Ontology-Aware Prediction from Rules: A Reconciliation-Based Approach

Fatiha Saïs[a], Rallou Thomopoulos[b,c]

[a]LRI, Paris Sud University & CNRS, Bât. Ada Lovelace, F-91190 Gif-sur-Yvette, France
[b]IATE Joint Research Unit, UMR1208, CIRAD-INRA-Supagro-Univ. Montpellier II, 2 place P. Viala, F-34060 Montpellier cedex 1, France
[c]INRIA GraphIK, LIRMM, 161 rue Ada, F-34392 Montpellier cedex 5, France

Abstract

Our work is related to the general problem of constructing predictions for decision support issues. It relies on knowledge expressed by numerous rules with homogeneous structure, extracted from various scientific publications in a specific domain. We propose a predictive approach that takes two stages: a reconciliation stage which identifies groups of rules expressing a common experimental tendency and a prediction stage which generates new rules, using both descriptions coming from experimental conditions and groups of reconciled rules obtained in stage one. The method has been tested with a case study related to food science and it has been compared to a classical approach based on decision trees. The results are promising in terms of accuracy, completeness and error rate.

Keywords: reasoning from knowledge, information integration, prediction, case-based reasoning, data reconciliation

1. Introduction

In the very last decades, extracting new knowledge from scientific publications has aroused great interest, in particular in experimental science domains, due to several converging circumstances and techniques: mass digitization of documents, web-enabled access to information, new experimental techniques allowing for high-throughput data acquisition, such as in genome sequencing for instance, but also new requirements for a higher and better-controlled production of goods. Indeed, the abundance of accessible scientific results both represents a real resource, and provides new needs for knowledge acquisition. This knowledge, once extracted from scientific publications, may be stored in a knowledge base. It can be exploited, among other uses, to answer user queries or to help for decision making issues.

However, one important problem of these knowledge bases is their incompleteness [1]. This incompleteness may be dealt: (i) by adapting the reasoning mechanisms for handling knowledge bases with omitted information [2]; (ii) by collecting new information from domain experts or from external sources like the World Wide Web [3]; or (iii) by using existing knowledge to predict unfilled information [4]. Our work falls in the third category. We propose a novel, case-based related approach for knowledge prediction that relies on reconciliation (which is a subfield of information integration).

Our application domain concerns food quality management in the cereal agrifood chain. Preliminary studies to this work were carried out on very different cases, outside the food science domain [5, 6]. They have the following characteristics:
(1) The knowledge base is composed of a set of causality rules with homogeneous structure made up from a collection of scientific publications. They express syntheses of published experimental studies, obtained and validated through repeated experimentations. These rules are used for prediction. However, there is a huge number of possible experimental conditions. Consequently the knowledge base is incomplete by nature, since only a limited part of the possible experimental con-

Email addresses: fatiha.sais@lri.fr (Fatiha Saïs), rallou.thomopoulos@supagro.inra.fr (Rallou Thomopoulos)

ditions have been explored in the literature and established as domain rules. Therefore, to make predictions concerning unexplored experimental conditions, a solution consists in using existing rules that concern close – although not identical – experimental conditions. In the more classic case where one starts from raw data, this approach is the principle of case-based reasoning.
(2) Although the rules concern distinct experimental conditions, they sometimes only differ by a small variation of one experimental parameter, which may be fundamental in the case of a highly discriminant parameter, but negligible for a parameter with low discriminance. Hence, rules which correspond to close experimental conditions and show similar results may be reconciled into groups of rules. Such groups have a semantics, since they express a common experimental tendency. They can also be exploited to reduce the search space in the prediction process. Performing this identification is a **reconciliation problem**. In addition to the experimental knowledge, general domain knowledge is available, and it has been modeled in an ontology. The ontology includes a vocabulary organized by subsumption, disjunction and synonymy relations. Moreover, it provides less common information concerning the status of concepts, such as functional dependencies and discriminance of concepts for prediction. Several existing methods aim at evaluating the similarity of data descriptions for various purposes (e.g. for prediction in case-based reasoning, for grouping into classes in classification, for detecting whether different data refer to the same real world entity in data reconciliation). Methods from data reconciliation [7, 5] are the most advanced ones: they take into account the logical semantics of an ontology, with particular attention to the functional dependencies.

The objective of this paper is to propose an approach to generate prediction rules relying on case-based and reconciliation methods, using an ontology. The approach we propose performs two stages that exploit the ontology:

- rule reconciliation into groups that express common experimental tendencies. From a mosaic of isolated pieces of knowledge, we identify the main experimental zones, which is also the experts' way of proceeding while analyzing an experimental domain of knowledge;
- computation of a prediction rule, starting from a new description of experimental conditions and from the "closest" group of reconciled rules.

Our method has been tested within a food science application concerning food quality management in the cereal agrifood chain and it has been compared to a classic predictive technique, using decision trees.

Not every predictive method may be used in the considered context. Experimental conditions have **missing values** (not all the parameters are described in each rule), use **both quantitative and qualitative** parameters (numerical and symbolic values), and are **scarce** since we are not in a high speed data application but in a scarce-knowledge context with only a few hundreds of rules available. Few methods are able to deal with all these three issues and they basically are case-based approaches or decision trees methods. This is the reason why we decided to compare our work against these approaches.

The paper is organized as follows: Section 2 describes the formalism used for ontology and domain rules representation. Section 3 gives an overview of related work in case-based reasoning, data reconciliation and decision tree prediction. Section 4 is dedicated to the proposed rule reconciliation method. Section 5 presents the proposed prediction method. Section 7 describes the context and related work in the application domain and proposes a comparative evaluation of the developed approach. Finally, Section 8 concludes the paper by giving some future work perspectives.

2. Preliminaries

In this section, we briefly recall essential elements regarding ontology and domain rule definition.

2.1. The domain Ontology

The ontology $\mathcal{O}$ is defined as a tuple $\mathcal{O} = \{\mathcal{C}, \mathcal{R}\}$ where $\mathcal{C}$ is a set of concepts and $\mathcal{R}$ is a set of relations.

2.1.1. *Ontology concepts*

Each concept c is associated with a definition domain by the def function. This definition domain can be:

- *numeric*, i.e. $def(c)$ is a closed interval $[min_c, max_c]$;
- *'flat' (non hierarchized) symbolic*, i.e. $def(c)$ is an unordered set of constants, such as a set of bibliographic references;
- *hierarchized symbolic*, i.e. $def(c)$ is a set of partially ordered constants, themselves are concepts belonging to $\mathcal{C}$.

In the sequel, we will refer to elements of concept domain definition by *values*.

2.1.2. *Ontology relations*

The set of relations $\mathcal{R}$ is composed of:

- the *subsumption* or 'kind of' relation denoted by $\preceq$, which defines a partial order over $\mathcal{C}$. Given $c \in \mathcal{C}$, we denote as $\mathcal{C}_c$ the set of sub-concepts of c, such that: $\mathcal{C}_c = \{c' \in \mathcal{C} | c' \preceq c\}$. When c is defined by hierarchized symbolic definition domain, we have $def(c) = \mathcal{C}_c$.
- the *equivalence* relation, denoted by $\equiv$, expressing a synonymy between concepts of the ontology.
- the *disjunction* relation between concepts, denoted by $\perp$. Given two concepts c and $c' \in \mathcal{C}$, $c \perp c' \Rightarrow (def(c) \cap def(c')) = \emptyset$. We note that the disjunction relation respects the subsumption relation. This means that if two general concepts c and c' are declared as disjoint then all the concepts that are more specific than c and c', respectively, are pairwise disjoint.

Figure 1 gives a small part of the set of concepts $\mathcal{C}$, partially ordered by the subsomption relation (pictured by '$\rightarrow$'). Examples of disjunctions are given apart for readability reasons. Note that the considered ontologies are not restricted to trees, they are general graphs. This is an important feature of our work with respect to previous approaches, such as [8], where only trees are considered.

2.1.3. *Least common subsumer*

Given two concepts c_1 and c_2, we denote as $lcs(c_1, c_2)$ their least common subsumer, that is $lcs(c_1, c_2) = \{c \in \mathcal{C} | c_i \preceq c, \text{ and } ((\exists c' \text{ s.t. } c_i \preceq c') \Rightarrow (c \preceq c')),\ i \in \{1, 2\}\}$.

For example, in the ontology of Fig. 1, the lcs of the concepts *LiposolubleVitamin* and *VitaminB* is the concept *Vitamin*. As commonly done, we consider a *Universal* concept subsuming all the other concepts of the ontology, to ensure that such a lcs always exists.

2.1.4. *Relationship between ontology concepts and experimental variables*

We consider a set of experimental descriptions containing K variables. Each variable X_k, $k = 1, \ldots, K$, is associated with a concept $c \in \mathcal{C}$ of the ontology $\mathcal{O}$. Each variable can be instantiated by a value that belongs to the definition domain of concept c.

2.1.5. *Variable discriminance*

For each variable X_k, a discriminance score, denoted by λ_k, is declared. It is a real value in the interval [0; 1]. It is obtained through an iterative approach performed with domain experts, as briefly explained in section 4.3 and in more detail in [9].

2.2. *Domain Rules*

Each domain rule expresses the relation cause-to-effect between a set of parameters (e.g. *KindOfWater*, *CookingTime*) of a unit operation belonging to the transformation process, and the variation of a given product property (e.g. variation of *vitamin content*). The set of domain rules is denoted by $\mathcal{R}$.

Definition 1 (Causality rule). *A causality rule $R \in \mathcal{R}$ is defined by a pair (H, C) where:*

- *$H = \{(X_1, v_1), \ldots, (X_h, v_h)\}$ corresponds to the rule hypothesis. It is a conjunction of variable/value criteria describing a set of experimental conditions in the form $(X_i = v_i)$. The value v_i may take numeric or symbolic (flat or hierarchized) values.*

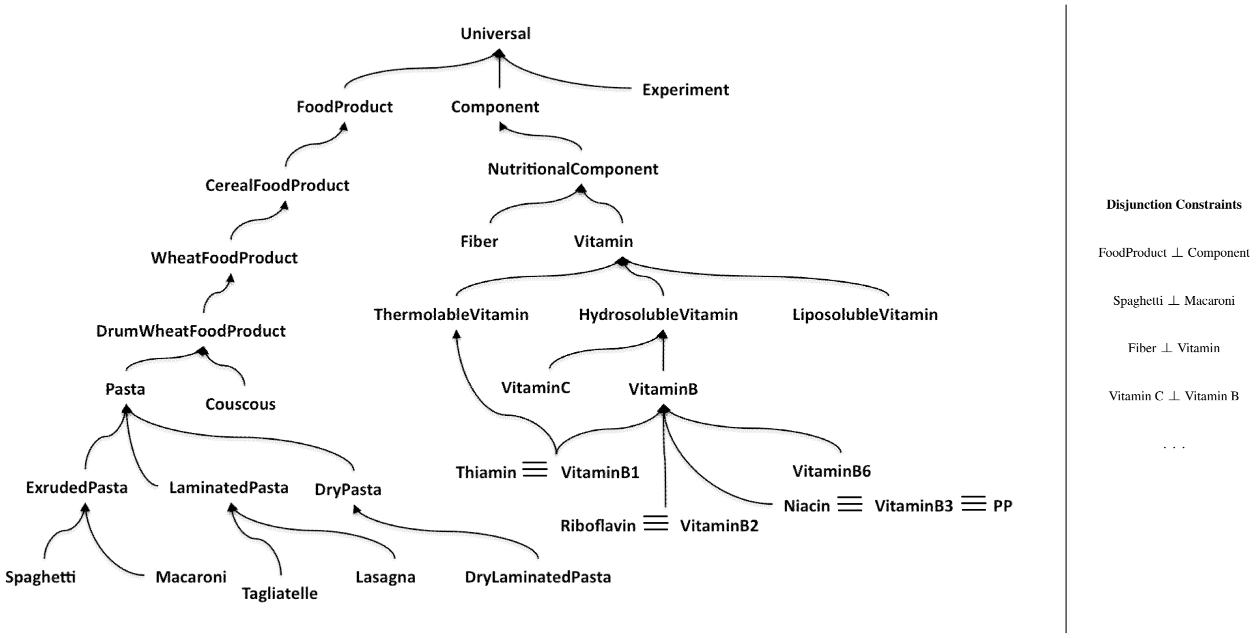


Figure 1: A part of the domain ontology

- *$C = (X_c, v_c)$ corresponds to the rule conclusion that is composed of a single variable/value effect describing the resulting impact on the considered property.*

It is interpreted by:

$$(X_1 = v_1) \wedge (X_2 = v_2) \wedge \ldots \wedge (X_h = v_h) \Rightarrow (X_c) = v_c)$$

Table 1 shows a part of the set of rules of the considered case study. The experimental variables are given in the first line and the values of these variables are given in the other lines of the table. The last column represents the conclusion part of the rules (i.e., the variable X_c and its values). For example, the rule $R1$ given in the second line is interpreted as follows:

(CookingTime = 12) $\wedge$ (KindOfWater = Tap Water) $\wedge$ (Component = Riboflavin)) $\Rightarrow$ (ConcentrationVariation = -53.3).

In the following, domain rules will be compared on the basis of the set of variables they have in common, called common description and defined as follows.

Definition 2 (Common description). *A common description between two causality rules R_1 and R_2, denoted by* Common-Desc(R$_1$, R$_2$)*, consists of the set of variables appearing at the same time in the set of variables/values of R_1 and R_2. Let H_1 and H_2 be the hypotheses of the rules R_1 and R_2 respectively. Let C_1 and C_2 be their conclusions.*
$CommonDesc(R_1, R_2) = \{X \mid \exists v_1, v_2,\ [(X = v_1) \in H_1\ and$
$(X = v_2) \in H_2]\ or\ [(X = v_1) = C_1\ and\ (X = v_2) = C_2]\}.$

We denote as *CommonDesc$_H$(R$_1$, R$_2$)* the common description of R_1, R_2 reduced to their hypotheses, i.e. *CommonDesc$_H$(R$_1$, R$_2$)* $= \{X \mid \exists v_1, v_2,\ [(X = v_1) \in H_1\ and\ (X = v_2) \in H_2]\}$.

3. Related Work

This section presents previous works in case-based reasoning and data reconciliation (in the framework of information integration) and provides a brief description about decision trees and some remarks.

3.1. Case-Based Reasoning

Case-based reasoning is a reasoning paradigm that relies on the adaptation of already solved problems, stored in a database, to solve new problems. This reasoning mechanism is not based on deduction, as rule-based reasoning, but on analogy [10].

Id.	CookingTime (min)	KindOfWater	Component		ConcentrationVariation (%)
R1:	12	Tap water	Riboflavin	⇒	-53.3
R2:	12	Tap water	Niacin	⇒	-45.6
R3:	12	Tap Water	Vitamin B	⇒	-45.6
R4:	24	Tap Water	Fiber	⇒	-45.6
R5:	13	Water	Vitamin B6	⇒	-46
R6:	10	Deionized water	Thiamin	⇒	-52.9
R7:	10	Deionized water	Vitamin B1	⇒	-51.8
R8:	15	Distilled water	Vitamin B2	⇒	-45.5

Table 1: *A set of causality rules*

A "case" is a description of a problem, associated with its solution. The two main steps of case-based reasoning are: (i) the "retrieve" step, which consists in finding, within the available cases stored in the database, the one(s) considered as most similar with the problem to solve; and (ii) the "reuse" step, which consists in adapting the solutions of the retrieved cases to solve the new problem.

Other complementary steps can be considered, as in [11], which is a reference paper in the domain. For instance, there may be a "revision" step, in which the generated solution is evaluated and possibly repaired. There is generally a "retainment" step, in which the solved problem is incorporated into the case base.

Various case-based reasoning systems have been implemented since the early works in this domain [12] and several approaches can be noticed. One of these approaches, called knowledge-intensive case-based reasoning in the literature [13, 14], consists in taking into account additional knowledge in the reasoning process. Two kinds of knowledge can be outlined: domain knowledge and adaptation knowledge. Domain knowledge defines concepts concerning the application domain, used in the case-based reasoning process, such as the domain vocabulary. Adaptation knowledge provides information useful for the evaluation of similar cases and for the adaptation of solutions, in for the "retrieve" and "reuse" steps.

Case-based reasoning has been considered as an alternative to rule-based systems, since it somehow reduces the knowledge acquisition process [15], since generally speaking cases are easier to obtain than rules. However this advantage is less relevant for knowledge-intensive case-based reasoning, where domain and/or adaptation knowledge has to be obtained and formalized. The work presented in [16], for instance, supports this point of view and proposes a formalization of domain and an adaptation knowledge using semantic web technologies.

Some works have proposed to combine case-based and knowledge-based reasoning methods. The main arguments found in literature for combining both approaches are: (i) achieving speedup learning and (ii) providing explanatory support to the case-based processes. For instance, in the CASEY system [17], the case-based method is used as first step, while the knowledge-based method (generally complex and slow), is used only in case of failure or the first step. Instead, in the BOLERO system [18], the case-based part provides meta-knowledge which is used to reduce the searching space used by rule-based part. The combination of the two approaches in used also in the CREEK system [14]mainly for providing explanatory support to the case-based processes.

In the present framework, the cases are not raw data but rules, since they express already consolidated knowledge obtained by synthesis and validation of experimental data. Therefore, the case- and knowledge-based parts are fused, which is not considered in the above works. Moreover, usually the domain and adaptation knowledge used in a case-based reasoning system are acquired through knowledge elicitation in the classic way for knowledge-based systems. Another option would be to also learn that knowledge from the cases. This line of work is adopted in this paper, since the reconciliation of the set of "cases" (the rules), which is part of the adaptation knowledge used in the prediction strategy, is learnt.

3.2. Data Reconciliation

Data reconciliation is one of the main problems encountered when different sources have to be integrated. It consists in deciding whether different data descriptions refer to the same real world entity (e.g. the same person or the same publication).

The traditional approaches of data reconciliation consider the data description as structured in several attributes. To decide on the reconciliation or on the non-reconciliation of data, some of these approches use probabilistic models [19, 20], such as Bayesian network or SVM. However, these probabilistic models need to be trained on labeled data. This training step can be very time-consuming what is not desirable in online applications. Indeed, in such online contexts, labeled data can not be acquired and runtime constraints are very strong. Alternative approaches have been proposed like [21] where the similarity measures are used to compute similarity scores between attribute values which are then gathered in a linear combination, like a weighted average. Although these approaches do not need training step, they however need to learn (starting from labeled data) some parameters as for example weights to be associated to similarity scores of the different attributes.

Recently, in the context of Linked Open Data, several tools, like [22], have been proposed to discover links between RDF data that are published on the Web, each with its own characteristics.

Among the existing methods of data reconciliation, only some recent ones [7, 5] take into account knowledge translating the logical semantics of the schema, in particular functional dependencies. Although several studies have considered the use of domain knowledge in classification, these concern fields where large amounts of data have to be treated, such as the semantic web [23] or image classification [24], with different issues, in particular the exploitation of topological knowledge [25] or of resource annotations [26]. In [25], the authors developed a query routing approach in a P2P network that is based on an aggregated view of the neighbors' content. This aggregated view is computed using a topological clustering algorithm. This is very different from the situation encountered in this paper, characterized by "poor data", i.e. relatively few rules are available and have to be exploited in the best way.

In [7, 5] a knowledge-based data reconciliation has been developed. The approach consists in a combination of two methods: the first method, called L2R, is logical and the second one, called N2R, is numerical. The two methods exploit knowledge that are declared in the schema that is expressed in OWL2. The exploited knowledge consists of a set of disjunction axioms between classes (e.g. Museum is disjoint with Painting), a set of (inverse) functional properties (e.g. a museum is located in at most one city) and the Unique Name Assumption. The logical method L2R is based on a set of logical inference rules that are generated from the translation of the logical semantics of the schema knowledge. The numerical method N2R is based on the computation of a similarity score for each part of data by using the values of the common properties. In the similarity computation, the functionality of the properties is exploited in a numerical way such that the similarity scores of the properties that are functional have a strong impact on the data pair similarity score. In both L2R and N2R a transitive closure is computed in order to obtain a partition of the set of reconciled data. Thus, each partition contains the set of data that refer to the same real world entity and two different partitions contain two data sets that refer to two different real world entities.

In the present work, two rules expressing the same (or very similar) experimental conditions (e.g. kind of water, cooking time) and results, may be considered as belonging to the same experimental tendency. Identifying domain rules that refer to the same experimental tendency is thus very close to the data reconciliation problem, and reconciliation techniques may prove to be suitable to this purpose. However, data reconciliation methods have to be adapted, since the reconciliation is no more applied on classical relational data, but on domain knowledge composed of causality rules enriched by ontological knowledge.

3.3. Decision Tree Prediction

It is worth noticing that a recent contribution [27] with similar objectives, also in a life science area, considered the same methods, case-based reasoning and decision trees. Several important differences compared to the present work can be highlighted. Firstly, [27] concerns a massive data application, opposed to a scarce-knowledge case here. Moreover, contrary to the proposed approach, it does not exploit ontological knowledge and aims to reduce interference with users, whereas we expect to ensure interpretability for users, in line with the results of [9]. Finally, it combines both case-based reasoning and decision tree in a sophisticated hybrid, fuzzy system, whereas we propose a reconciliation-based approach related to case-based reasoning, that we compare with decision-tree prediction.

Another significant approach combining expert knowledge and data is given in [28]. However it handles a classification problem and focuses on the optimization of rule fuzzy membership functions for predefined classes.

Decision trees are well established learning methods in supervised data mining. They can handle both classification and regression tasks. Compared to other methods, the advantages of decision trees are the descriptive/predictive capabilities, the simultaneous handling of numerical/symbolical features, the use of a similar formalism for regression and classification cases and finally the possibility of dealing with missing values.

In the tree structure, leaves represent classifications (or average values) of a dependent variable. Branches represent conjunctions of input features that lead to those classifications (or average values). The tree can be seen as a collection of rules based on values of the more discriminant variables in the modelling data set. Several implementations of decision trees are available in the literature, depending on the criteria used for building the tree. In this paper, we use C4.5 recursive dichotomous partitioning [29] in the R software [30] rpart implementation. Rules are selected based on how well splits determined using input features values can differentiate observations regarding the dependent variable. Once a rule is selected and splits a node into two, the same logic is applied to each "child" node (i.e. it is a recursive procedure). Splitting stops when C4.5 detects no further gain can be made, or some pre-set stopping rules are met. Each branch of the tree ends in a terminal node: Each observation falls into one and exactly one terminal node; each terminal node is uniquely defined by a set of rules.

A splitting criterion is used to decide which variable gives the best split at each node. Missing values are distributed over known values, which exploits missing values in an optimistic way.

In their descriptive form, decision trees summarize a set of descriptions. In their predictive form, once the trees have been learnt from the set of descriptions, they can be used to predict a value from a set of input parameters: the latter use will be considered in section 7.

4. Domain Rule Reconciliation

This section presents a method for domain rule reconciliation that partitions the set of rules by using techniques adapted from data reconciliation. We first give the general principle of rule partitioning approach. Then, we present how rules are filtered, compared and finally grouped into several groups of similar rules.

4.1. Principle

The general principle behind rule reconciliation consists in determining which rules can be considered as similar, thus providing a partition of the given set of rules. The partition is computed according to a specific similarity measure. Some rules are considered similar if their similarity value is above a certain threshold, otherwise they are called dissimilar. Similar rules are positioned in the same reconciliation group (partition subset).

Obviously, the key problem is related to the definition of an appropriate similarity measure which must take into account several features. In the literature, classic similarity measures are used for basic parameter values (as in [31]). The choice of these similarity measures is done according to the features of the values, e.g. symbolic/numeric, length of values, and so on. It is also possible to consider a semantic similarity measure that exploits the hierarchy distance of the values in a given domain ontology (see Wu and Palmer measure [32]). In a complementary approach, additional ontological knowledge, as synonymy relations between concepts, can be exploited. Finally, it is possi ble to compute the importance of the variables which appear in the rules, or take into account the declared knowledge, as functional dependencies or keys that can be discovered automatically as proposed in [33, 34].

4.2. Ontology-based filtering step

A pre-processing step relies on knowledge declared in the ontology. In this step, we exploit the semantics of the disjunction relations between concepts. For example, the concepts *ExtrudedPasta* and *LaminatedPasta* of Figure 1 are declared as disjoint. In this step we claim that two rules containing disjoint variable values cannot belong to the same group, and they are defined as "disjoint".

Definition 3 (Disjoint causality rules). *Two rules R_1 and R_2 are said to be disjoint if and only if:*

- *either $CommonDesc(R_1, R_2) = \emptyset$;*
- *or $\exists X \in CommonDesc(R_1, R_2)$ such that, given v_1, v_2 the respective values of X in R_1 and R_2, the disjointness constraint $v_1 \perp v_2$ can be inferred from the ontology.*

For example, in Table 1, the rule R4 is disjoint from all the other rules because - in the ontology - the value Fiber of the Component variable is declared as disjoint from the Vitamin(see Fig.1). All the other values taken by the $Component$ variable are sub-concepts of $Vitamin$ in the ontology, and therefore disjoint from $Fiber$ since the disjointness constraint is propagated to all the sub-concepts of $Vitamin$.

4.3. Variable relevance in rule similarity computation

When computing the similarity between rules, it is worth distinguishing between variables values according to the relevance of the variables. For example, to compute the similarity of two persons, the name, birth date and address are more relevant than their gender; and the birth date is more relevant than the address, which may change over the time. In our application domain, for instance, it is proper to consider that the *CookingTime* variable has a greater impact on pasta texture than the kind of method used to measure it.

In classical databases, the distinction of the importance of attributes may be obtained by exploiting the declaration of functional dependencies between attributes and from the key attributes. Furthermore, this distinction is binary, since it separates the very important attributes from those that are less important. However, in a knowledge base, which contains a collection of rules, it is more difficult to obtain such distinction among variables and expecting a binary evaluation of variable relevance can be a too strong assumption. Expected knowledge is more likely to provide an evaluation of the discriminance of variables.

Adopted approach.

In order to associate each variable X_k with a score λ_k reflecting its discriminance power, there are two possible strategies: (i) ask a domain expert to specify such knowledge or (ii) obtain it automatically by using a learning method. In this work, we adopted both approaches in a complementary way. Thus the computation of the score λ_k is adapted from the N2R method proposed in [5], but it is determined using both declared expert knowledge and supervised learning, in the following way:

- Functional dependencies are declared knowledge, obtained through a collaboration with domain experts. For instance, in the considered application, a subset of the variables describing the experimental conditions has been identified by the experts as determining the *ConcentrationVariation* variable, i.e. the one to be predicted. These variables are: *Temperature*, *SaltPercentage*, *CookingTime*, *KindOfWater*, *Component*, *AdditionOfIngredients*. The *FoodProduct*, *hasMethod*, and *ValueBefore* variables do not belong to this list. We propose to take into account this knowledge by associating a low discriminance score with the latter variables (the chosen weight is 0 in the practical evaluation of section 7, i.e. the variables are ignored).

- In order to distinguish between the variable importance among those variables that belong to functional dependancies an iterative method in interaction with experts is applied (see [9] for details).

4.4. Similarity measures

In order to compute the similarity of two causality rules, a prerequisite is to measure the similarity of the pairs of values, in an arbitrary order, belonging to their common descriptions.

4.4.1. Similarity measures between values

To measure the similarity between values, we use existing similarity measures. A similarity measure is usually defined as follows.

Definition 4 (Similarity measure). *Given a universe of discourse $\mathcal{U}$, a similarity measure Sim is a function $\mathcal{U} \times \mathcal{U} \to \mathbb{R}$ that satisfies the following properties:*

- *positivity:* $\forall v_1, v_2 \in \mathcal{U}, Sim(v_1, v_2) \geq 0$
- *symmetry:* $\forall v_1, v_2 \in \mathcal{U}, Sim(v_1, v_2) = Sim(v_2, v_1)$
- *maximality:* $\forall v_1, v_2 \in \mathcal{U}, Sim(v_1, v_1) \geq Sim(v_1, v_2)$.

Other properties may be required, like normalisation that will be assumed here, which imposes the Sim function to take values in the interval $[0; 1]$.

In [35] a survey and a deep analysis of similarity measures is presented, while in [31] experimental approaches are summarized. Since there is no universal measure which can be qualified as the most efficient for every kind of value, the choice of a suitable similarity measure depends on the features of the considered basic values, as for example, symbolic/numeric values or length of the values.

We present in details the choices we made in the practical evaluation section (see section 7 of this paper). Here, we outline that three cases may be distinguished, depending on the kind of the definition domain of the considered variables.

Hierarchized symbolic variables: a semantic similarity measure that exploits the hierarchy distance of the property values in the domain ontology may be used, as Wu and Palmer [32] and Lin [36]. A framework for similarity measures in ontologies is proposed by [37]. We retained the Wu and Palmer measure, which is more intuitive and easy to implement. Its principle is based on the length of the path between two concepts in the hierarchy. Its definition is as follows:

$$Sim(v_1, v_2) = \frac{2 * depth(v)}{depth_v(v_1) + depth_v(v_2)},$$

where v is the least common subsumer (lcs) of the values v_1 and v_2, $depth(v)$ is the length of the path (number of arcs) between v and the root node of the hierarchy, and $depth_v(v_i)$ is the length of the path between v_i and the root node through the node v.

For example, in the ontology of Figure 1, the semantic similarity of the values ($Spaghetti, Lasagna$) by using the Wu and Palmer measure is computed as follows:
$Sim(Spaghetti, Lasagna) = \frac{2*5}{7+7} = 0.71$.

Moreover we pose: $v_1 \equiv v_2 \Rightarrow Sim(v_1, v_2) = 1$.
For instance, $Sim(Thiamin, VitaminB1) = 1$ by hypothesis, since we have $Thiamin \equiv VitaminB1$.

'Flat' (non hierarchized) symbolic variables: numerous similarity measures have been proposed in the literature. One of the most exhaustive inventory is given in [38]. We use the Braun & Banquet similarity measure, which is defined as follows.
Let V_1 be the set of elements (in our case, words) composing v_1

and V_2 the set of elements composing v_2:

$$Sim_{Braun\&Banquet}(v_1, v_2) = \frac{|V_1 \cap V_2|}{max(|V_1|, |V_2|)}.$$

This measure is related to the well-known Jaccard coefficient: $Sim_{Jaccard}(v_1, v_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|}$. Both Braun & Banquet and Jaccard similarity measures take the value 0 when V_1 and V_2 have no common elements, and the value 1 when V_1 and V_2 are equal. However in intermediate cases $Sim_{Braun\&Banquet} \geq Sim_{Jaccard}$. For instance, for $v_1 =$ "$deionized\ water''$ and $v_2 =$ "$distilled\ water''$, we have $Sim_{Braun\&Banquet}(v_1, v_2) = \frac{1}{2}$ and $Sim_{Jaccard}(v_1, v_2) = \frac{1}{3}$. The mismatch between "*deionized*" and "*distilled*" is sanctioned twice in the Jaccard measure, since its denominator indicates the number of distinct words ("deionized", "distilled", "water"), whereas in the Braun & Banquet denominator the mismatch is only counted once, which we privileged here.

Numeric variables: a synthesis on similarity measures is presented in [39]. Let X be a numeric variable, v_1 and v_2 two values of $Range(X)$. The similarity measure we use is defined in a classical way, as the complement to 1 of a normalised distance:

$$Sim(v_1, v_2) = 1 - \frac{|v_2 - v_1|}{|Range(X)|}.$$

4.4.2. Similarity measures between two causality rules

In this work, the similarity between two rules is computed as a weighted average of the similarity scores of the values taken by their common variables. We denote as $Similarity(R_1, R_2)$ the real function which measures the similarity between two causality rules R_1 and R_2. It is computed as follows:

Definition 5 (Similarity of two causality rules).

$$Similarity(R_1, R_2) = \sum_k \lambda_k * Sim(v_{k_1}, v_{k_2}),$$

where $k \in [1\ ;\ |CommonDesc(R_1, R_2)|\ ]$, λ_k *is the discriminance power of the variable* $X_k \in CommonDesc(R_1, R_2)$, *and* v_{k_1}, v_{k_2} *are the values taken by* X_k *in* R_1 *and* R_2 *respectively.*

Illustrative example of two causality rules similarity. Let us consider $R1$ and $R5$ of Table 1, two rules which are not disjoint. To compute their similarity score we perform the following steps.

1. compute the common description : $CommonDesc(R1, R5) = \{CookingTime, KindOfWater, Component, ConcentrationVariation\}$;
2. determine the discriminance power of the considered variables. According to the assumptions of the subsection 4.3 all the considered variables have a high discriminance power. Therefore the scores λ_k equal to $1/4$;
3. compute the elementary similarity scores:
   $S_1 = Sim(12, 13) = 0.93$, by using the similarity measure between numeric values.
   $S_2 = Sim(Riboflavin, VitaminB6) = \frac{2*5}{6+6} = 0.83$, by using the Wu & Palmer measure.
   $S_3 = Sim(-53.3, -46) = 0.84$, by using the similarity measure between numeric values.
   $S_4 = Sim($"$Tap\ Water''$, "$Water''$$) = 0.5$ by using the Braun & Banquet similarity measure.

$$Similarity(R1, R5) = \tfrac{1}{4} * (S_1 + S_2 + S_3 + S_4) = 0.77.$$

4.5. Partitioning the set of rules into reconciliation groups

To decide which rules must be reconciled, we use a similarity threshold and assign to the same group each pair of rules having a similarity score beyond this given threshold. The threshold is obtained experimentally and its sensitivity is described in details in section 7. Rules having a similarity score beyond the threshold express a common experimental tendency.

We denote as $Reconcile(R_1, R_2)$ the predicate which expresses that the causality rules R_1 and R_2 express the same experimental tendency and must be reconciled. Let th be the similarity threshold for rule reconciliation. The reconciliation decision, for each pair of causality rules (R_1, R_2), is expressed as follows:

If $Similarity(R_1, R_2) \geq th$, then $Reconcile(R_1, R_2)$.

For example, the rules $R1$ and $R5$ (see their similarity score in the above section) will be reconciled if we consider a threshold th lower than 0.77.

We note that the threshold is fixed experimentally by exploiting the size and the homogeneity nature of the rule set (see Section 7). Finally, to obtain disjoint groups of rules, we compute a transitive closure for the set of reconciliation decisions previously computed. This choice, developed in [5], could be relaxed in the future by considering a fuzzy framework with overlapping groups. The transitive closure can be computed by using the following rule (TC):

$\forall R_1, R_2, R_3,\ \mathit{Reconcile}(R_1, R_2) \wedge \mathit{Reconcile}(R_2, R_3)$
$\Rightarrow \mathit{Reconcile}(R_1, R_3)$.

Algorithm 1 presents the reconciliation method. We can note that:

1. the result does not depend on the order of comparisons between rules;
2. the decision of reconciliation is based on pairwise comparisons between rules;
3. the algorithm does not guarantee a minimum number of rules per group, i.e. it is not excluded to end up with:
    - a rule group containing only a single rule, or
    - a single group for all the rules.

In Algorithm 1, we used several additional functions that can be defined as follows:

- the function $Disjoint(R_i, R_j)$ returns a boolean value indicating whether the rules R_i and R_j are disjoint or not, as defined in Definition 3;
- the function $Similarity_{\Lambda,\mathcal{S}}(R_i, R_j)$ computes the similarity between the rules R_i and R_j by using the discriminance scores Λ and the similarity measures $\mathcal{S}$, as defined in Definition 5;
- the function $TransitiveClosure(G)$ computes the transitive closure of G by applying the transitivity rule (TC) on the set of computed groups.

Algorithm 1: RECONCILIATION ALGORITHM
Input:

- $\mathcal{R}$: a set of causality rules, defined using the list of variables $\mathcal{X}$
- Λ: a list of discriminance scores associated with the list of variables $\mathcal{X}$
- $\mathcal{S}$: a list of similarity measures specified for the list of variables $\mathcal{X}$
- th: a threshold

Output: G: a set of groups of reconciled rules

```
$G \leftarrow \emptyset$
$PairSim \leftarrow 0$ // similarity between two rules
foreach $(R_i, R_j) \in \mathcal{R} \times \mathcal{R}$, with $(i \neq j)$
  if ($Disjoint(R_i, R_j)$ = false)
    $PairSim \leftarrow Similarity_{\Lambda,\mathcal{S}}(R_i, R_j)$
    if ($PairSim \geq th$)
      if ($\exists g_i \in G,\ R_i \in g_i$)
        $g_i \leftarrow g_i \cup \{R_j\}$
      else if ($\exists g_j \in G,\ R_j \in g_j$)
        $g_j \leftarrow g \cup \{R_i\}$
      else
        $g \leftarrow \{R_i, R_j\}$
        $G \leftarrow G \cup \{g\}$
$G \leftarrow TransitiveClosure(G)$
```

5. Predicting New Rules

This section presents the predictive step of the proposed method.

5.1. Principle

We consider G and R. G is the set G of reconciliation groups computed as presented in the previous section, and R is a new rule whose conclusion value is still unknown, and which has a set of variable/value criteria corresponding to its hypothesis. The new rule R is called an “unknown output rule”, denoted by “UO-rule”, and it is in the following form:

$R = (H, C)$ with $H = \{(X_1 = v_1), \ldots, (X_H = v_H)\}$ and $C = (X_c = x)$, where x is unknown.

The objective of this step is to predict the conclusion value x of the UO-rule R, that is, the expected variable/value effect, by comparing it to the existing known rules.

The method consists in selecting the closest reconciliation group and then use it to predict the conclusion value x. To perform this stage, R is compared to representative member(s) – called the "kernel" – of each reconciliation group. R is then assumed to be part of the group g whose kernel contains the rule the most similar to R. The value of the conclusion of R is predicted as a combination of the conclusion values of the rules of g.

5.2. Kernel Rule(s) of a Group

Both for combinatory and for semantic reasons, a *kernel* is computed for each reconciliation group, which corresponds to the set of rules that are most representative of the group. From a combinatory point of view, comparing the UO-rule R to the kernel of each group is of course simpler than comparing it to the whole set of rules. From a semantic point of view, the kernel can be seen as a characteristic sample of a reconciliation group. Aggregating multiple knowledge sources by retaining a well-chosen piece of them is a common feature in knowledge fusion [40]. An aggregation framework close to our approach, in a totally different domain, is proposed in [41].

The kernel of a group g is defined as follows.

Definition 6 (Kernel). *Let g be a group of rules and $R_i \in g$. The representativity of R_i is measured by:*

$$Repr(R_i) = \sum_{j=1\ (j \neq i)}^{j=|g|} Similarity(R_i, R_j).$$

The kernel of g is then defined by:

$$Kernel(g) = \{r \in g \mid Repr(r) = \max_{R_i \in g} Repr(R_i)\}.$$

The most representative rule is thus chosen (or several ones in case of *ex aequo*). Note that the set $Kernel(g)$ is often reduced to a unique rule.

5.3. Allocation of R to the closest reconciliation group

To compare R with the rule(s) of $Kernel(g)$, a similarity score is computed as in the Definition 5. The difference is that only the variable/value criteria appearing in the rule hypotheses are taken into account, since the conclusion value is unknown for R.

Definition 7 (Similarity between a UO-rule and a rule). *Let R be a UO-rule and R' a rule.*

$$Similarity(R, R') = \sum_k \lambda_k * Sim(v_k, v'_k),$$

where $k \in [1\ ;\ |CommonDesc_H(R, R')|\ ]$, λ_k is the discriminance power of the variable $X_k \in CommonDesc_H(R_1, R_2)$, and v_k, v'_k are the values taken by X_k in R and R' respectively.

R is then allocated to the group g whose kernel contains the rule the most similar to R.

Definition 8 (Group allocated to a UO-rule). *Let G be a set of reconciliation groups and R a UO-rule. R is allocated to a group $g \in G$, which satisfies the following condition:*

$\exists R' \in Kernel(g)$,
$Similarity(R, R') = \max_{r \in Kernel(g_i), i \in [1;|G|]} Similarity(R, r).$

5.4. Prediction Method

To predict the conclusion value x of R, two cases are distinguished, according to the nature – symbolic or numeric – of the conclusion variable X_c.

Symbolic case. It is the case where the definition domain of the considered variable is hierarchized or flat symbolic. The conclusion variable X_c may take several values in the reconciled rules of group g. We make the assumption that the value x to be predicted figures among those already appearing in the group g, i.e. that X_c does not take a new value in the conclusion of R.

For each distinct value v_i taken by X_c in the group g, a confidence degree $conf_{v_i}$ (a real value in [0;1]) is computed. It can be interpreted as the confidence in the prediction that "the value taken by X_c in R is v_i". It is defined as the ratio between the sum of similarity scores between R and the rules where X_c takes the value v_i, and the total sum of similarity scores between R and the rules of g, that is:

$$conf_{v_i} = \frac{\sum_{\{r \in g | X_c = v_i\}} Similarity(R, r)}{\sum_{\{r \in g\}} Similarity(R, r)}.$$

The value with the highest confidence provides the prediction of the conclusion value x:

$$(\tilde{x} = v) \ such \ that \ conf_v = \max_i \ conf_{v_i},$$

where $\tilde{x}$ denotes the value predicted for x.

Numeric case. The prediction of the value x taken by X_c in R is computed as a weighted mean of the values taken by X_c in all the rules of the group g. The weight associated with each rule r of g is the similarity score between R and r. Let v_r be the value taken by X_c in r, the value x is predicted by:

$$\tilde{x} = \frac{\sum_{r \in g}(Similarity(R, r) \times v_r)}{\sum_{r \in g} Similarity(R, r)}.$$

A confidence degree $conf$ is attached to the prediction. This confidence degree is the weighted mean of the similarity scores between R and the rules of g:

$$conf = \frac{\sum_{r \in g}(Similarity(R, r))^2}{\sum_{r \in g} Similarity(R, r)}.$$

Algorithm 2 presents the prediction method. For the sake of simplicity, only the numeric case is considered in the following algorithm.

Algorithm 2: PREDICTION ALGORITHM

Input:

- $G = \{g_1, \ldots, g_{|G|}\}$: a set of groups of reconciled rules
- $R = (H, C)$: a UO-rule with $H = \{(X_1 = v_1), \ldots, (X_H = v_H)\}$ and $C = (X_c = x)$, where the value x is unknown

Output: $(\tilde{x}, conf)$: with $\tilde{x}$ a prediction of the value x and $conf$ a confidence degree for the prediction $\tilde{x}$

```
$S \leftarrow 0$ // the best similarity between kernel rules and $R$
$g \leftarrow null$ // the selected group
foreach $\{g_i \in G\}$
  foreach $r \in Kernel(g_i)$
    if $(S < Similarity(R, r))$
      $S \leftarrow Similarity(R, r)$
      $g \leftarrow g_i$
$\tilde{x}\_num \leftarrow 0$ // numerator of $\tilde{x}$: weighted sum of the conclusion
    values of the rules of group $g$
$conf\_num \leftarrow 0$ // numerator of $conf$: sum of the squares of
    similarities between $R$ and the rules of group $g$
$denom \leftarrow 0$ // denominator of $\tilde{x}$ and $conf$ : sum of similarities
    between $R$ and the rules of group $g$
foreach $\{r = (h, c) \in g\}$, with $c = (X_c, v_c)$
  $s \leftarrow Similarity(R, r)$
  $\tilde{x}\_num \leftarrow \tilde{x}\_num + s \times v_c$
  $conf\_num \leftarrow conf\_num + s^2$
  $denom \leftarrow denom + s$
$\tilde{x} \leftarrow \tilde{x}\_num \div denom$
$conf \leftarrow (conf\_num \div denom)$
```

In Algorithm 2, we used several additional functions that can be defined as follows:

- the function $Kernel(g_i)$ selects the set of kernel rules of g_i, according to Definition 6;
- the function $Similarity(R, r)$ computes the similarity between the UO-rule R and the rule r, according to Definition 7;
- the function $Size(g_i)$ returns the number of rules in the group g_i.

5.5. Discussion

The similarity of the new rule R with the existing rules can vary a lot. It can be excellent, for instance if there is an exact correspondence between R and an existing rule, or lower if more distant rules have to be used. This is reflected by the confidence associated with the prediction.

To conclude on the whole approach, both reconciliation step and prediction step are based on the prior definition of similarities. Several levels of similarities are used in this paper, from similarity measures between basic values, to similarity measures between rules.

For the former (similarities between basic values), dozens of measures are proposed in the literature and we made the choice not to propose yet another one but to make a deep review of existing ones. For hierarchical and numerical values, our choices are among the most commonly used ones. However for flat symbolic numerical values, the chosen metric is little known and rarely used. We have given the justification of this choice in Section 4.4.1 to contribute to the analysis of the issue.

The latter (similarities between rules) are defined on the basis of similarities between basic values, but also take into account (via discriminance), both expert considerations and ontological knowledge such as functional dependencies, which is a contribution of the paper.

6. Complexity

Computing the complexity of any data-driven algorithm is a challenging task, because it is hard to model and capture all relevant characteristics of the data distribution (e.g. the number of unfilled attributes, the distribution of the values, data homogeneity).

In this section we, first, present the space complexity which can be significantly reduced thanks to the filtering step (presented in Section 4.2). Then, we give an estimation of time complexity for both reconciliation and prediction algorithms.

6.1. Space complexity

To compute all the comparisons between two rules, the reconciliation space is composed of the set of rule pairs built from $\mathcal{R} \times \mathcal{R}$. The problem can be reduced, since it only considers rules that are not disjoint.

Let n denotes the number of domain rules in $\mathcal{R}$. Without any filtering the reconciliation method has to perform $\frac{n*(n-1)}{2}$ comparisons.

By using disjunctions between rules, the number of comparisons is decreased by n'. To estimate n', let us consider the following cases:

- a disjunction $v_1 \perp v_2$ between two values of a given variable X. Let n_1 be the number of rules in $\mathcal{R}$ where $X = v_1$ and let n_2 the number of those where $X = v_2$.
- the common description is empty. Let n_3 be the number of rule pairs of $(\mathcal{R} \times \mathcal{R})$ where the common description is empty.

We obtain the value of n' as follows: $n' = n_1 + n_2 + n_3$. Then the obtained space complexity is of: $O(\frac{(n^2-n)}{2} - n')$.

6.2. Time complexity

The complexity of reconciliation algorithm depends on the number of rules $n =\mid \mathcal{R} \mid$. The algorithm applies first pairwise comparisons on the set of rules (Algorithm 1, lines 3–13). When no filtering step is used this step costs $((n*(n-1))/2)$. The similarity computation (line 5) is done in an insignificant time, because: (i) the process is dependent on the size of the common description of r_i and r_j, which is in the worst case of $max(\mid r_i \mid, \mid r_j \mid)$ and (ii) the number of compared attributes comparing to the number of rules is often negligible. The step of finding an existing group that contains g_i or g_j (lines 7–11) costs n in the worst case, i.e. when the size of g equals n. Finally, the transitive closure (line 14) is preformed by scanning all the groups, which costs n. Finally, we obtain a time complexity for the reconciliation algorithm of $O(\frac{(n^2-n)}{2} * n + n)$.

The complexity of the prediction algorithm increases linearly with the number of groups g_i (which equals n in the worst case), during the stage of allocation (Algorithm 2 lines 3–7) of R to the closest group. Indeed, within a given group, the use of the kernel limits the number of rule comparisons to one – in the general case – or very few – in the case where the kernel contains several rules, i.e. if several rules of the same group have exactly the same representativity.

The complexity also increases linearly with the size of the selected group g (which equals n in the worst case), during the stage of the prediction computation. Since the number of groups and their sizes are limited in practice, the complexity remains reduced. For the prediction algorithm, time complexity is of $O(n + n)$, in the worst case.

7. Practical Evaluation

In this section, we describe the application domain, we collocate our proposed approach with respect to those approaches already existing in the literature. Successively, we describe the adopted evaluation protocol, present the results providing a qualitative and quantitative evaluation.

7.1. Motivating Case Study: the Context

The efforts in cereal food design during the last 20 years resulted in an explosion of scientific papers which can hardly be used in practice because they have not been completely integrated into a corpus of knowledge. At the same time, cereal and pasta industry has evolved from traditional companies to a dynamic industry geared to follow consumer trends: healthy, safe, easy to prepare, pleasant to eat [42]. To meet such criteria, current industry requires knowledge from its own know-how as well as from different disciplines. Moreover, it has to extract the key information from the available knowledge in order to support decision.

Knowledge on the domain of transformation and properties of cereal-based food is available through the international literature of the domain concerning the impact of the transformation process on food properties. More precisely, the following elements are described:

- the “technical” information concern the unit operations which are involved in transformation from the wheat grains to the ready-to-use food (e.g. grinding, storage, drying, baking, etc.);
- the “property” information define the criteria which are used to represent the properties of products, according to three aspects: organoleptic, nutritional and safety properties (e.g. colors, vitamins contents, pesticides contents, etc.);
- the “result” information provide the impact of a unit operation on a property (i.e. what happens at the “intersection” between an element of the technical information and an element of the property information).

For each unit operation composing the transformation process, and for each family of product properties, this knowledge has been expressed as causality rules [43]. Approximately 600 rules are available in the application.

7.2. Previous Approaches in the Application Domain

Previous systems have been proposed in food science in order to help prediction. In particular, several works have dealt with the problem of food security, and therefore have proposed tools for risk assessment, such as in predictive microbiology, to prevent microbiological risk in food products. Such systems [44, 45, 46] combine a database with mathematical models which, applied to the data, allow one to deduce complementary information.

In the field of cereal transformation, we can also note the “Virtual Grain” project [see 47], which gathers heterogeneous information concerning the properties of cereal grains and their agronomical conditions in a database in order to identify potential relationships between properties that are usually studied separately: morphological, biochemical, histological, mechanical and technological properties. The database is connected to statistical and numerical computing tools, in particular a wheat grain cartography tool developed using matlab. Based on wheat grain properties and information on the components distribution, it proposes a representation of components content in each tissue. However, the objectives of the Virtual Grain project are related to the explanation of the grain behavior during fractioning. Contrary to our concern, they do not include considerations on food products, industry process analysis and final quality prediction.

In the close domain of breadmaking technology, the “Bread Advisor” tool has been a pioneer as a knowledge software for the baking industries [see e.g. 48]. This tool, exclusively based on expert knowledge stored in a database, provides three kinds of information: text information about the processing methods, list of possible faults and their causes, and generic messages about the effects of process changes. More recently, [49] have proposed a qualitative model of French breadmaking. However in both approaches knowledge from the scientific literature and dynamic prediction are not proposed.

The approach we propose has several original advantages from the application point of view. Firstly, concerning its objectives, it is not only a risk assessment system, but allows evaluating both defaults and qualities of food products. Secondly, although mostly tested on the durum wheat chain, it is not dedicated to a specific risk or chain, as it is not based on predetermined mathematical models, and therefore can be generalized to other application fields. Finally, our approach for prediction is not based on predetermined domain models. Indeed, such models are often unavailable. Thus an important contribution of our work is to provide prediction solutions in case of lack of models.

7.3. Evaluation Protocol

The evaluation was made in collaboration with food science experts. The proposed predictive method was compared with a classic decision tree prediction approach (see section 3.3). The methodology used to evaluate the proposed method followed six steps:

1. definition of requirements on the quantity and form of the test queries (i.e. UO-rules), so that the results are significant;
2. definition of requirements on the rule base, so that the results are interpretable;
3. definition of a set of test queries that satisfy the previous requirements;
4. definition of the parameters used by the methods;
5. execution of the set of test queries;
6. analysis of the results.

In the following we describe the procedure step by step.

7.3.1. Significance Conditions on the Test Queries

The significance conditions put on the test queries aimed at ensuring that the set of test queries is as heterogeneous and representative as possible. They are of six kinds:

- the set of test queries should cover at least thirty percent of the rule base entries (value chosen because a 70-30 ratio is often used in data exploration);
- the set of test queries should cover all branches of the ontology;
- the set of test queries should include specific rules (i.e. without missing data), as well as general rules (with missing data);
- the set of test queries should include rules that have an exact answer in the queried rule base, as well as rules that are absent from the queried rule base. The latter should cover cases where there is a close answer in the queried rule base, or no close answer in the queried rule base;
- the values used in the test queries should be taken from an external rule base, whose conclusion values are known, so that the results can be objectively evaluated;
- several thresholds should be tested.

7.3.2. Interpretability Conditions on the Rule Base

The interpretability conditions put on the rule base were the following: several (at least two) different rule bases should be used, in order to introduce a variability in the rule base content. Considering its impact on the method results will allow testing the robustness of the method.

7.3.3. The Set of Test Queries

The test queries are defined as UO-rule hypotheses. The objective is to predict their conclusion values. Considering the above conditions, the test queries presented in Table 2 were defined.

7.3.4. Method Parameters

The predictive method proposed in this paper was used with two different thresholds for partitioning, respectively 0.8 and 0.9.

The implementation used for decision trees is the R software with the *rpart* package for CART trees. The parameters of the rpart algorithm are: cross validation = 100, minimum instances per leaf = 6 (default value).

7.3.5. Experimental Results

Each of these methods was executed on two different rule bases. Rule base 1 contains 109 rules. Rule base 2 contains 117 rules. All of them concern the "cooking in water" unit operation and the "vitamin content" and "mineral content" properties. The execution of the set of test queries gave the results presented in Table 3 (a). The queries were executed both using the reconciliation method presented in this paper, and using the decision tree method.

7.4. Result Analysis and Discussion

In Table 3 (b), the error rates obtained with each method are computed. In Figures 2 (a) and (b), these error rates are presented as histograms, respectively for rule base 1 and rule base 2. The left columns represent the error rates obtained with the proposed method, for the two tested thresholds. The right columns represent the error rate obtained with the decision tree method.

These results clearly show a lower error rate obtained with the proposed predictive method, in all cases.

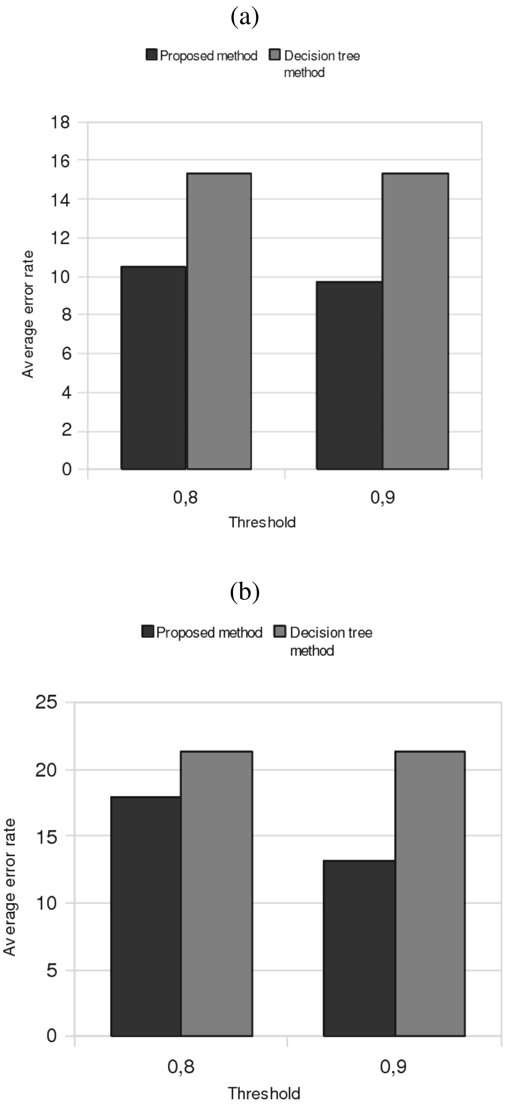


Figure 2: Average error rate for (a) rule base 1 and (b) rule base 2

The two tested thresholds were chosen experimentally. The obtained results tend to show that there is an optimum threshold of 0.9. The choice of the threshold impacts the number and size of the groups obtained in the reconciliation step of the method. These must neither be too numerous and small (as in the case of a high threshold) nor too few and large (as in the case of a low threshold).

With rule base 1, the number of obtained groups was: 14 for threshold 0.8, and 45 for threshold 0.9. With rule base 2, the number of obtained groups was: 5 for threshold 0.8, and 25 for threshold 0.9. When the obtained groups are few and large (threshold 0.8), we can notice that the obtained predictions are

Q1	{(FoodProduct=Spaghetti), (Component=Riboflavin), (ValueBefore= 0.6), (Temperature= 100), (SaltPercentage= 0), (Time= 12), (KindOfWater=Tap water), (Riboflavin addition=0.65), (Drying cycle=LT), (Drying time=28), (Max drying temperature=39)}
Q2	{(FoodProduct=Spaghetti), (Component=Riboflavin), (ValueBefore=0.4), (Temperature=100), (SaltPercentage=0), (Time=24), (KindOfWater=Tap water), (Riboflavin addition=0.65), (Drying cycle=HT-B), (Drying time=14), (Max drying temperature=70)}
Q3	{(FoodProduct=Spaghetti), (Component=Riboflavin), (ValueBefore=1.63), (Temperature=100), (SaltPercentage=0), (Time=12), (KindOfWater=Tap water), (Riboflavin addition=1.95), (Drying cycle=HT-B), (Drying time=14), (Max drying temperature=70)}
Q4	{(FoodProduct=Spaghetti), (Component=Thiamin), (ValueBefore=1), (Temperature=100), (SaltPercentage=0), (Time=12), (KindOfWater=Tap water), (Thiamin addition=0.96), (Drying cycle=HT-A), (Drying time=13), (Max drying temperature=75)}
Q5	{(FoodProduct=Spaghetti), (Component=Niacin), (ValueBefore=5.7), (Temperature=100), (SaltPercentage=0), (Time=24), (KindOfWater=Tap water), (Niacin addition=2.24), (Drying cycle=LT), (Drying time=28), (Max drying temperature=39)}
Q6	{(FoodProduct=Spaghetti), (Component=Niacin), (ValueBefore=9.6), (Temperature=100), (SaltPercentage=0), (Time=12), (KindOfWater=Tap water), (Niacin addition=6.72), (Drying cycle=LT), (Drying time=28), (Max drying temperature=39)}
Q7	{(FoodProduct=Spaghetti), (Component=Niacin), (ValueBefore=9.6), (Temperature=100), (SaltPercentage=0), (Time=24), (KindOfWater=Tap water), (Niacin addition=6.72), (Drying cycle=HT-A), (Drying time=13), (Max drying temperature=75)}
Q8	{(FoodProduct=Macaroni), (Component=Thiamin), (ValueBefore=11.4), (Temperature=100), (Time=10), (KindOfWater=Deionized water)}
Q9	{(FoodProduct=Noodles), (Component=Folic acid), (ValueBefore=0.026), (Temperature=100), (SaltPercentage=0), (Time=14)}
Q10	{(FoodProduct=Macaroni), (Component=Vitamin B6), (ValueBefore=1.129), (Temperature=100), (SaltPercentage=0), (Time=14)}
Q11	{(FoodProduct=Pasta), (Component=Thiamin), (ValueBefore, 1.08), (Temperature, 100), (SaltPercentage, 0), (KindOfWater, Distilled deionized water)}
Q12	{(FoodProduct=Pasta), (Component=Riboflavin), (ValueBefore=0.43), (Temperature=100), (SaltPercentage=0), (KindOfWater=Distilled deionized water)}
Q13	{(FoodProduct=Pasta), (Component=Niacin), (ValueBefore=7.82), (Temperature=100), (SaltPercentage=0), (KindOfWater=Distilled deionized water)}
Q14	{(FoodProduct=Noodles), (Component=Phosphorous), (ValueBefore=201.2), (Temperature=100), (SaltPercentage=0), (Time=8), (KindOfWater=Tap water)}
Q15	{(FoodProduct=Noodles), (Component=Potassium), (ValueBefore=227.2), (Temperature=100), (SaltPercentage=0.5), (Time=8), (KindOfWater=Tap water)}
Q16	{(FoodProduct=Noodles), (Component=Calcium), (ValueBefore=27.2), (Temperature=100), (SaltPercentage=0), (Time=8), (KindOfWater=Tap water)}
Q17	{(FoodProduct=Noodles), (Component=Magnesium), (ValueBefore=56.6), (Temperature=100), (SaltPercentage=0.5), (Time=8), (KindOfWater=Tap water)}
Q18	{(FoodProduct=Noodles), (Component=Iron), (ValueBefore=3.4), (Temperature=100), (SaltPercentage=0), (Time=8), (KindOfWater=Tap water)}
Q19	{(FoodProduct=Noodles), (Component=Manganese), (ValueBefore=0.7), (Temperature=100), (SaltPercentage=0.5), (Time=8), (KindOfWater=Tap water)}
Q20	{(FoodProduct=Noodles), (Component=Zinc), (ValueBefore=1.6), (Temperature=100), (SaltPercentage=0), (Time=8), (KindOfWater =Tap water)}

Table 2: Set of test queries

(a)

Query	Rule base 1					Rule base 2					Expected
	Proposed method				Decision trees	Proposed method				Decision trees	
	Threshold 0.8		Threshold 0.9			Threshold 0.8		Threshold 0.9			
	pred.	conf.	pred.	conf.		pred.	conf.	pred.	conf.		
Q1	-57.16	0.72	-66.71	0.91	-57	-41.65	0.65	-43.1	0.82	-54	-53.3
Q2	-58.58	0.72	-60.05	0.89	-64	-41.77	0.65	-44	0.81	-49	-62.5
Q3	-57.32	0.73	-59.83	0.88	-56	-41.69	0.65	-43.26	0.81	-60	-50.9
Q4	-56.44	0.78	-64.3	0.88	-57	-42.56	0.69	-43.01	0.81	-49	-54
Q5	-47.98	0.79	-45.6	0.93	-63	-42.03	0.57	-58.14	0.81	-48	-42.1
Q6	-47.95	0.83	-53.1	0.93	-55	-42.48	0.57	-57.35	0.79	-67	-57.3
Q7	-48.23	0.85	-50.04	0.89	-61	-43.06	0.59	-58.18	0.75	-47	-59.4
Q8	-40.51	0.52	-43.93	0.94	-45	-42.72	0.62	-53.73	0.68	-35	-42.2
Q9	-23.46	0.81	-21	0.86	-37	-41.65	0.66	-21	0.87	-13	-21
Q10	-47	0.85	-47	0.85	-40	-41.7	0.68	-35	0.97	-12	-44
Q11	-48.99	0.72	-49.2	0.94	-46	-42.65	0.69	-52.56	0.77	-38	-49.3
Q12	-56.33	0.67	-37	0.83	-46	-42.53	0.7	-52.3	0.77	-37	-40.2
Q13	-47.85	0.78	-35	0.83	-46	-43.37	0.63	-51.77	0.72	-39	-50
Q14	-63.64	0.84	-66.13	0.92	-67	-61.17	0.78	-67.73	0.88	-69	-69.53
Q15	-88.12	1	-88.12	1	-66	-84.16	0.83	-84.84	0.93	-60	-88.12
Q16	-62.24	0.53	-62.3	0.68	-67	-49.41	0.63	-44.29	0.88	-67	-51.84
Q17	-63.11	0.7	-61.16	0.91	-67	-46.87	0.65	-51.69	0.88	-47	-56.18
Q18	-62.41	0.71	-75.68	0.69	-67	-74.28	0.55	-66.36	0.89	-67	-70.59
Q19	-63.13	0.74	-55.56	0.92	-66	-46.81	0.66	-50	0.89	-36	-57.14
Q20	-62.33	0.71	-64.71	0.69	-67	-55.55	0.61	-59.96	0.89	-78	-62.5

(b)

Rule base 1			Rule base 2		
Proposed method		Decision trees	Proposed method		Decision trees
Threshold 0.8	Threshold 0.9		Threshold 0.8	Threshold 0.9	
10.53 %	9.68 %	15.37 %	17.87 %	13.19 %	21.39 %

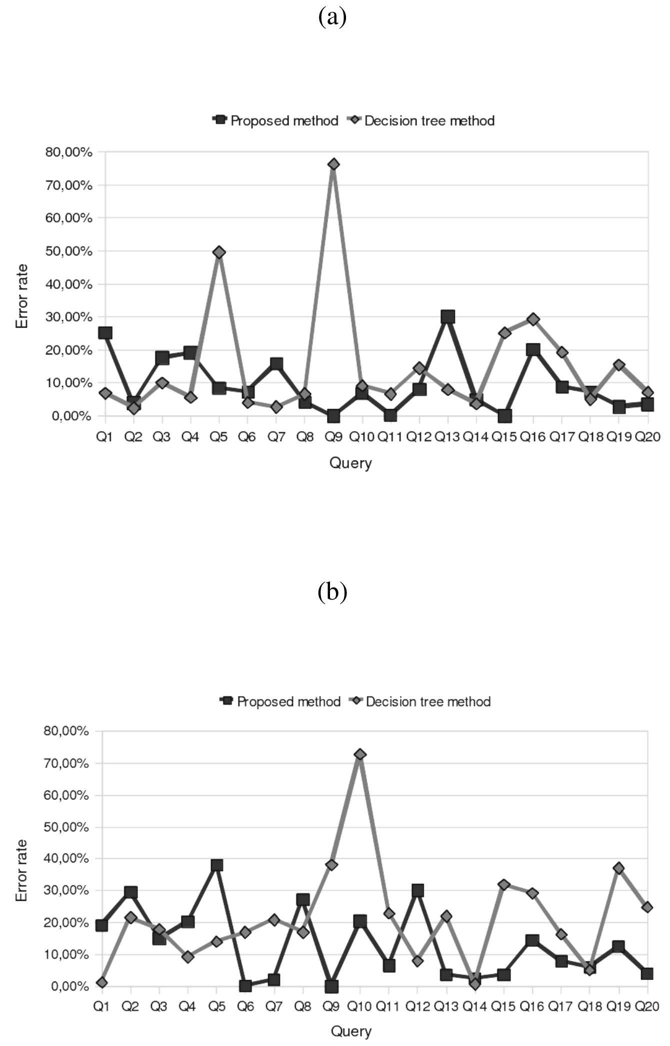
Table 3: (a) Execution results and (b) Average error rates, for the proposed method vs decision trees

more homogeneous among the tested queries. On the contrary, when the obtained groups are numerous and small (threshold 0.9), the obtained predictions are more various among the tested queries.

Figures 3 (a) and (b) present the error rates obtained query by query, with the proposed method for the optimum threshold 0.9, and with the decision tree method, respectively for rule base 1 and rule base 2.

We can make the following observation. The queries that obtained the most different results, if we compare both methods, are those for which exact or close answers were present in the rule base (such as for query $Q9$ in rule base 1), or those for which the closest answers, even if not so close, are quite different from the rest of the base and show different trends (such as for query $Q10$ in rule base 2).

The latter result is not very surprising since sensitivity to outliers is a well-known drawback of decision trees, and a strong point of our method which relies on the identification of common tendencies. Let us recall that our interest in (i) the case-based and (ii) the decision tree approaches is motivated, as previously mentioned, by specific features that are not handled by other methods (or not simultaneously), namely (i) missing values, (ii) both numerical and symbolic values and (iii) a limited number of cases (here rules). The proposed method thus takes the best from case-based and reconciliation approaches, moreover it is aware of ontological knowledge, and an improvement may legitimely be expected. Here we can note that the decision tree strategy, which processes step by step by considering each variable separately, turns out to be less relevant than the proposed method, which considers the rules globally, involving all the variables.



(a)

(b)

Figure 3: Error rates obtained for each query with (a) rule base 1 and (b) rule base 2

8. Conclusion and Future Work

This paper presented an approach to generate prediction rules relying on case-based and reconciliation methods, using an ontology. Rule reconciliation showed to have advantages from a semantic and from a computational point of view. From a semantic viewpoint, it constitutes a consolidation of the knowledge represented by the set of rules, since it highlights the expression of common experimental tendancies. From a computational viewpoint, it speeds up the performances of the method by providing a restriction of the search space. The use of ontological knowledge both participates in performance improvement, through the disjunction relation in particular, and allows taking into account variable relevance, through identified functional dependancies. The experimentation on the application domain of cereal food design has shown a real potential of the approach. Compared with the results obtained by decision tree prediction, the proposed approach is more complete and more accurate.

To deal with information incompleteness which appears in the condition part of the rules, different ways may be followed during the similarity computation between two rules: (i) by considering, in the common description, only the filled properties in both rules. This means that the missing values in one of the two rules are ignored and do not participate in the similarity computation; (ii) by considering, in the common description, the whole description, i.e. the filled and the unfilled properties. In this case, the missing values are exploited as a negative information since they decrease the similarity scores. It will be worth studying more deeply the impact of these choices on the prediction results.

A complementary ongoing work deals with the design and validation of a domain expertise. It aims at the cooperation of two kinds of information, heterogeneous by their granularity levels and their formalisms: expert statements expressed in a knowledge representation model and experimental data represented in the relational model. In such a framework, our prediction approach may be usefully introduced, in order to validate or invalidate the obtained predictions.

Finally, to demonstrate the generality of the approach, we plan to experiment it on other application domains, such as chemical and environmental risk.

9. References

[1] H. J. Levesque, Incompleteness in knowledge bases, SIGPLAN Not. 16 (1) (1981) 150–152. doi:http://doi.acm.org/10.1145/960124.806905.

[2] P. Buche, C. Dervin, O. Haemmerlé, R. Thomopoulos, Fuzzy querying of incomplete, imprecise, and heterogeneously structured data in the relational model using ontologies and rules, IEEE T. Fuzzy Systems 13 (3) (2005) 373–383.

[3] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, M. Houhou, Towards flexible querying of xml imprecise data in a dataware house opened on the web, in: FQAS, 2004, pp. 28–40.

[4] T. Gabaldon, M. Huynen, Prediction of protein function and pathways in the genome era 61(7-8) (2004) 930–944.

[5] F. Saïs, N. Pernelle, M.-C. Rousset, Combining a logical and a numerical method for data reconciliation, Journal of Data Semantics (JoDS) 12 (2009) 66–94, lNCS 5480.

[6] F. Saïs, R. Thomopoulos, Reference fusion and flexible querying, in: ODBASE-OTM Conferences (2), 2008, pp. 1541–1549.

[7] F. Saïs, N. Pernelle, M.-C. Rousset, L2r: A logical method for reference reconciliation, in: AAAI, 2007, pp. 329–334.

[8] J. Zhang, A. Silvescu, V. Honavar, Ontology-driven induction of decision trees at multiple levels of abstraction, Lecture Notes in Computer Science (2002) 316–323.

[9] R. Thomopoulos, S. Destercke, B. Charnomordic, I. Johnson, J. Abécassis, An iterative approach to build relevant ontology-aware data-driven models, Information Sciences 221 (2013) 452–472.

[10] J. P. Haton, M. T. Keane, M. Manago (Eds.), Advances in Case-Based Reasoning, Second European Workshop, EWCBR-94, Chantilly, France, November 7-10, 1994, Selected Papers, Vol. 984 of LNCS, Springer, 1995.

[11] A. Aamodt, E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–59.

[12] C. K. Riesbeck, R. C. Schank, Inside Case-Based Reasoning, Lawrence Erbaum Associates, Inc., Hillsdale, New Jersey, 1989.

[13] A. Aamodt, Knowledge-intensive case-based reasoning and sustained learning, in: ECAI, 1990, pp. 1–6.

[14] A. Aamodt, Knowledge-intensive case-based reasoning in creek, in: P. Funk, P. A. González-Calero (Eds.), ECCBR, Vol. 3155 of LNCS, Springer, 2004, pp. 1–15.

[15] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, 1993.

[16] M. d'Aquin, J. Lieber, A. Napoli, Case-based reasoning within semantic web technologies, in: J. Euzenat, J. Domingue (Eds.), AIMSA, Vol. 4183 of LNCS, Springer, 2006, pp. 190–200.

[17] P. Koton, Reasoning about evidence in causal explanations, in: AAAI, 1988, pp. 256–263.

[18] B. López, E. Plaza, Case-based planning for medical diagnosis, in: H. J. Komorowski, Z. W. Ras (Eds.), ISMIS, Vol. 689 of LNCS, Springer, 1993, pp. 96–105.

[19] I. P. Fellegi, A. B. Sunter, A theory for record linkage, Journal of the American Statistical Association 64 (328) (1969) 1183–1210.

[20] W. E. Winkler, Overview of record linkage and current research directions, Tech. rep., Statistical Research Division U.S. Census Bureau Washington, DC 20233 (2006).

[21] X. Dong, A. Halevy, J. Madhavan, Reference reconciliation in complex information spaces, in: ACM SIGMOD, ACM Press, 2005, pp. 85–96. doi:10.1145/1066157.1066168.
URL `http://dx.doi.org/10.1145/1066157.1066168`

[22] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, M. Wang, A framework for semantic link discovery over relational data, in: ACM CIKM Conference, Hong Kong, China, 2009, pp. 1027–1036.

[23] G. Stumme, A. Hotho, B. Berendt, Semantic web mining: State of the art and future directions, J. of Web Semantics 4 (2006) 124–143.

[24] N. Maillot, M. Thonnat, Ontology based complex object recognition, Image and Vision Computing 26 (2008) 102–113.

[25] C. Schmitz, A. Hotho, R. Jäschke, G. Stumme, Content aggregation on knowledge bases using graph clustering, in: Y. Sure, J. Domingue (Eds.), ESWC, Vol. 4011 of LNCS, Springer, 2006, pp. 530–544.

[26] N. Fanizzi, C. d'Amato, F. Esposito, A multi-relational hierarchical clustering method for datalogknowledge bases, in: A. An, S. Matwin, Z. W. Ras, D. Slezak (Eds.), ISMIS, Vol. 4994 of LNCS, Springer, 2008, pp. 137–142.

[27] C.-Y. Fan, P.-C. Chang, J.-J. Lin, J. Hsieh, A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification, Applied Soft Computing 11 (1) (2011) 632 – 644.

[28] W. Tang, K. Z. Mao, L. O. Mak, G. W. Ng, Adaptive fuzzy rule-based classification system integrating both expert knowledge and data, in: Proceedings of the 2012 IEEE 24th International Conference on Tools with Artificial Intelligence - Volume 01, ICTAI '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 814–821.

[29] L. Breiman, Classification and Regression Trees, Boca Raton: Chapman & Hall/CRC, 1984.

[30] R. Ihaka, R. Gentleman, R: A language for data analysis and graphics, Journal of Computational and Graphical Statistics 5 (3) (1996) 299–314.

[31] W. W. Cohen, P. Ravikumar, S. E. Fienberg, A comparison of string distance metrics for name-matching tasks., in: Proceedings of IJCAI-03 Workshop on Information Integration, 2003, pp. 73–78.

[32] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, 1994, pp. 133–138. doi:http://dx.doi.org/10.3115/981732.981751.

[33] N. Pernelle, F. Saïs, D. Symeonidou, An automatic key discovery approach for data linking, J. Web Sem. 23 (2013) 16–30.

[34] D. Z. Wang, X. L. Dong, A. D. Sarma, M. J. Franklin, A. Y. Halevy, Functional dependency generation and applications in pay-as-you-go data integration systems, in: WebDB, 2009.

[35] M.-J. Lesot, M. Rifqi, H. Benhadda, Similarity measures for binary and numerical data: a survey, Int. J. Knowl. Eng. Soft Data Paradigm. 1 (2009) 63–84.

[36] D. Lin, An information-theoretic definition of similarity, in: ICML, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 296–304.

[37] M. Ehrig, Similarity for ontologies - a comprehensive framework, in: in 13th European Conference on Information Systems, 2005.

[38] S. Choi, S. Cha, C. Tappert, A survey of binary similarity and distance measures, Journal of Systemics, Cybernetics and Informatics 8 (2010) 43–48.

[39] M.-J. Lesot, Similarity, typicality and fuzzy prototypes for numerical data, Journal Res-Systemica, Special issue on the 6th European Congress on Systems Science 5.

[40] D. Dubois, H. Prade, On the use of aggregation operations in information fusion processes, Fuzzy Sets and Systems 142 (1) (2004) 143–161.

[41] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, P. Marquis, On the merging of dung's argumentation systems, Artif. Intell. 171 (10-15) (2007) 730–753.

[42] G. Dalbon, D. Grivon, M. Pagnani, Continuous manufacturing process, in: J. Kruger, R. Matsuo, J. Dick (Eds.), Pasta and noodle technology, AACC, St Paul (MN-USA), 1996.

[43] R. Thomopoulos, J.-F. Baget, O. Haemmerlé, Conceptual graphs as cooperative formalism to build and validate a domain expertise, in: ICCS, 2007, pp. 112–125.

[44] M. Peck, T. Roberts, J. Sutherland, S. Walker, Modelling the growth, survival and death of microorganisms in foods: the UK Food Micromodel approach, International Journal of Food Microbiology 23 (1994) 265–275.

[45] J. Baranyi, M. Tamplin, A new international effort for the advance of predictive microbiology, ComBase: Combined database of microbial responses to food environments, in: ICPMF'03, Quimper, France, 2003, pp. 50–51.

[46] O. Haemmerlé, P. Buche, R. Thomopoulos, The miel system: Uniform interrogation of structured and weakly-structured imprecise data, J. Intell. Inf. Syst. 29 (3) (2007) 279–304.

[47] N. Mueangdee, F. Mabille, R. Thomopoulos, J. Abecassis, Virtual grain: a data warehouse for mesh grid representation of cereal grain properties, in: Proceedings of the 9th European Conference on Food Industry and Statistics, Agrostat'2006, Montpellier, France, 2006, pp. 291–299.

[48] L. Young, Application of Baking Knowledge in Software Systems, in: Technology of Breadmaking - 2nd edition, Springer, US, 2007, pp. 207–222.

[49] A. Ndiaye, G. D. Valle, P. Roussel, Qualitative modelling of a multi-step process: The case of french breadmaking, Expert Systems with Applications 36 (2, Part 1) (2009) 1020 – 1038.