	e
men	man		

TAB. 6.2 – Règles de substitution du lemmatiseur.

Dans notre prototype, nous appliquons séquentiellement sur chaque tableau l'ensemble des étapes décrites dans les chapitres 4 et 5 pour produire les tableaux annotés : nous décrivons ci-après le format des tableaux annotés.

6.1.4 Tableaux annotés au format SML

Dans notre prototype, nous avons représenté les tableaux annotés au format SML (Semantic Markup Language) [Saïs, 2004] que nous avons étendu pour représenter des annotations floues. Le format SML a initialement été proposé par [Saïs, 2004] pour la représentation de tableaux XTab annotés sémantiquement.

Chaque ligne du tableau est remplacée par la balise *relLine* pour exprimer le fait qu'on travaille sur des relations. Chaque relation est alors représentée par une balise du nom de la relation, englobant des balises portant les noms des différents types de la signature de la relation.

Chaque balise de type contient un attribut *indOnto*, permettant de tracer la manière dont a été faite l'annotation : *indOnto="complete"* si l'annotation a été trouvée par égalité des ensembles de mots lemmatisés, *indOnto="inclusion"* ou *indOnto="intersection"* si l'annotation a été trouvée par inclusion ou intersection des ensembles de mots lemmatisés, *indOnto="notFound"* si aucun terme de l'ontologie n'a été retenu pour l'annotation du terme du web. Chaque balise de type contient une balise *originVal* indiquant la valeur initiale trouvée dans le tableau, et éventuellement une ou des balise(s) *finalVal* contenant la (les) valeur(s) correspondante(s) dans l'ontologie.

Nous reprenons ici en exemple le tableau 6.1, représenté au format SML avec ses annotations. Ce format n'est pas prévu pour la représentation des données numériques, aussi pour ces données il n'y a qu'une *originVal* sans *finalVal*. Ce format prévoit une seule manière de reconnaître les valeurs correspondantes dans l'ontologie (égalité, inclusion ou intersection de mots) pour un type donné sur une ligne donnée : aussi nous ne faisons figurer dans cette annotation que les correspondances par inclusion.

```
<table>
```

```
  <title>
    <tabTitle>Approximate aw values of selected food categories</tabTitle>
    <colTitle>Animal Products</colTitle>
    <colTitle>aw</colTitle>
  </title>
  <content>
    <relLine>
      <ProductPropertyAw>
        <FoodProduct indOnto="inclusion">
          <originVal nbwords="4">fresh meat, poultry, fish</originVal>
          <finalVal nbwords="2">Fresh meat</finalVal>
          <finalVal nbwords="2">Fresh fish</finalVal>
          <finalVal nbwords="2">Poultry meat</finalVal>
          <finalVal nbwords="1">Fish</finalVal>
          <finalVal nbwords="1">Poultry</finalVal>
          <finalVal nbwords="1">Meat</finalVal>
        </FoodProduct>
        <AwWaterActivity><originVal>0.99 - 1.00</originVal><AwWaterActivity>
      </ProductPropertyAw>
    </relLine>
    <relLine>
      <ProductPropertyAw>
        <FoodProduct indOnto="inclusion">
          <originVal nbwords="2">natural cheeses</originVal>
          <finalVal nbwords="1">cheese</finalVal>
        </FoodProduct>
        <AwWaterActivity><originVal>0.99 - 1.00</originVal><AwWaterActivity>
      </ProductPropertyAw>
    </relLine>
  </content>
</table>
```

Dans [Buche et al., 2006], nous avons proposé une extension du format SML pour pouvoir représenter des sous-ensembles flous dans l'annotation des cellules. Cette extension prévoit une seule *finalVal* couplée à une *originVal*, cette *finalVal* étant un sous-ensemble flou. Nous avons prévu de pouvoir représenter des sous-ensembles flous pour les types symboliques, mais aussi pour les types numériques. Pour les types symboliques, la balise *DFS* (Discrete Fuzzy Set) représente le sous-ensemble flou et contient les différents termes ayant un degré d'appartenance au sous-ensemble flou non nul. Le degré d'appartenance d'un terme au sous-ensemble flou est donné dans la balise *MD* (Membership Degree). On peut ici représenter l'ensemble de tous les termes de l'ontologie correspondant au terme trouvé dans le tableau, qu'ils aient été trouvés par égalité, inclusion ou intersection de mots. Pour les types numériques, la balise *CFS* représente le sous-ensemble flou et contient les valeurs minimales et maximales du support et du noyau du sous-ensemble flou. Nous donnons ici en exemple comment est annotée la première ligne du tableau 6.1 avec ce format SML étendu aux sous-ensembles flous (pour raccourcir l'exemple, dans le sous-ensemble flou pour le type FoodProduct, nous n'avons représenté que les termes ayant un degré d'appartenance supérieur à 0,5).

```
<relLine>
  <ProductPropertyAw>
    <FoodProduct>
      <originVal>fresh meat, poultry, fish</originVal>
      <finalVal>
        <DFS>
          <ValF><Item>Meat and meat products</Item><MD>0,667</MD></ValF>
```

```
          <ValF><Item>Fresh fish</Item><MD>0,671</MD></ValF>
          <ValF><Item>Fresh meat cuts</Item><MD>0,612</MD></ValF>
          <ValF><Item>Minced meat fresh</Item><MD>0,612</MD></ValF>
          <ValF><Item>Fresh meat</Item><MD>0,671</MD></ValF>
          <ValF><Item>Meat</Item><MD>0,5</MD></ValF>
          <ValF><Item>Fish</Item><MD>0,5</MD></ValF>
          <ValF><Item>Poultry</Item><MD>0,5</MD></ValF>
          <ValF><Item>Poultry meat</Item><MD>0,671</MD></ValF>
        </DFS>
      </finalVal>
    </FoodProduct>
    <AwWaterActivity>
      <originVal>0.99 - 1.00</originVal>
      <finalVal>
        <MinSupport>0,99</MinSupport><MaxSupport>1,00</MaxSupport>
        <MinKernel>0,99</MinKernel><MaxKernel>1,00</MaxKernel>
      </finalVal>
    <AwWaterActivity>
  </ProductPropertyAw>
</relLine>
```

Ce format n'a pas été étendu pour représenter les informations venant de plusieurs colonnes, ni les sous-ensembles flous composés d'une union de sous-ensembles flous de forme trapèze pour les valeurs numériques. En effet, lorsque nous avons abordé ces problèmes, le format XTab et le format SML avaient été abandonnés, un autre format d'échange de documents ayant été défini dans le cadre du projet WebContent. Ce nouveau format d'échange est présenté en section 6.3.

6.2 Représentation de l'ontologie au format OWL

Dans notre prototype, nous avons utilisé un format ad-hoc de représentation de l'ontologie, dans lequel il était facile de représenter des poids sur les mots pour les différents termes, tout autant qu'il était aisé de représenter les signatures des relation n-aires. Dans le cadre du projet WebContent cependant, le format de représentation des ontologies choisi est le format OWL. Ce format ne permet pas directement de représenter des relations n-aires : les seules relations possibles sont des relations binaires, également appelées propriétés. Nous avons donc dû trouver une modélisation appropriée pour l'ontologie, en tenant compte des contraintes liées à l'utilisation du langage OWL.

Le changement de format de représentation de l'ontologie n'a pas beaucoup d'impact sur l'implémentation de notre méthode d'annotation : il suffit de créer un adaptateur capable de lire l'ontologie en format OWL et de la représenter en mémoire selon le schéma de représentation utilisé dans notre prototype.

Nous présentons en section 6.2.1 comment sont représentés en OWL les termes pondérés de notre ontologie. Nous présentons ensuite comment sont représentés les types de données (section 6.2.2) ainsi que les relations de l'ontologie (section 6.2.3).

6.2.1 Représentation d'un terme pondéré

Comme cela a été présenté dans le chapitre 3, on souhaite pouvoir pondérer chacun des mots constituant un terme de l'ontologie. On définit donc une classe WeightedTerm qui représente un terme, composé de plusieurs mots pondérés représentés par la classe WeightedWord, chaque WeightedWord étant associé à un mot et son poids. Tout objet de l'ontologie peut être associé à un WeightedTerm via la propriété AssociatedWeightedTerm.

```
<owl:Class rdf:ID="WeightedTerm"></owl:Class>
<owl:Class rdf:ID="WeightedWord"></owl:Class>
<owl:ObjectProperty rdf:ID="HasForWeightedWord">
  <rdfs:domain rdf:resource="#WeightedTerm"/>
  <rdfs:range rdf:resource="#WeightedWord"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="HasForWeight">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#WeightedWord"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="HasForWord">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#WeightedWord"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="AssociatedWeightedTerm">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="#WeightedTerm"/>
</owl:FunctionalProperty>
```

Nous présentons ici comme exemple la définition du terme *Food product*, qui correspond au nom d'un type symbolique.

```
<WeightedTerm rdf:ID="WeightedFoodProduct">
  <HasForWeightedWord>
    <WeightedWord rdf:ID="Food1.0">
      <HasForWord rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Food</
          HasForWord>
      <HasForWeight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</
          HasForWeight>
    </WeightedWord>
  </HasForWeightedWord>
  <HasForWeightedWord>
    <WeightedWord rdf:ID="Product0.5">
      <HasForWord rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Product
          </HasForWord>
      <HasForWeight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.5</
          HasForWeight>
    </WeightedWord>
  </HasForWeightedWord>
</WeightedTerm>
```

Le même modèle est utilisé pour représenter tous les termes de l'ontologie, qu'ils représentent des noms de types ou de relation, ou bien des termes de la hiérarchie d'un type symbolique. Nous donnons ici l'exemple du terme *Animal product*, terme faisant partie de la hiérarchie du type "Food product".

```
<WeightedTerm rdf:ID="WeightedAnimalProduct">
  <HasForWeightedWord>
    <WeightedWord rdf:ID="Animal1.0">
```

```
      <HasForWord rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Animal
          </HasForWord>
```

```
      <HasForWeight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</
          HasForWeight>
    </WeightedWord>
  </HasForWeightedWord>
  <HasForWeightedWord>
    <WeightedWord rdf:resource="#Products0.5" />
  </HasForWeightedWord>
</WeightedTerm>
```

6.2.2 Représentation des types de données

Les types symboliques et numériques de l'ontologie sont représentés comme des sous-classe d'une classe de type de données générique appelée Attribute : les types symboliques seront sous-classe de SymbolicAttribute, les types numériques sous-classe de NumericalAttribute.

```
<owl:Class rdf:ID="Attribute"></owl:Class>
<owl:Class rdf:ID="NumericalAttribute">
  <rdfs:subClassOf rdf:resource="#Attribute"/>
</owl:Class>
<owl:Class rdf:ID="SymbolicAttribute">
  <rdfs:subClassOf rdf:resource="#Attribute"/>
  <owl:disjointWith rdf:resource="#NumericalAttribute"/>
</owl:Class>
```

Types symboliques

La hiérarchie d'un type symbolique est représentée comme une hiérarchie de classes OWL. On crée une classe racine de la hiérarchie, sous-classe de la classe Taxonomy, qui sert d'ancrage à la hiérarchie même si elle ne porte pas de terme. Ainsi, le type symbolique aura un lien vers la racine de sa hiérarchie via la propriété HasForTaxonomyRoot. Cela permet d'avoir une bonne distinction entre les termes constitutifs de la hiérarchie du type symbolique (rattachés à une sous-classe de Taxonomy) et le nom du type symbolique lui-même (rattaché à une sous-classe de SymbolicAttribute). Nous présentons ici un exemple sur le type symbolique "Food Product".

```
<owl:Class rdf:ID="FoodProduct">
  <rdfs:subClassOf rdf:resource="#SymbolicAttribute"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#HasForTaxonomyRoot"/>
      <owl:allValuesFrom rdf:resource="#FoodProductRoot"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedWeightedTerm"/>
      <owl:hasValue>
        <WeightedTerm rdf:resource="#WeightedFoodProduct" />
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:ID="FoodProductRoot">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Taxonomy"/>
  </rdfs:subClassOf>
</owl:Class>
```

Dans cet exemple, le terme *Animal Product* est directement rattaché à la racine de la hiérarchie du type "Food Product".

```
<owl:Class rdf:ID="AnimalProduct">
  <rdfs:subClassOf rdf:resource="#FoodProductRoot"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedWeightedTerm"/>
      <owl:hasValue>
        <WeightedTerm rdf:resource="#WeightedAnimalProduct" />
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Types numériques

L'intervalle de valeurs possibles pour un type numérique est représenté par deux propriétés rattachées à la classe représentant le type : HasForMinValue et HasForMaxValue.

```
<owl:FunctionalProperty rdf:ID="HasForMinValue">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#NumericalAttribute"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="HasForMaxValue">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#NumericalAttribute"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
```

La liste des unités valides pour le type numérique est représentée dans une classe AttributeUnitList, reliée au type par la propriété AssociatedUnitList. Les différentes unités possibles sont des chaînes de caractères, intégrées dans un AttributeUnitList via la propriété HasForPossibleUnits.

```
<owl:Class rdf:ID="AttributeUnitList"></owl:Class>
<owl:FunctionalProperty rdf:ID="AssociatedUnitList">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#NumericalAttribute"/>
  <rdfs:range rdf:resource="#AttributeUnitList"/>
</owl:FunctionalProperty>
<owl:DatatypeProperty rdf:ID="HasForPossibleUnits">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#AttributeUnitList"/>
</owl:DatatypeProperty>
```

Nous présentons ici un exemple sur le type numérique "NACL", ayant pour unité % et pour intervalle de valeurs $[0, 100]$.

```
<owl:Class rdf:ID="NACL">
  <rdfs:subClassOf rdf:resource="#NumericalAttribute"/>
```

```
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedWeightedTerm"/>
      <owl:hasValue>
        <WeightedTerm rdf:ID="WeightedNACL">
          <HasForWeightedWord>
            <WeightedWord rdf:ID="NACL1.0">
              <HasForWeight rdf:datatype="http://www.w3.org/2001/XMLSchema#float
                  ">1.0</HasForWeight>
              <HasForWord rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                  >NACL</HasForWord>
            </WeightedWord>
          </HasForWeightedWord>
        </WeightedTerm>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#HasForMaxValue"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</owl
          :hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty ref:resource="#HasForMinValue"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</owl:
          hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty ref:resource="#AssociatedUnitList"/>
      <owl:hasValue>
        <AttributeUnitList rdf:ID="NACLList">
          <HasForPossibleUnits rdf:datatype="http://www.w3.org/2001/XMLSchema#
              string">%</HasForPossibleUnits>
        </AttributeUnitList>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

6.2.3 Représentation des relations

Une relation de l'ontologie est représentée par une sous-classe de la classe Relation. Une relation possède un et un seul type résultat : cela est représenté par la propriété AssociatedResult sur laquelle est définie une restriction de cardinalité (égale à un). Une relation possède au moins un type d'accès : cela est représenté par la propriété AssociatedKey sur laquelle est définie une restriction de cardinalité (supérieure ou égale à un).

```
<owl:ObjectProperty rdf:ID="AssociatedKey">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Relation"/>
  <rdfs:range rdf:resource="#Attribute"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="AssociatedResult">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Relation"/>
```

```
  <rdfs:range rdf:resource="#Attribute"/>
</owl:FunctionalProperty>
<owl:Class rdf:ID="Relation">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedResult"/>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#
          nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedKey"/>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#
          nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Nous présentons ici un exemple avec la relation "Prevalence", qui associe aux types d'accès "Food Product" et "Microorganism" le type résultat "Samples Positive".

```
<owl:Class rdf:ID="Prevalence">
  <rdfs:subClassOf rdf:resource="#Relation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedWeightedTerm"/>
      <owl:hasValue>
        <WeightedTerm rdf:ID="WeightedPrevalence">
          <HasForWeightedWord>
            <WeightedWord rdf:ID="Prevalence1.0">
              <HasForWeight rdf:datatype="http://www.w3.org/2001/XMLSchema#float
                  ">1.0</HasForWeight>
              <HasForWord rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                  >Prevalence</HasForWord>
            </WeightedWord>
          </HasForWeightedWord>
        </WeightedTerm>
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedResult"/>
      <owl:allValuesFrom rdf:resource="#SamplesPositive"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedKey"/>
      <owl:someValuesFrom rdf:resource="#FoodProduct"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#AssociatedKey"/>
      <owl:someValuesFrom rdf:resource="#Microorganism"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

6.3 Représentation des tableaux au format d'échange WebContent

Le format de représentation des tableaux XTab, qui a été initialement adopté pour notre travail, n'est pas celui qui a été retenu dans le cadre du projet WebContent. Nous présentons donc le format d'échange de documents qui a été retenu en section 6.3.1, et nous montrons comment les annotations du tableau sont représentées dans ce format en section 6.3.2.

Le changement de format de représentation des tableaux n'a pas d'influence sur le pré-traitement des tableaux avec GATE : en effet, il s'agit toujours de documents XML, qui sont lus par l'application et réécrits tels quels avec l'ajout des balises correspondant aux traitements dans GATE. Ce changement de format n'a pas un impact trop important sur le programme principal d'annotation. Il suffit de construire un adaptateur en entrée qui lit le nouveau format et le représente en mémoire selon le format de représentation du prototype, et de construire un adaptateur en sortie qui écrit les annotations selon le bon format : le moteur d'annotation lui-même reste inchangé.

6.3.1 Le format d'échange de documents WebContent

Le format d'échange de documents dans le projet WebContent se devait de respecter les recommandations du W3C (World Wide Web Consortium). Il a été décidé d'utiliser un format XHTML (XML-HTML, c'est-à-dire un format HTML correspondant à un document XML bien formé) amélioré, dans lequel chaque élément du document est une ressource ayant une URI (Uniform Resource Identifier) permettant de référencer cet élément dans les annotations. Nous présentons ici en exemple la représentation au format WebContent du tableau 6.3.

Organism	aw minimum	aw optimum	aw maximum
Clostridium perfringens	0.943	0.95-0.96	0.97
Staphylococcus aureus	0.83	0.98	0.99

TAB. 6.3 – Extrait d'un tableau : Approximate a_w values for growth of selected pathogens in food (issu de [U.S. Food and Drug Administration, 2001])

```
<table uri="docURI/Table:1">
 <caption uri="docURI/Table:1/Caption:1">Approximate aw values for growth of
     selected pathogens in food</caption>
 <thead>
  <tr uri="docURI/Table:1/Row:1">
   <th>
    <text uri="docURI/Table:1/Row:1/Text:1"><content>Organism</content></text>
   </th>
   <th>
    <text uri="docURI/Table:1/Row:1/Text:2"><content>aw minimum</content></text>
   </th>
```

```
   <th>
    <text uri="docURI/Table:1/Row:1/Text:3"><content>aw optimum</content></text>
   </th>
   <th>
    <text uri="docURI/Table:1/Row:1/Text:4"><content>aw maximum</content></text>
   </th>
  </tr>
 </thead>
 <tbody>
  <tr uri="docURI/Table:1/Row:2">
   <td>
    <text uri="docURI/Table:1/Row:2/Text:1"><content>Clostridium perfringens</
        content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:2/Text:2"><content>0.943</content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:2/Text:3"><content>0.95-0.96</content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:2/Text:4"><content>0.97</content></text>
   </td>
  </tr>
  <tr uri="docURI/Table:1/Row:3">
   <td>
    <text uri="docURI/Table:1/Row:3/Text:1"><content>Staphylococcus aureus</
        content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:3/Text:2"><content>0.83</content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:3/Text:3"><content>0.98</content></text>
   </td>
   <td>
    <text uri="docURI/Table:1/Row:3/Text:4"><content>0.99</content></text>
   </td>
  </tr>
 </tbody>
</table>
```

6.3.2 Représentation des annotations d'un tableau

Comme indiqué dans le chapitre 5, nous construisons des annotations représentées sous la forme de sous-ensembles flous. On rajoute donc à l'ontologie des classes utilitaires nous permettant de représenter des sous-ensembles flous. Un ensemble flou (classe FuzzySet) peut porter différentes sémantiques : similarité, imprécision, optimalité.

```
<owl:Class rdf:ID="FuzzySet"></owl:Class>
<owl:FunctionalProperty rdf:ID="HasForSemantic">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#FuzzySet"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
            similarity</rdf:first>
        <rdf:rest rdf:parseType="Resource">
```

```
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
              impreciseValue</rdf:first>

          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
                optimality</rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#
                nil"/>

          </rdf:rest>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:FunctionalProperty>
```

Lorsqu'une relation est instanciée sur une ligne du tableau, ce sont les types de données de sa signature qui sont instanciés. Ces instances de types de données contiennent deux informations : la liste des éléments (cellule, titre de colonne, titre du tableau) ayant permis de construire l'annotation, représentée grâce à la propriété IsConstructedFrom, ainsi que le sous-ensemble flou utilisé pour l'annotation, représenté grâce à la propriété IsAnnotatedBy.

```
<owl:ObjectProperty rdf:ID="IsConstructedFrom">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Attribute"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="IsAnnotatedBy">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="#FuzzySet"/>
</owl:FunctionalProperty>
```

Nous manipulons des sous-ensembles flous sur des domaines discrets (pour l'instanciation des types symboliques) et sur des domaines continus (pour l'instanciation des types numériques). Pour un sous-ensemble flou ayant un domaine de définition discret, nous représentons chaque valeur du domaine de définition pour laquelle la fonction d'appartenance n'est pas nulle, associée à la fonction d'appartenance pour cette valeur. Un ensemble flou sur un domaine de définition discret est une instance de DFS (Discrete Fuzzy Set), reliée à autant d'instances de Taxonomy que nécessaire via la propriété HasForElements : ces instances portent leur degré d'appartenance à l'ensemble flou via la propriété HasForMembershipDegree.

```
<owl:Class rdf:ID="DFS">
  <rdfs:subClassOf rdf:resource="#FuzzySet"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="HasForElements">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>

  <rdfs:domain rdf:resource="#DFS"/>
  <rdfs:range rdf:resource="#Taxonomy"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="HasForMembershipDegree">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#Taxonomy"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
```

Les sous-ensembles flous sur des domaines de définition continus sont représentés sous la forme d'une union de sous-ensembles flous de types trapèzes : les valeurs isolées sont alors simplement représentées comme un sous-ensemble flou de type trapèze dont les valeurs minimum du noyau, maximum du noyau, minimum du support et maximum du support sont toutes égales à la valeur isolée considérée. Un ensemble flou sur un domaine de définition continu est une instance de CFS (Continuous Fuzzy Set), reliée à autant d'instances de TrapezoidFuzzySet que nécessaire via la propriété IsComposedOf. Chaque TrapezoidFuzzySet est alors associé à son noyau et son support via les propriétés HasForMinKernel, HasForMaxKernel, HasForMinSupport et HasForMaxSupport.

```
<owl:Class rdf:ID="CFS">
  <rdfs:subClassOf rdf:resource="#FuzzySet"/>
</owl:Class>
<owl:Class rdf:ID="TrapezoidFuzzySet"/>
<owl:ObjectProperty rdf:ID="IsComposedOf">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#CFS"/>
  <rdfs:range rdf:resource="#TrapezoidFuzzySet"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="HasForMinSupport">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#TrapezoidFuzzySet"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="HasForMaxSupport">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#TrapezoidFuzzySet"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="HasForMinKernel">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#TrapezoidFuzzySet"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="HasForMaxKernel">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#TrapezoidFuzzySet"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
```

Tous les éléments nécessaires sont mis en place dans l'ontologie en OWL pour représenter les annotations des relations dans un tableau. Reprenons comme exemple le tableau 6.3, dont la représentation au format d'échange WebContent a été donnée en section 6.3.1. Les annotations créées pour chaque ligne du tableau sont ajoutées directement dans le document représentant le tableau. Nous donnons ici comme exemple l'annotation de la première ligne du tableau 6.3.

```
<annotation subject="docURI/Table:1/Row:2" uri="docURI/Table:1/Row:2/annotation
    :1"><data>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:onto="
     http://metarisk.inapg.inra.fr/MicrobialOntologyV4/">
  <onto:GrowthParameterAw rdf:about="docURI/Table:1/Row:2" >
   <onto:HasForNbColRec>2</onto:HasForNbColRec>
   <onto:HasForScore>1.0</onto:HasForScore>
   <onto:AssociatedKey rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw:1/
       Microorganism:1" />
```

```
   <onto:AssociatedResult rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw
       :1/Aw:1" />
  </onto:GrowthParameterAw>
  <onto:Microorganism rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/
      Microorganism:1">
   <onto:IsConstructedFrom rdf:resource="docURI/Table:1/Row:2/Text:1"/>
   <onto:IsAnnotatedBy>
    <onto:DFS rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/Microorganism
        :1/DFS:1">
     <onto:HasForSemantic>similarity</onto:HasForSemantic>
     <onto:HasForElements rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw
         :1/Microorganism:1/DFS:1/elt:1"/>
     <onto:HasForElements rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw
         :1/Microorganism:1/DFS:1/elt:2"/>
     <onto:HasForElements rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw
         :1/Microorganism:1/DFS:1/elt:3"/>
    </onto:DFS>
   </onto:IsAnnotatedBy>
  </onto:Microorganism>
  <onto:ClostridiumPerfringens rdf:about="docURI/Table:1/Row:2/GrowthParameterAw
      :1/Microorganism:1/DFS:1/elt:1">
   <onto:HasForMembershipDegree>1.0</onto:HasForMembershipDegree>
  </onto:ClostridiumPerfringens>
  <onto:Clostridium rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/
      Microorganism:1/DFS:1/elt:2">
   <onto:HasForMembershipDegree>0.707</onto:HasForMembershipDegree>
  </onto:Clostridium>
  <onto:ClostridiumBotulinum rdf:about="docURI/Table:1/Row:2/GrowthParameterAw
      :1/Microorganism:1/DFS:1/elt:3">
   <onto:HasForMembershipDegree>0.5</onto:HasForMembershipDegree>
  </onto:ClostridiumBotulinum>
  <onto:Aw rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/Aw:1">
   <onto:IsConstructedFrom rdf:resource="docURI/Table:1/Row:2/Text:2"/>
   <onto:IsConstructedFrom rdf:resource="docURI/Table:1/Row:2/Text:3"/>
   <onto:IsConstructedFrom rdf:resource="docURI/Table:1/Row:2/Text:4"/>
   <onto:IsAnnotatedBy rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw:1/Aw
       :1/CFS:1" />
  </onto:Aw>
  <onto:CFS rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/Aw:1/CFS:1">
   <onto:HasForSemantic>optimality</onto:HasForSemantic>
   <onto:IsComposedOf rdf:resource="docURI/Table:1/Row:2/GrowthParameterAw:1/Aw
       :1/CFS:1/TFS:1"/>
  </onto:CFS>
  <onto:TrapezoidFuzzySet rdf:about="docURI/Table:1/Row:2/GrowthParameterAw:1/Aw
      :1/CFS:1/TFS:1">
   <onto:HasForMinSupport>0.943</onto:HasForMinSupport>
   <onto:HasForMaxSupport>0.97</onto:HasForMaxSupport>
   <onto:HasForMinKernel>0.95</onto:HasForMinKernel>
   <onto:HasForMaxKernel>0.96</onto:HasForMaxKernel>
  </onto:TrapezoidFuzzySet>
 </rdf:RDF>
</data></annotation>
```

6.4 Intégration sous forme de service web

Pour le moment, notre système d'annotation fonctionne en local sur la machine sur laquelle il a été installé. Dans le cadre du projet WebContent, notre système d'annotation doit être disponible sous forme de service web, accessible via la plate-forme construite durant le projet. Nous proposons de mettre en place

un appel au service d'annotation en deux temps :

1. tout d'abord, l'ontologie au format OWL est envoyée au service. Le service teste si cette ontologie est déjà enregistrée dans sa base d'ontologies : si c'est le cas, le service renvoie l'identifiant de l'ontologie dans la base. Si l'ontologie n'est pas déjà enregistrée, le service teste si la structure de l'ontologie est compatible avec notre système d'annotation : si ce n'est pas le cas, le service renvoie un message d'erreur. Si la structure de l'ontologie est compatible avec notre système d'annotation, alors l'ontologie est enregistrée dans la base d'ontologies, et le service renvoie l'identifiant de l'ontologie dans la base ;
2. ensuite, un document à annoter au format d'échange de WebContent est envoyé au service, en même temps que l'identifiant de l'ontologie à utiliser pour l'annotation du document. Le document est analysé et annoté par notre système, et le service renvoie le document annoté.

L'intégration de notre système d'annotation sous forme de service web est en cours d'implémentation.

Conclusion du chapitre

Nous avons présenté dans ce chapitre comment nous avons implémenté notre prototype d'annotation de tableaux, mais aussi comment nous sommes en train d'ouvrir notre système pour en faire un service web qui utilisera les langages standards du web sémantique. Notre système d'annotation de tableaux, qui a pour l'instant été testé à petite échelle sur le domaine de la microbiologie alimentaire, pourra ainsi être testé à plus grande échelle par divers partenaires sur d'autres domaines d'applications.

Conclusion et perspectives

Nous avons présenté dans ce mémoire de thèse une méthode originale pour l'annotation non supervisée de tableaux. Notre méthode utilise uniquement les connaissances du domaine d'application formalisées dans une ontologie et traite tous les problèmes depuis la reconnaissance du type de données jusqu'à l'instanciation des relations, en tenant compte du fait que le vocabulaire employé dans le tableau ne correspond pas forcément à celui disponible dans l'ontologie.

L'ontologie de domaine avec laquelle les tableaux sont annotés est utilisée à chaque étape de notre système d'annotation. Ces différentes étapes sont :

1. distinction entre les colonnes numériques et symboliques. Cette étape permet d'utiliser par la suite deux méthodes d'annotation différentes pour les colonnes, selon qu'elles sont numériques ou symboliques. Les unités définies dans l'ontologie pour les types numériques sont utilisées comme indice pour déterminer si la colonne est numérique. La liste des indicateurs de résultat absent définie dans l'ontologie permet quant à elle d'éviter de considérer comme un indice de symbolique un commentaire textuel lorsqu'il indique simplement qu'une mesure n'a pas été effectuée. Cette prise en compte de l'ontologie dès l'étape de distinction entre les colonnes numériques et symboliques nous permet d'obtenir une bonne détermination des colonnes, indispensable pour les étapes suivantes ;
2. détermination du type numérique de l'ontologie pour une colonne numérique. Nous utilisons une approche multi-critères, dans laquelle le titre de la colonne est comparé avec les noms des différents types numériques de l'ontologie, mais également dans laquelle les unités présentes dans la colonne sont comparées avec celles définies pour chaque type numérique dans l'ontologie. L'intervalle de valeurs possibles défini pour les différents types numériques de l'ontologie est également utilisé pour filtrer les types utilisables dans l'annotation d'une colonne numérique, en fonction des valeurs numériques présentes dans la colonne. Toutes les connaissances du domaine présentes dans l'ontologie sont donc prises en compte pour obtenir une précision maximale dans la reconnaissance des types des colonnes numériques ;
3. détermination du type symbolique de l'ontologie pour une colonne symbolique. Nous utilisons là aussi une approche multi-critères, dans laquelle

le titre de la colonne est comparé avec les noms des différents types symboliques de l'ontologie, mais qui tient également compte de la similarité entre les termes utilisés dans la colonne et les termes de la hiérarchie de chaque type symbolique de l'ontologie. Ainsi, l'approche multi-critères permet d'être moins tributaire de l'hétérogénéité du vocabulaire utilisé : il y a plus de chances qu'un terme du web ressemble à un terme de l'ontologie si tous les termes utilisés dans la colonne ou son titre sont pris en compte, plutôt que seul le titre ou le contenu ;

4. reconnaissance des relations représentées par le tableau. Nous utilisons toujours une approche multi-critères, dans laquelle le titre du tableau est comparé avec les noms des différentes relations de l'ontologie et la signature du tableau (ensemble des types des colonnes reconnus dans les deux étapes précédentes) est comparée avec la signature des différentes relations définies dans l'ontologie. Il est nécessaire d'avoir reconnu au moins le type résultat de la relation parmi les colonnes du tableau pour pouvoir reconnaître une relation : contrairement au travail sur les colonnes, le titre du tableau n'est pas utilisé pour augmenter la couverture mais la précision (il permet de discriminer entre deux relations ayant le même type résultat et le même nombre de colonnes reconnues) ;
5. instanciation des relations sur chaque ligne du tableau, à travers l'instanciation de chaque type de la signature de ces relations. Pour les types symboliques, ce travail s'appuie sur les calculs de similarité entre les termes utilisés dans le tableau et les termes de la hiérarchie de chaque type symbolique de l'ontologie qui ont déjà été réalisés pour la reconnaissance du type des colonnes. Pour les types numériques, les valeurs numériques sont extraites et, dans le cas de valeurs multiples, leur sémantique est analysée pour construire le sous-ensemble flou correspondant au mieux à la signification des données dans le tableau.

Notre approche multi-critères dans l'annotation des tableaux se traduit par le calcul de scores (pour les différents types de données à utiliser pour annoter une colonne ou pour les différentes relations à utiliser pour annoter un tableau) qui sont la combinaison de différents scores calculés sur des critères distincts. Il sera possible dans l'avenir d'ajouter d'autres critères pour la reconnaissance des types de colonnes ou des relations afin de rendre notre méthode d'annotation encore plus robuste.

Nos premiers résultats expérimentaux sur 60 tableaux de données dans le domaine de la microbiologie alimentaire sont encourrageants. En effet, nous avons obtenu une précision de 98% pour la reconnaissance des colonnes numériques et symboliques. La reconnaissance du type de la colonne au niveau des colonnes symboliques se fait avec une précision de 93%. Nous avons obtenu une précision de 96% pour la reconnaissance du type des colonnes numériques. Ces bons résultats en termes de précision sont indispensables dans notre système, où

la détermination de ces types est nécessaire à la reconnaissance des relations : toute erreur de classification d'une colonne est propagée à la reconnaissance des relations, même si notre méthode multi-critères atténue ce phénomène par la prise en compte du titre du tableau. Sur les aliments annotés avec l'ontologie Sym'Previus, pour 66% des termes du web, le terme de l'ontologie choisi par l'annotation manuelle se trouve parmi les 5 premiers termes proposés par l'annotation automatique. Ces résultats sont encourageants, mais pas encore suffisants pour une réelle adoption du système par les utilisateurs. De même, pour l'annotation des relations dans les tableaux, nous obtenons une bonne couverture (95%) mais au détriment de la précision (69%).

Cette première évaluation expérimentale nous a donc permis de valider la pertinence de notre méthode d'annotation de tableaux, tout en reconnaissant certaines faiblesses en termes de précision au niveau de l'annotation des données symboliques et de la reconnaissance des relations. Nous proposons de pallier ces faiblesses grâce à une amélioration de la méthode d'annotation en elle-même (quelques pistes pour une telle amélioration sont présentées dans le paragraphe suivant), mais également grâce à la validation manuelle des données annotées. Dans le cadre du projet WebContent, nous comptons réaliser une évaluation expérimentale de plus grande ampleur, toujours avec la microbiologie alimentaire comme domaine d'application, mais nous avons également pour projet d'évaluer notre méthode dans d'autres domaines, tels que la contamination chimique des aliments ou l'aéronautique, afin de valider la généricité de notre approche.

Nous envisageons plusieurs pistes de travail pour améliorer la qualité de notre méthode d'annotation. Une des possibilités est l'utilisation de critères supplémentaires dans les scores permettant de déterminer le type d'une colonne symbolique, par exemple en combinant plusieurs mesures de similarité entre les termes du web et les termes de l'ontologie, mais également en prenant en compte des méthodes d'apprentissage, qui pourraient être entraînées à l'aide des annotations qui auraient été validées manuellement. Nous pensons également que la prise en compte du texte référençant le tableau peut apporter de bons résultats, tant pour aider à la reconnaissance des relations représentées par le tableau que pour trouver les valeurs sur certains types faisant partie de la signature des relations, non mentionnées dans le tableau lorsqu'il s'agit de valeurs constantes.

La validation manuelle des annotations est un enjeu majeur pour l'adoption de notre système : en effet, les utilisateurs n'accepteront le système que s'il donne des résultats de bonne qualité, mais la validation manuelle ne doit pas devenir une tâche tellement pénible qu'elle retire tout intérêt au système. Ceci pose l'épineux problème de la validation des annotations, qui constitue une deuxième perspective à notre travail. Nous envisageons de mettre en place un système de validation au moment de la requête. Un utilisateur se verrait présenter des résultats ordonnés selon leur pertinence à la requête : il pourrait alors, au choix, naviguer parmi les résultats à sa requête sans faire de travail de validation, ou bien valider au fur et à mesure les résultats obtenus pour sa requête. Nous présentons ici quelques

pistes pour une telle validation :

- au niveau des relations, l'utilisateur pourrait ainsi confirmer l'annotation (« ce tableau représente en effet la relation sur laquelle j'ai interrogé ») ou bien infirmer l'annotation (« ce tableau ne représente pas la relation sur laquelle j'ai interrogé »). Dans ce cas, il pourrait au choix se contenter de signaler une annotation fausse, ou bien aller plus loin dans la validation et donner quelles sont les relations de l'ontologie que représente le tableau. Il s'agit là d'une opportunité pour faire évoluer l'ontologie : c'est en effet à ce moment là que l'utilisateur pourrait signaler à l'administrateur qu'il manque une relation dans l'ontologie (et éventuellement les types de données correspondants) pour représenter la signification d'un tableau ;
- au niveau des colonnes, nous considérons que la validation est incluse dans la validation des relations ;
- au niveau des données symboliques, l'utilisateur pourrait également confirmer l'annotation au moment de la requête : nous avons proposé dans le chapitre 3 de travailler sur les 5 premiers termes de l'ontologie proposés pour l'annotation d'un terme dans une cellule du tableau. Sur chacune des instances de type symbolique ramenée par l'interrogation sur une relation, l'utilisateur pourrait examiner ces 5 premiers termes et choisir, pour chacun d'entre eux, s'il s'agit d'un terme de l'ontologie correspondant exactement à la signification du terme du web (dans ce cas les autres termes de l'ontologie seraient éliminés de l'annotation du terme du web), s'il sagit d'un terme proche de la signification du terme du web (dans ce cas, ce terme serait conservé pour l'annotation, mais les autres termes proches également) ou s'il s'agit d'un terme qui ne correspond pas à la signification du terme du web (dans ce cas, il serait retiré de l'annotation du terme du web). Dans le cas où l'annotation produite par la validation ou la suppression des termes de l'annotation automatique ne serait pas satisfaisante, l'utilisateur pourrait parcourir l'ontologie pour désigner le terme de l'ontologie correspondant à la signification du terme du web, quitte à signaler à l'administrateur la nécessité d'ajouter un nouveau terme dans l'ontologie. Dans les cas où une correspondance exacte de signification entre le terme du web et un terme de l'ontologie serait signalée, cela pourrait permettre d'enrichir l'ontologie en synonymes, et ainsi d'améliorer la qualité des annotations ultérieures ;
- au niveau des données numériques, l'utilisateur pourrait également corriger les sous-ensembles flous générés.

Cette technique de validation permettrait d'améliorer les résultats au fur et à mesure de l'utilisation du système, en fonction de l'implication des utilisateurs. Cela permettrait en outre une mise à jour de l'ontologie à partir des données, ce qui est indispensable dans des domaines où il peut y avoir des thématiques émergentes.

Une troisième perspective de notre travail est l'interrogation des tableaux, en tenant compte du caractère flou des annotations générées. Nous avons présenté

en section 5.5.2 nos premières pistes en termes de comparaison des préférences d'un utilisateur avec les annotations floues.

Outre le problème de savoir comment comparer un à un les sous-ensembles flous dans les annotations avec les préférences d'un utilisateur, nous devons également réfléchir à la façon dont on peut combiner les différentes mesures d'adéquation des données à une requête en fonction des différentes sources d'imprécision et d'incertitude, afin d'apporter à un utilisateur les réponses les plus pertinentes possibles à sa requête. Les différentes sources d'imprécision et d'incertitude que nous avons recensées sont :

- incertitude sur la reconnaissance des relations représentées par le tableau. Le niveau de certitude que l'on a dans la reconnaissance d'une relation peut être évalué par la valeur du score final de la relation pour le tableau ;
- incertitude sur la reconnaissance du type des colonnes ;
- incertitude sur la reconnaissance des valeurs pour les données symboliques. Cette incertitude peut être mesurée grâce aux scores de similarité entre le terme du web et chaque terme de l'ontologie. Il s'agit de la source d'incertitude la plus grave dans notre système. En effet, pour que les utilisateurs adoptent le système, il faut que le cas de figure où ils interrogent sur un produit alimentaire (par exemple, fruits secs) et obtiennent des résultats sur un produit n'ayant aucun rapport (par exemple, saucisse sèche) soit le plus rare possible ;
- incertitude sur la reconnaissance des valeurs et la construction des sous-ensembles flous pour les données numériques ;
- imprécision initialement présente dans les tableaux pour les données numériques. Il s'agit du type d'imprécision qui était déjà traité dans le système MIEL : les résultats étaient ordonnés au choix de l'utilisateur en fonction de leur degré de possibilité d'adéquation ou de leur degré de nécessité d'adéquation avec la requête.

Pour chacune de ces incertitudes/imprécisions, une ou plusieurs mesures d'adéquation permettront de comparer la donnée et les préférences de l'utilisateur. Il n'est pas envisageable de proposer toutes ces mesures aux utilisateurs pour qu'ils choisissent eux-mêmes comment ordonner les résultats, car cela ferait trop de paramètres à gérer. Il faudra donc étudier le comportement de requête des utilisateurs pour proposer un (ou au maximum deux) indicateur(s) composite(s) de la qualité d'une réponse à une requête, afin d'ordonner au mieux les réponses suivant leur pertinence par rapport aux besoins des utilisateurs.

Les annotations générées par notre système sont représentées en format RDF. Or, il existe un lien direct entre les triplets RDF et le formalisme des graphes conceptuels [Corby et al., 2000]. L'extension des graphes conceptuels à la représentation de données floues sous la forme de distributions de possibilité, ainsi que leur interrogation, a été étudiée dans [Thomopoulos, 2003]. Une de nos perspectives de recherche est de compléter cette extension pour tenir compte des sémantiques des sous-ensembles flous générés par l'annotation, ainsi que de l'in-

certitude sur les relations. Il serait alors possible de comparer les performances d'interrogation des tableaux annotés en utilisant soit des méthodes d'interrogation nativement construites pour interroger les graphes conceptuels, telles que celles fournies dans CoGITaNT [Genest & Salvat, 1998], soit des méthodes nativement construites pour interroger des annotations en RDF, telles que le moteur d'interrogation en SPARQL implémenté dans Jena [Jena, 2007].

Ce travail de thèse nous a amenés à produire une méthode d'annotation générique, applicable à de nombreux domaines, pour la communauté de recherche en informatique, tout en proposant un outil utilisable et appliqué pour la communauté de recherche en microbiologie alimentaire. Il nous semble qu'il nous faudrait également nous rapprocher de la communauté des documentalistes, que nous n'avons pas cotoyé durant ce travail de thèse mais qui nous paraît pouvoir apporter un point de vue intéressant, sur ce travail comme sur l'annotation de documents en général. En effet, le but du travail des documentalistes est de traiter l'information, dans un cadre d'analyse documentaire ou de veille, pour aider à sa meilleure diffusion : il s'agit notamment d'annoter les documents pour permettre un accès facilité par mots clefs. Il serait bon de s'inspirer des problématiques soulevées par cette communauté pour améliorer l'annotation automatique de documents.

L'annotation automatique de documents est rendue difficile par l'hétérogénéité des documents à annoter. Nous avons proposé des mesures de similarité entre termes pour pallier le problème de l'hétérogénéité de vocabulaire. Cependant, notre méthode d'annotation donnerait de bien meilleurs résultats si elle n'était pas tributaire de l'hétérogénéité de structure, qui nous a amenés à vouloir annoter des tableaux de format « invalide » (plusieurs mesures de types différents dans une même colonne, une même mesure répétée avec différentes unités dans une cellule, valeur constante de température précisée dans le texte plusieurs pages avant le tableau et non rappelée dans le titre du tableau, etc.). Une solution possible à ce problème serait de proposer des recommendations pour structurer les tableaux de façon à ce qu'ils soient plus facilement annotables, mais également plus facilement compréhensibles pour un lecteur humain. De telles recommendations seraient faciles à diffuser dans un contexte scientifique par le biais de *guidelines* proposées par les différents journaux : une fois l'habitude prise, il est probable que même les rapports de projets et autres documents non publiés dans des journaux respecteraient ces *guidelines*. De telles recommendations peuvent aussi se faire par le biais du W3C : on a vu par exemple apparaître avec le format XHTML les balises $< caption >$, $< thead >$ et $< tbody >$ pour structurer un tableau avec son titre, les entêtes de colonnes et les données proprement dites. Cette structuration, encore peu employée actuellement car l'ancienne structure de tableau du format HTML est toujours valide, devrait se répandre dans les prochaines années au fur et à mesure que les auteurs du web se formeront à l'utilisation du format XHTML et que les éditeurs graphiques de pages web prendront en compte ces

efforts de structuration, non plus basée uniquement sur l'apparence d'une page mais aussi sur la signification des données dans la page.

Plus généralement, si des règles d'organisation des documents pour la publication sur le web étaient adoptées de façon universelle, la tâche d'annotation sémantique serait simplifiée. Contraindre les auteurs de contenu web à respecter une certaine structure facilite l'annotation, mais peut également rendre plus aisée la navigation pour les lecteurs, sans pour autant entraver la liberté d'expression sur le web. En effet, les sites dont le contenu est éditable mais dont la structure est cadrée par un certain nombre de règles, tels que les blogs ou l'encyclopédie Wikipedia, sont faciles à lire et à éditer, et rares sont les auteurs qui se plaignent des contraintes de structure imposées par ces sites.

Bibliographie

[AgroVoc, 2007] AgroVoc (2007). Téléchargement en fichier plat. ftp ://ftp.fao.org/gi/gil/gilws/aims/kos/agrovoc_formats/tagtext.

[Antoniou et al., 2005] Antoniou, G., Franconi, E., & van Harmelen, F. (2005). Introduction to semantic web ontology languages. In N. Eisinger & J. Małuszyński (Eds.), *Reasoning Web, Proceedings of the Summer School, Malta, 2005*, Lecture Notes in Computer Science : Springer-Verlag.

[Arslan & Egecioglu, 2000] Arslan, A. N. & Egecioglu, O. (2000). Efficient algorithms for normalized edit distance. *Journal of Discrete Algorithms, Special Issue on Matching Patterns*, (pp. 3–20).

[Baumgartner et al., 2001] Baumgartner, R., Flesca, S., & Gottlob, G. (2001). Visual web information extraction with Lixto. In *VLDB '01 : Proceedings of the 27th International Conference on Very Large Data Bases* (pp. 119–128).

[Baziz et al., 2006] Baziz, M., Boughanem, M., Prade, H., & Pasi, G. (2006). *Fuzzy Logic and the Semantic Web*, chapter A Fuzzy Logic Approach to Information Retrieval Using an Ontology-based Representation of Documents, (pp. 363–377). Capturing Intelligence. Elsevier.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34–43.

[Bick, 2004] Bick, E. (2004). Parsing and evaluating the French Europarl corpus. In *Méthodes et outils pour l'évaluation des analyseurs syntaxiques Journée ATALA* (pp. 4–9). Paris, France.

[Blanchard et al., 2005] Blanchard, E., Harzallah, M., Briand, H., & Kuntz, P. (2005). A typology of ontology-based semantic measures. In *Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Co-located with CAiSE'05 Conference* Porto, Portugal.

[Bouquet et al., 2004] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., & Stuckenschmidt, H. (2004). Contextualizing ontologies. *Web Semantics : Science, Services and Agents on the World Wide Web*, 1(4), 325–343.

[Buche et al., 2006] Buche, P., Dibie-Barthélemy, J., Haemmerlé, O., & Hignette, G. (2006). Fuzzy semantic tagging and flexible querying of xml documents extracted from the web. *Journal of Intelligent Information Systems*, 26(1), 25–40.

[Buche & Haemmerlé, 2000] Buche, P. & Haemmerlé, O. (2000). Towards contextual fuzzy querying of both structured and semi-structured imprecise data. In *Proceedings of the 4th International Conference on Flexible Query Answering Systems, FQAS 2000, Advances in Soft Computing* (pp. 362–375). Warsaw, Poland : Physica Verlag.

[Buche et al., 2003] Buche, P., Haemmerlé, O., & Thomopoulos, R. (2003). Integration of heterogeneous, imprecise and incomplete data : an application to the microbiological risk assessment. In *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS'2003*, volume 2871 of *Lecture Notes in Artificial Intelligence* (pp. 98–107). Maebashi, Japan : Springer.

[Chinchor & Marsh, 1998] Chinchor, N. & Marsh, E. (1998). MUC-7 information extraction task definition. In *Message Understanding Conference Proceedings*. http ://www-24.nist.gov/related_projects/muc/proceedings/ie_task.html.

[Cilibrasi & Vitanyi, 2007] Cilibrasi, R. L. & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), 370–383.

[Ciravegna, 2001] Ciravegna, F. (2001). Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)* Seattle.

[Ciravegna, 2003] Ciravegna, F. (2003). $(LP)^2$*, Rule Induction for Information Extraction using Linguistic Constraints*. Technical report, University of Sheffield. Technical Report no CS-03-07.

[Ciravegna, 2004] Ciravegna, F. (2004). Amilcare : adaptive IE tool. http ://nlp.shef.ac.uk/amilcare/.

[Cliver et al., 2004] Cliver, D. O., Hajmeer, M. N., & Jay-Russell, M. (2004). Foodborne infections and intoxications. Document de cours, PHR 150, School of Veterinary Medicine, University of California.

[Corby et al., 2000] Corby, O., Dieng, R., & Hébert, C. (2000). A conceptual graph model for W3C Resource Description Framework. In *ICCS '00 : Proceedings of the Linguistic on Conceptual Structures* (pp. 468–482).

[Cunningham et al., 2005] Cunningham, H., Bontcheva, K., Tablan, V., Maynard, D., Saggion, H., Peters, W., Aswani, N., Li, Y., Roberts, I., Funk, A., Shafirin, A., Aswani, S., Sun, H., Damljanovic, D., & Agatonovic, M. (2005). GATE : a General Architecture for Text Engineering. http ://gate.ac.uk/.

[Cunningham et al., 2002] Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE : A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

[Decagon Devices, Note] Decagon Devices (Application Note). Water activity for meat and poultry jerky.

[Didion & Barton, 2003] Didion, J. & Barton, G. (2003). JWNL project summary . http ://sourceforge.net/projects/jwordnet/.

[Dubois & Prade, 1988] Dubois, D. & Prade, H. (1988). *Possibility theory- An approach to computerized processing of uncertainty*. Plenum Press, New York.

[Dubois & Prade, 1997] Dubois, D. & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2), 141–150.

[e.dot, 2005] e.dot (2005). E.dot : Entrepôts de données ouverts sur la toile. http ://gemo.futurs.inria.fr/projects/edot/.

[Embley et al., 2002] Embley, D. W., Tao, C., & Liddle, S. W. (2002). Automatically extracting ontologically specified data from html tables of unknown structure. In *Conceptual Modeling - ER 2002 : 21st International Conference on Conceptual Modeling* (pp. 322–337). Tampere, Finland.

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., & Yates, A. (2005). Unsupervised named-entity extraction from the web : an experimental study. *Artificial Intelligence*, 165(1), 91–134.

[Fellbaum, 1998] Fellbaum, C., Ed. (1998). *WordNet : An Electronic Lexical Database*. MIT Press.

[Frank et al., 2006] Frank, E., Hall, M., & Trigg, L. (2006). Weka software homepage. http ://www.cs.waikato.ac.nz/ml/weka.

[Freitag & Kushmerick, 2000] Freitag, D. & Kushmerick, N. (2000). Boosted wrapper induction. In *Proceedings of the 17th National Conference on Artificial Intelligence AAAI-2000* (pp. 577–583).

[Genest & Salvat, 1998] Genest, D. & Salvat, E. (1998). A platform allowing typed nested graphs : How cogito became cogitant (research note). In *ICCS* (pp. 154–164).

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220.

[Guarino, 1998] Guarino, N. (1998). Formal ontology and information systems. In *Proceedings of FOIS'98* (pp. 3–15). Trento, Italy.

[Guetari et al., 1998] Guetari, R., Quint, V., & Vatton, I. (1998). Amaya : an authoring tool for the web. In *Maghrebian Conference on Software Engineering and Artificial Intelligence* Tunis, Tunisia.

[Handschuh et al., 2002] Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-cream - semi-automatic creation of metadata. In *EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web* (pp. 358–372). Siguenza, Spain.

[Handschuh et al., 2001] Handschuh, S., Staab, S., & Maedche, A. (2001). Cream : creating relational metadata with a component-based, ontology-driven annotation framework. In *K-CAP '01 : Proceedings of the 1st international conference on Knowledge capture* (pp. 76–83). Victoria, British Columbia, Canada.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics* (pp. 539–545). Nantes, France.

[Hepple, 2000] Hepple, M. (2000). Independence and commitment : assumptions for rapid training and execution of rule-based POS taggers. In *ACL '00 : Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 278–277). Hong Kong : Association for Computational Linguistics.

[Hignette, 2005] Hignette, G. (2005). Etiquetage sémantique flou de documents xml. In *RJCIA 2005, 7es rencontres nationales des jeunes chercheurs en intelligence artificielle* (pp. 57–70). Nice, France.

[Hignette, 2007] Hignette, G. (2007). Annotation de tableaux de données guidée par une ontologie. In *3èmes Rencontres Inter-Associations (RIA's2007) Recherche et extraction d'information* Lyon, France.

[Hignette et al., 2006] Hignette, G., Buche, P., Dervin, C., Dibie-Barthélemy, J., Haemmerlé, O., & Soler, L. (2006). Fuzzy semantic approach for data integration applied to risk in food : an example about the cold chain. In *Proceedings of the 13th World Congress of Food Science and Technology, Food is Life* Nantes, France.

[Hignette et al., 2005] Hignette, G., Buche, P., Dibie-Barthélemy, J., & Haemmerlé, O. (2005). Fuzzy semantic annotation of xml documents. In *Proceedings of the CAiSE'05 WORKSHOPS - Vol. 1, The 17th Conference on Advanced Information Systems Engineering. DISWeb'05 International Workshop on Data Integration and the Semantic Web* (pp. 319–332). Porto, Portugal.

[Hignette et al., 2007a] Hignette, G., Buche, P., Dibie-Barthélemy, J., & Haemmerlé, O. (2007a). Annotation sémantique floue de tableaux guidée par une ontologie. In *RNTI-E-9 : Extraction et gestion des connaissances (EGC'2007) Volume II* (pp. 587–598). Namur, Belgique.

[Hignette et al., 2007b] Hignette, G., Buche, P., Dibie-Barthélemy, J., & Haemmerlé, O. (2007b). An ontology-driven annotation of data tables. In *Approaches and Architectures for Web Data Integration and Mining in Life Sciences (WebDIM4LS), WISE Workshops* Nancy, France. à paraître.

[Hignette et al., 2007c] Hignette, G., Buche, P., Dibie-Barthélemy, J., & Haemmerlé, O. (2007c). Semantic annotation of data tables using a domain ontology. In *Discovery Science* (pp. 253–258). Sendai, Japan.

[Hignette et al., 2007d] Hignette, G., Buche, P., Dibie-Barthélemy, O. C. J., Doussot, D., Haemmerlé, O., Mettler, E., & Soler, L. (2007d). Semantic annotation of web data applied to risk in food. In *5th International Conference Predictive Modelling in Foods (ICPMF 2007) Fundamentals, State of the Art and New Horizons* (pp. 155–158). Athens, Greece.

[Hogg et al., 2003] Hogg, D. G., Tan, A., Cusack, M. B., Beaton, S., Chai, M., Holland, G., Petalotis, A., & Subasinghe, N. (2003). Home delivery project report. The University of Melbourne.

[Jaccard, 1912] Jaccard, P. (1912). The distribution of flora in the alpine zone. *The New Phytologist*, 11(2), 37–50.

[Jena, 2007] Jena (2007). Jena - a semantic web framework for java. http ://-jena.sourceforge.net.

[Jiang & Conrath, 1997] Jiang, J. J. & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)* Taiwan.

[Kahan et al., 2002] Kahan, J., Koivunen, M.-R., Prud'Hommeaux, E., & Swick, R. (2002). Annotea : an open RDF infrastructure for shared web annotations. *Computer Networks*, 39, 589–608.

[Kiryakov et al., 2004] Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2.

[Kjærsgaard et al., 2007] Kjærsgaard, P. S., Bick, E., Schmid, H., & Stein, A. (1996–2007). VISL : Visual interactive syntax learning. http ://-beta.visl.sdu.dk/visl/fr/parsing/automatic/.

[Kobayashi & Takeda, 2000] Kobayashi, M. & Takeda, K. (2000). Information retrieval on the web. *ACM Computing Surveys*, 32(2), 144–173.

[Koh & Mui, 2001] Koh, W. & Mui, L. (2001). An information theoretic approach to ontology-based interest matching. In *Proceedings of the Second Workshop on Ontology Learning OL'2001, Held in conjunction with the 17th International Conference on Artificial Intelligence IJCAI'2001* Seattle, USA.

[Labský & Svátek, 2006] Labský, M. & Svátek, V. (2006). *Information Extraction with Presentation Ontologies.* Technical report, University of Economics, Prague.

[Lake et al., 2003] Lake, R., Hudson, A., Cressey, P., & Nortje, G. (2003). *Risk Profile : Campylobacter jejuni/coli in Poultry (whole and pieces).* Technical report, Institute of Environmental Science and Research Limited, Christchurch Science Centre, New Zealand.

[Lin, 1998] Lin, D. (1998). An information-theoretic definition of similarity. In *ICML '98 : Proceedings of the Fifteenth International Conference on Machine*

Learning (pp. 296–304). San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

[McDowell et al., 2003] McDowell, L., Etzioni, O., Gribble, S. D., Halevy, A., Levy, H., Pentney, W., Verma, D., & Vlasseva, S. (2003). Mangrove : Enticing ordinary people onto the semantic web via instant gratification. In *The SemanticWeb - ISWC 2003* (pp. 754–770). : Springer.

[Miller & team, 2005] Miller, G. A. & team (2005). WordNet 2.1 : Téléchargement pour Windows. http ://wordnet.princeton.edu/2.1/WordNet-2.1.exe.

[Miller & team, 2006] Miller, G. A. & team (2006). WordNet Search - 3.0. http ://wordnet.princeton.edu/perl/webwn.

[Neches et al., 1991] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Magazine*, 12(3), 36–56.

[Norton et al., 2005] Norton, B., Chapman, S., & Ciravegna, F. (2005). *The Semantic Web : Research and Applications*, chapter Orchestration of Semantic Web Services for Large-Scale Document Annotation, (pp. 649–663). Springer.

[Pivk et al., 2004] Pivk, A., Cimiano, P., & Sure, Y. (2004). From tables to frames. In *ISWC 2004 : Proceedings of the Third International Semantic Web Conference* (pp. 116–181). Hiroshima, Japan.

[Platt, 1999] Platt, J. C. (1999). *Fast training of support vector machines using sequential minimal optimization*, (pp. 185–208). MIT Press : Cambridge, MA, USA.

[Quinlan, 1993] Quinlan, J. R. (1993). Induction of decision trees. (pp. 349–361).

[Resnik, 1999] Resnik, P. (1999). Semantic similarity in a taxonomy : An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11, 95–130.

[Saïs, 2004] Saïs, F. (2004). Transformation d'informations structurées en documents XML guidée par une ontologie, rapport de stage de dea.

[Saïs et al., 2005] Saïs, F., Gagliardi, H., Haemmerlé, O., & Pernelle, N. (2005). Enrichissement sémantique de documents SML représentant des tableaux. In *Extraction et Gestion des Connaissances (EGC'2005) volume II - RNTI-E-3* (pp. 407–418). Paris, France.

[Seco et al., 2004] Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)* Valencia, Spain.

[Smith & Welty, 2001] Smith, B. & Welty, C. (2001). FOIS introduction : Ontology—towards a new synthesis. In *FOIS '01 : Proceedings of the International Conference on Formal Ontology in Information Systems* (pp. 3–9). Ogunquit, Maine, USA : ACM Press.

[Soergel et al., 2004] Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., & Katz, S. (2004). Reengineering thesauri for new applications : the AGROVOC example. *Journal of Digital Information*, 4(4). Article No. 257, 2004-03-17.

[Tenier et al., 2006] Tenier, S., Toussaint, Y., Napoli, A., & Polanco, X. (2006). Instantiation of relations for semantic annotation. In *International Conference on Web Intelligence* (pp. 463–472).

[Thomopoulos, 2003] Thomopoulos, R. (2003). *Représentation et interrogation élargie de données imprécises et faiblement structurées*. PhD thesis, Institut National Agronomique Paris-Grignon.

[Tijerino et al., 2005] Tijerino, Y. A., Embley, D. W., Lonsdale, D. W., Ding, Y., & Nagy, G. (2005). Towards ontology generation from tables. *World Wide Web*, 8(3), 261–285.

[U.S. Food and Drug Administration, 2001] U.S. Food and Drug Administration (2001). Evaluation and definition of potentially hazardous foods, chapter 3 : Factors that influence microbial growth. http ://www.cfsan.fda.gov/%7Ecomm/ift4-3.html.

[Van Rijsbergen, 1979] Van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.

[WebContent, 2007] WebContent (2007). WebContent, the semantic web platform. http ://www.webcontent.fr.

[Winkler, 1990] Winkler, W. (1990). String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association* (pp. 354–359).

[Yangarber et al., 2002] Yangarber, R., Lin, W., & Grishman, R. (2002). Unsupervised learning of generalized names. In *Proceedings of the 19th international conference on Computational linguistics* (pp. 1–7). Taipei, Taiwan : Association for Computational Linguistics.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and control*, 8, 338–353.

[Zadeh, 1978] Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1, 3–28.

[Zanibbi et al., 2004] Zanibbi, R., Blostein, D., & Cordy, J. R. (2004). A survey of table recognition : Models, observations, transformations, and inferences. *International Journal on Document Analysis and Recognition*, 7, 1–16.