2018

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

HDR / UNIVERSITÉ RENNES
sous le sceau de l'Université Bretagne Loire

pour l'obtention d'une

HABILITATION À DIRIGER DES RECHERCHES

Mention : Informatique (CNU-27)

présentée par

Grégory Smits

préparée à l'unité de recherche (IRISA)
Institut de Recherche en Informatique et Systèmes Aléatoires
IUT de Lannion

Personnalisation et enrichissement des méthodes d'accès aux données

Soutenue à Lannion
le 14 mars 2018

devant le jury composé de :

Anne LAURENT (rapporteur)
Gabriella PASI (rapporteur)
Henri PRADE (rapporteur)

Christophe MARSALA (examinateur)
Patrice BUCHE (examinateur)
Olivier PIVERT (examinateur)

Table des matières

Résumé

La transformation de données en connaissances constitue une tâche cruciale au cœur de nombreuses activités professionnelles. Deux principales stratégies peuvent être envisagées pour effectuer cette transformation : l'interaction par requêtage avec un système de gestion de bases de données ou l'application de méthodes souvent automatiques de fouille de données. Ces deux approches ont jusqu'alors été étudiées de manière indépendante par deux communautés scientifiques distinctes : celle des bases de données et celle de la fouille de données.

Les travaux décrits dans ce document, dont un objectif est d'effectuer une synthèse constructive des résultats obtenus au cours de mes huit années de recherche à l'IRISA, s'inscrivent principalement dans le cadre de l'interrogation de bases de données. Cependant, de par l'importance grandissante prise par les données non structurées, mes dernières contributions établissent une intersection entre l'acquisition automatique non supervisée de connaissances et l'interrogation de données. Le fil conducteur de ce document est l'enrichissement des méthodes d'accès aux données. L'accès aux données y est vu comme un processus en trois étapes, 1) l'expression d'un besoin d'information, 2) la récupération efficace des données satisfaisant le besoin d'information exprimé et 3) la restitution des résultats à l'utilisateur. Le trait singulier de la chaîne de traitement de données décrite dans ce document provient de la place prépondérante accordée à l'utilisateur à chaque étape du processus de transformation des données en connaissances.

La première partie de ce document est consacrée à l'enrichissement des méthodes d'accès aux données. Mes contributions sur l'enrichissement de l'étape d'expression des besoins d'information s'articulent autour de deux axes. Le premier consiste à rendre flexibles les interfaces d'interrogation et à améliorer leur expressivité en permettant à l'utilisateur d'accéder aux données à travers l'utilisation d'un vocabulaire personnel composé de descripteurs linguistiques. Le second axe consiste à assister l'utilisateur, avec des stratégies coopératives ou des interfaces d'interrogation intuitives, lors de la traduction de son besoin d'information en requête.

Les systèmes commerciaux de gestion de données n'étant pas initialement pourvus de fonctionnalités d'interrogation flexible à l'aide de descripteurs linguistiques, la seconde partie du document décrit mes contributions sur l'évaluation de conditions de sélection de données exprimées à l'aide de descripteurs linguistiques subjectifs. Ces travaux m'ont permis de montrer qu'il était possible de trouver un compromis intéressant entre flexibilité et efficacité lors de l'interrogation de données.

Un système intelligent d'accès aux données se doit d'accompagner l'utilisateur lors de l'analyse des résultats de sa requête. Les stratégies de réponse coopérative visent à aider l'utilisateur à comprendre un ensemble de résultats et à l'enrichir avec des données ou connaissances complémentaires. La troisième partie de ce document détaille plusieurs stratégies coopératives permettant à l'utilisateur de transformer plus rapidement les résultats de ses requêtes en connaissances.

Le cadre théorique qui unifie les maillons de la chaîne de traitement de données présentée dans ce document est celui du *soft computing*. Ce document a également pour objectif de montrer que les théories et techniques de *soft computing* apportent des solutions pragmatiques et novatrices à un enjeu actuel crucial, celui de la valorisation des données. Le bilan, dressé sous forme de perspectives de recherche à la fin de ce document, souligne le rôle majeur que peut jouer la communauté scientifique du *soft computing* en promouvant l'idée de représenter, calculer et raisonner sur des données avec des mots.

Mots clés : Interrogation flexible de BD, approche coopérative, exécution de requêtes floues, résumé linguistique de données, exploration de données, extraction de connaissances.

Abstract

The translation of data into knowledge is a crucial task at the heart of many professional activities. Two main strategies may be envisaged to perform this translation : by querying a database management system or by using data mining techniques. These two approaches have been so far studied independently by two distinct communities, namely the database community and the data mining one.

The works described in this document, whose aim is to synthetize the research results obtained during the last eight years passed in the IRISA laboratory, mainly belong to the database area. However, considering the growing importance of unstructured data, my last contributions are at the intersection of data mining and databases. The common thread in this document is the enrichment of the methods used to access data. Data access is considered as a three steps process : 1) the expression of an information need, 2) the efficient retrieval of data satisfying the considered information need, and 3) the restitution of the query results to the user. The singular aspect of the data processing chain described in this document relies on the leading role given to the user at each step of the process defined to translate data into knowledge.

The first part of the document is dedicated to the enrichment of some methods used to access data. My contributions on that point are twofold. The first one aims at making querying interfaces more flexible and at increasing their expressivity by letting users access data using their own vocabulary composed of linguistic terms. The second approach consists in helping users, with cooperative strategies or intuitive query interfaces, translate their information needs into queries.

As commercial database systems do not provide flexible querying functionalities, the second part of the document describes my contributions on the evaluation of selection statements involving conditions based on the satisfaction of subjective linguistic terms. Through these last works, I have shown that a compromise may be found between flexibility and efficiency when querying data.

An intelligent data management system should also assist users during the analysis of the results of their queries. Cooperative answering strategies aim at helping users understand the content of a result set and also aim at enriching it with indirect answers and complementary knowledge. The third part of the document details several cooperative answering strategies that ease the translation of query results into knowledge.

The theoretical framework that links the different parts of the data processing chain presented in this document is soft computing. In this sense, an underlying objective of this document is also to show that the theories and techniques of soft computing bring pragmatic and innovative solutions to answer the crucial issue of data management. A positive conclusion and perspectives for future research directions are given at the end of this document about the role the soft computing community can play by promoting the idea of representing, computing and reasoning about data with words.

Keywords : DB flexible querying, cooperative approaches, fuzzy query processing, linguistic summary of data, data exploration, knowledge extraction.

Donnée *[nom féminin]* [1]
— Ce qui est connu et admis, et qui sert de base, à un raisonnement, à un examen ou à une recherche.
— Ensemble des indications enregistrées en machine pour permettre l'analyse et/ou la recherche automatique des informations.
Connaissance *[nom féminin]*
— Action ou acte de se faire une représentation, de s'informer ou d'être informé de l'existence de quelque chose.
— Action ou fait d'apprendre quelque chose par l'étude et/ou la pratique ; résultat de cette action ou de ce fait.

Introduction : des données aux connaissances

Initialement conçus pour stocker de grandes quantités de données et leur appliquer des traitements complexes, les ordinateurs n'ont jamais aussi bien rempli leur rôle que ces dernières années. À des fins d'observation, d'analyse ou d'optimisation, nous assistons à la numérisation de quasiment tous les phénomènes et objets mesurables. Ce constat a conduit à considérer le début du XXI^{ème} siècle comme l'ère des données, des données ouvertes, des données distribuées, des grandes masses de données. Face à cet afflux massif de données numériques, des solutions techniques et algorithmiques performantes ont tout d'abord été développées par les acteurs industriels et académiques œuvrant dans le domaine de la gestion des données pour garantir leur stockage, leur transmission et leur traitement. C'est notamment grâce à ces prouesses technologiques que chaque minute plus de 70 heures de vidéo peuvent être transférées sur la plateforme Youtube, près de 300 000 messages peuvent être publiés sur Twitter et plus de 200 000 photographies sont partagées sur Instagram.

Au-delà de ces services ayant principalement un caractère ludique, le processus de numérisation, de centralisation et de publication des données est également à l'origine de mutations parfois profondes des activités quotidiennes de nombreux professionnels, et ceci dans des contextes applicatifs très variés. De la finance à la médecine en passant par la politique et la gestion de collectivités territoriales, de nombreuses décisions métier se basent désormais sur l'analyse de données décrivant une population ou un phénomène ciblé. À titre d'exemple, l'ajustement des grilles tarifaires d'un assureur ne repose plus sur son expertise ou sur l'application de modèles économiques, mais s'appuie désormais sur des indicateurs statistiques extraits de données croisant des descriptions de sinistres avec des informations démographiques. Afin de prendre les bonnes décisions sur la base d'inférences inductives ou abductives, les experts métier ont besoin d'outils pragmatiques mais génériques, efficaces mais expressifs, intuitifs mais coopératifs, propriétés généralement mutuellement antagonistes, leurs permettant de transformer des données en connaissances. Ainsi, selon plusieurs observateurs économiques et technologiques [2], le principal enjeu auquel les scientifiques des données (industriels et académiques) doivent répondre concerne le développement d'outils permettant aux experts métier d'être indépendants et efficaces lors de la transformation de leurs données en connaissances. Compte tenu du potentiel économique de ce secteur, cette problématique est pour le moment principalement abordée par les grands acteurs industriels de l'informatique décisionnelle (IBM Watson Analytics, Microsoft Power BI, Tableau Desktop, etc.). Il est alors légitime de s'interroger sur la place que peuvent prendre des acteurs scientifiques universitaires dans ce domaine, et ce document apporte un angle de réponse à cette question.

L'objectif de ce document est double. D'un point de vue personnel, ce travail me permet d'effectuer une synthèse critique sur les travaux que j'ai réalisés au sein de l'IRISA lors des 8 dernières années. Plus globalement, ce document montre que le soft computing, et plus précisément la théorie des sous-ensembles flous, offre un cadre théorique idéal pour le développement d'une solution novatrice à une problématique actuelle cruciale : **aider les experts métier à transformer des données en connaissances**. L'originalité de mes contributions dans ce contexte d'extraction assistée de connaissances à partir de données repose sur l'application d'une idée fantastique datant de

1. Source : TLFi : Trésor de la langue Française informatisé, `http://www.atilf.fr/tlfi`, ATILF - CNRS & Université de Lorraine.
2. `https://www.quantzig.com/blog/top-10-data-analytics-trends-watch-2017` et `https://bi-survey.com/self-service-bi`

plus d'un demi-siècle [144] : celle de **calculer avec des mots (Computing with Words)** [147]. Ce paradigme est utilisé pour créer une interface personnalisée entre l'espace, généralement numérique et catégoriel, de définition des données et l'espace, davantage linguistique, de la cognition et du raisonnement humains. De par sa capacité à formaliser mathématiquement des notions langagières subjectives que nous employons pour décrire des valeurs (e.g. un « montant *très élevé* »), des phénomènes (e.g. une « corrélation *faible* ») ou des tendances (e.g. la « *plupart* des transactions *risquées* sont *récentes* »), la théorie des sous-ensembles flous est utilisée dans ce contexte d'extraction de connaissances à partir des données pour :

- **formaliser un vocabulaire subjectif** composé de descripteurs linguistiques de données brutes,
- **personnaliser et simplifier l'expression des recherches d'informations** en permettant à l'expert de les formuler à l'aide de son propre vocabulaire,
- **améliorer l'expressivité des langages formels d'interrogation** à travers la définition d'opérateurs d'agrégation aux sémantiques riches et variées,
- fournir des **explications les plus interprétables possibles** sur les différentes étapes et les résultats intermédiaires ou finaux du processus d'extraction de connaissances.

Comme l'illustre la figure 1, les contributions scientifiques décrites dans ce document forment, de par leur complémentarité, un processus complet d'extraction de connaissances à partir des données. La singularité de cette vision du traitement des données provient principalement de l'importance accordée à l'utilisateur en charge de la tâche d'analyse, lors de la phase d'**interrogation** (Phase 1 sur la figure 1) et lors de la **restitution** des connaissances extraites (Phase 3 sur la figure 1). La personnalisation et l'enrichissement des méthodes d'interrogation conduisent à la nécessité de développer des méthodes spécifiques et efficaces d'**exécution** des requêtes (Phase 2 sur la Fig. 1). En fournissant des fonctionnalités d'interrogation flexible de données et en cherchant à enrichir ces interactions et leurs résultats par des connaissances indirectes, les travaux résumés dans ce document se situent à l'intersection de domaines de recherche complémentaires, à savoir les bases de données, la gestion de connaissances et le soft computing.

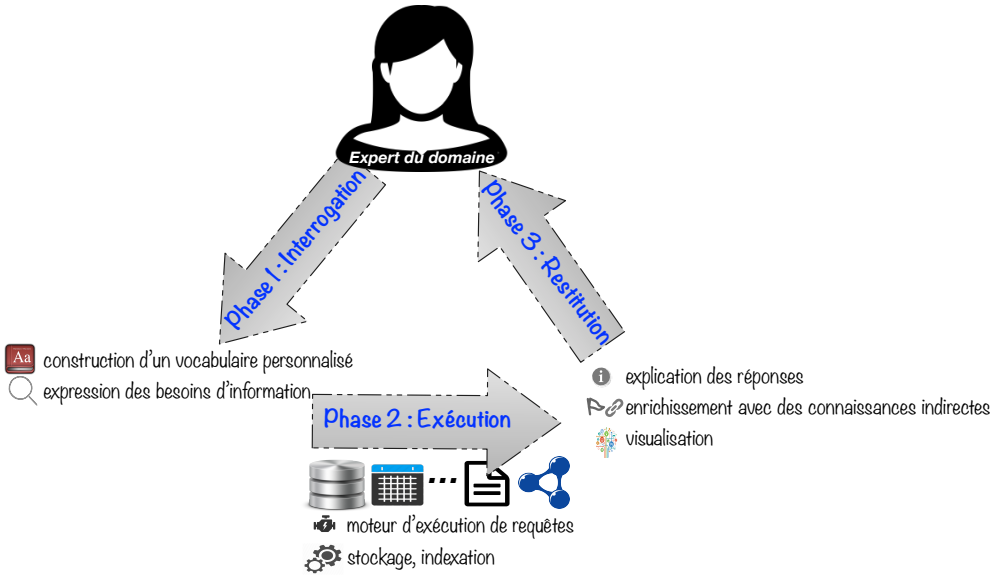

FIGURE 1 – Les trois phases du processus d'exploitation des données

La structure du présent document reprend les trois phases du processus d'extraction de connaissances à partir des données. La partie I présente dans un premier temps mes contributions autour de la formulation d'un vocabulaire expert, puis des méthodes flexibles et coopératives d'interrogation simplifiant et enrichissant l'expression d'un besoin d'information. La seconde partie (Part. II) décrit les techniques mises en œuvre pour exécuter efficacement ces requêtes personnalisées, notamment avec des implémentations et des index dédiés. Pour aller au-delà de la restitution des données

satisfaisant le besoin d'information exprimé, la troisième partie (Part. III) introduit des stratégies de réponse coopérative dont l'objectif est de simplifier et d'accélérer la transformation des résultats en connaissances exploitables. Cette coopération intervient à différents niveaux, de l'explication des résultats directs à leur enrichissement avec des réponses et des connaissances indirectes.

La synthèse de mes contributions passées, présentée sous-forme d'une chaîne d'extraction de connaissances personnalisées à partir de données, me permettra ensuite d'établir dans la partie 12 des perspectives de travail à moyen et long terme. Je tenterai également dans la dernière partie de ce document de dresser un bilan, subjectif, des apports et de la place du soft computing dans le contexte applicatif de l'extraction de connaissances à partir de données brutes.

Première partie

Personnalisation des méthodes d'accès aux données

Personnalisation *[nom féminin]*[3]
— Action de disposer et d'arranger quelque chose de façon à répondre aux besoins et aux goûts d'une personne.
Préférence *[nom féminin]*
— Jugement d'estime ou sentiment de prédilection par lequel on donne à une personne ou à une chose la prééminence sur une autre.

Les outils et principales technologies d'accès aux sources de données, structurées ou non, ont principalement été conçus sous l'angle de la généricité. Cependant, une manière d'accélérer et de simplifier l'exploitation des données est de personnaliser les méthodes d'interrogation. Le gain, en termes de pertinence et d'interprétabilité des résultats générés, a été démontré dans de nombreux contextes applicatifs, du résumé de données [116, 7] à l'interrogation flexible [16, 48, 135] en passant par la navigation par facettes [126].

Les travaux présentés dans cette section abordent la personnalisation de systèmes d'information à travers la prise en compte de descripteurs linguistiques dont la définition subjective permet de personnaliser et de discriminer les descriptions des données. Des données définissant un prix, une marque de voiture ou une température, peuvent en effet être décrites par des descripteurs linguistiques comme *« acceptable »*, *« fiable »* ou *« élevée »* traduisant l'interprétation subjective et souvent imprécise que l'utilisateur en fait. L'ensemble des descripteurs forme un vocabulaire personnel pouvant être utilisé pour personnaliser les processus de manipulation de données. Un vocabulaire utilisateur constitué d'un ensemble de descripteurs linguistiques est à la fois subjectif, car il dépend du sens que l'utilisateur leur accorde, mais il est également lié à un contexte applicatif, c'est-à-dire aux données concernées. Le descripteur *température « élevée »* n'est généralement pas associé à la même définition s'il s'agit de décrire la température ambiante d'une pièce ou s'il s'agit de décrire la température remontée par un capteur branché sur un moteur thermique.

Dans un contexte d'interrogation de données, ces descripteurs linguistiques peuvent être utilisés pour former des prédicats. De par la subjectivité et l'imprécision de leur définition, les prédicats construits à partir de descripteurs linguistiques sont généralement perçus comme des préférences qui expriment des différentiations personnelles des objets manipulés selon les valeurs qui leurs sont associées : e.g., parmi les couleurs de voiture, je préfère les couleurs claires aux sombres, ou alors dans cette gamme de prix je souhaite maximiser la puissance du moteur, etc. De nombreux travaux sont focalisés sur la formalisation de ces préférences [120], puis sur leur prise en compte dans les langages et les interfaces d'interrogation [16, 95].

Cependant, définir des descripteurs linguistiques dans un contexte applicatif particulier n'est pas une tâche aisée pour l'utilisateur final. Il est donc indispensable de doter les systèmes d'interrogation flexible de fonctionnalités coopératives visant à assister l'utilisateur dans la définition de descripteurs pouvant ensuite être utilisés efficacement pour manipuler les données. Deux principales approches peuvent être envisagées pour assister un utilisateur dans la définition de ses préférences : l'élicitation d'un modèle à partir des données ou d'exemples [58, 62], et le développement d'interfaces intuitives et interactives de construction de préférences.

Après avoir fait quelques rappels succincts sur la formalisation à l'aide de sous-ensembles flous de descripteurs linguistiques dans la section 1, je présenterai mes contributions autour de l'aide à la construction d'un vocabulaire utilisateur en adéquation avec les données manipulées (Sec. 2). En complément de ces travaux sur l'élicitation d'un vocabulaire à partir des données, la section 3 détaillera mes contributions portant sur l'enrichissement des interfaces d'interrogation de données par la prise en compte de descripteurs linguistiques subjectifs.

3. Source : TLFi : Trésor de la langue Française informatisé, `http://www.atilf.fr/tlfi`, ATILF - CNRS & Université de Lorraine.

1 Modélisation de descripteurs linguistiques subjectifs

Rendre les méthodes d'accès aux données plus flexibles, c'est tout d'abord permettre à l'utilisateur d'exprimer ses connaissances sur les données manipulées. Cette section n'aborde évidemment qu'une infime partie de ce domaine de recherche très vaste qui est celui de la représentation des connaissances [26], celle concernant la définition de descripteurs subjectifs de données. De par leur nature subjective et imprécise, ces descripteurs sont généralement considérés comme des préférences pouvant ensuite être utilisées pour décrire et discriminer un ensemble d'objets selon les données qui les caractérisent. On distingue généralement deux familles de modèles de préférences : les approches qualitatives par opposition à celles dites quantitatives.

La vision qualitative des préférences consiste à définir une relation d'ordre sur les valeurs d'un univers, soit de manière totalement indépendante des autres dimensions [75, 30] (e.g. $break \succ SUV \succ berline \succ sport \succ 4X4$), soit de manière conditionnelle [24] (e.g. si $type = sport$ alors $rouge \succ blanc \succ noir$).

La seconde vision adoptée pour modéliser les préférences d'un utilisateur est quantitative car elle repose sur le calcul et l'affectation d'un score pour chaque objet comparé. Ce score quantifie dans quelle mesure l'objet satisfait les préférences exprimées et sert donc de discriminant pour identifier les réponses les plus intéressantes, i.e. celles ayant les scores les plus élevés. Le modèle quantitatif le plus étudié jusqu'alors consiste à représenter les préférences sous forme d'une fonction d'utilité dont le score généré est à maximiser [54, 1, 25]. Principalement exploitée dans le but d'automatiser la prise de décision à partir de connaissances métier [55], ces approches, ou plus précisément les fonctions d'utilité et leurs paramètres, souffrent d'un manque d'interprétabilité les rendant difficilement exploitables dans un contexte d'interrogation de données par des experts souvent non informaticiens ni mathématiciens. Les travaux présentés dans cette section autour de l'interrogation de données par requêtes à préférences exploitent un modèle quantitatif de représentation des préférences basé sur la théorie des sous-ensembles flous [144]. Ce cadre théorique de représentation d'ensembles de valeurs aux frontières vagues permet de modéliser fidèlement le caractère subjectif et imprécis des préférences utilisateur [97].

1.1 Modélisation de descripteurs linguistiques à l'aide de sous-ensembles flous

Dans un contexte d'interrogation flexible de données, nous entendons par connaissance experte la formalisation d'un descripteur de données brutes, numériques ou catégorielles, dont la signification est subjective et contextuelle. La table 1 illustre une extension possible de relation décrivant des employés et leur département d'affectation. Des descripteurs subjectifs pour ces données seraient par exemple *« jeune »* pour l'attribut *âge*, *« élevée »* pour l'attribut *ancienneté*, *« à fortes responsabilités »* pour l'attribut *posteOccupé*, etc.

TABLE 1 – Exemple de données relationnelles

TABLE 2 – Relation *emp* décrivant des employées

idE	nom	ancienneté	dep	posteOccupé
e1	Sophie	8	d3	chef d'équipe
e2	Lucas	10	d3	chercheur
e3	Georges	2	d3	doctorant
e4	Théo	1	d2	programmeur
e5	Youri	3	d2	chef de projet
e6	Frank	5	d1	resp. des ventes
e7	Yoann	2	d1	vendeur
e8	Marie	19	d4	PDG
e9	Loran	9	d3	chercheur

TABLE 3 – Relation *dep* des départements de l'entreprise

idD	désignation	site	budget en K€
d1	commercial	Paris	200
d2	développement	Lannion	3750
d3	recherche	Lannion	2128
d4	direction	Paris	360

La théorie des Sous-Ensembles Flous (SEF) constitue un formalisme mathématique idéal pour représenter le caractère subjectif et imprécis de ces descripteurs de données. L'utilisation de descripteurs linguistiques pour manipuler des données brutes constitue sans doute l'un des premiers objectifs applicatifs de cette théorie, celui de **calculer avec des mots** [147].

Soit A un attribut numérique ou catégoriel de domaine de définition X et F un SEF de X. F est défini par deux composantes :

— sa fonction d'appartenance $\mu_F : X \to [0, 1]$,

— et un label linguistique l^F.

La fonction d'appartenance μ_F associée à un SEF F permet à la fois de définir une relation d'indistinguabilité entre certaines valeurs (le noyau $\{x \in X | \mu_F(x) = 1\}$), comme le fait un ensemble classique, mais également d'établir une zone d'imprécision pour discriminer les données ne satisfaisant que partiellement le descripteur considéré. Afin de clairement distinguer ces deux zones, nous utiliserons des représentations trapézoïdales des SEFs.

Le label linguistique l^F associé à F, généralement de nature adjectivale, forme une interface entre l'espace, principalement numérique et catégoriel, de définition des données brutes et l'espace linguistique de la cognition humaine. La figure 2 et la table 4 illustrent graphiquement une définition possible des descripteurs *ancienneté « élevée »* et *posteOccupé « à fortes responsabilités »*.

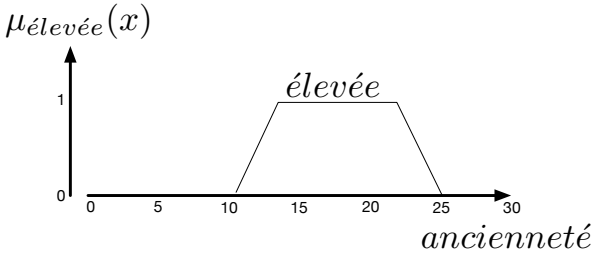

FIGURE 2 – Définition subjective et contextuelle du descripteur *ancienneté « élevée »*

TABLE 4 – Définition subjective et contextuelle du descripteur *posteOccupé « à fortes responsabilités »*

posteOccupé	$\mu_{àfortesresponsabilités}$	*posteOccupé*	$\mu_{àfortesresponsabilités}$
chef d'équipe	0,8	programmeur	0,3
chercheur	0,7	res. des ventes	0,5
doctorant	0,4	vendeur	0,3
chef de projet	0,7	PDG	1

En utilisant un SEF pour former une condition de recherche, subjective et imprécise, le lien entre SEF et préférence est alors établi dans la mesure où le degré d'appartenance au SEF s'interprète comme un degré de satisfaction à maximiser et permettant de discriminer les objets entre eux.

1.2 Modélisation d'un vocabulaire utilisateur

Soit un ensemble d'objets $\mathcal{D} : \{d_1, d_2, \ldots, d_m\}$, chaque objet étant décrit par un ensemble d'attributs $\mathcal{A} : \{A_1, A_2, \ldots, A_n\}$. Une manière de modéliser des connaissances sur $\mathcal{D}$ consiste à définir des partitions sur les domaines de définition des attributs. Pour un attribut donné, la définition d'une partition discrétisant son domaine de définition permet de passer d'une définition initiale des données par variables numériques et catégorielles à une description linguistique à l'aide de variables linguistiques et donc de termes [145].

Definition *Soit $A \in \mathcal{A}$ un attribut sur lequel les objets de $\mathcal{D}$ sont décrits numériquement ou symboliquement. Une* **partition** *discrétisant le domaine de définition X de A est notée $\mathcal{P}_A$ et forme une variable linguistique définie par un triplet $\langle A, \{\mu_1^A, \mu_2^A, \ldots, \mu_q^A\}, \{l_1^A, l_2^A, \ldots, l_q^A\}\rangle$, où les μ_i^As correspondent aux fonctions d'appartenances des modalités présentes dans la partition et les l_i^A sont les labels linguistiques associés. Pour des raisons d'interprétabilité et d'utilisabilité, il est imposé que $\mathcal{P}_A$ constitue une partition forte dite de Ruspini, c'est-à-dire que toute valeur de X doit être complètement réécrite et au maximum par deux modalités adjacentes :*

— $\forall x \in X, \sum_{i=1}^{q} \mu_i^A(x) = 1$,

— $|\{\mu_i^A, \mu_i^A(x) > 0\}| \leq 2$,
— *et* $\forall x \in X, \forall i = 2..q-1$ *si* $0 < \mu_i^A(x)$ *alors* $\mu_i^A(x) = 1 - max(\mu_{i-1}^A(x), \mu_{i+1}^A(x))$.

Dans le cas d'un domaine non ordonné, comme la plupart des attributs catégoriels, la troisième contrainte définie ci-dessus n'a évidemment pas lieu d'être.

La figure 3 illustre une discrétisation possible du domaine de définition de l'attribut *ancienneté* à l'aide d'une partition floue.

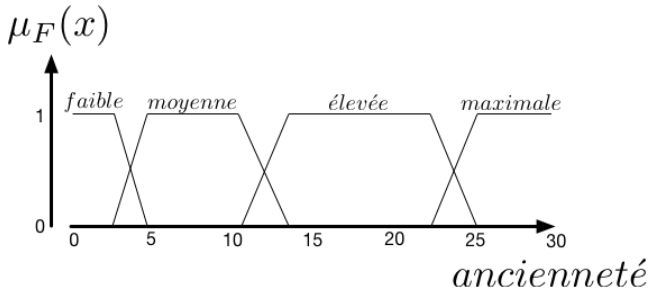

FIGURE 3 – Définition d'une variable linguistique pour l'attribut *ancienneté*

Definition *Un* **vocabulaire utilisateur**, *noté* $\mathcal{V}$, *pour* $\mathcal{D}$ *consiste en une discrétisation à l'aide de partitions floues (associées à des variables linguistiques) d'un sous-ensemble d'attributs* $\mathcal{A}'$ *de* $\mathcal{A}$ *jugés intéressants par l'utilisateur :*

$$\mathcal{V} : \{P_A, A \in \mathcal{A}' \subseteq \mathcal{A}\}.$$

1.3 Espaces de représentation des données

Soit $\mathcal{D} : \{d_1, d_2, \ldots, d_m\}$ un ensemble d'objets, chacun décrit sur un ensemble d'attributs $\mathcal{A} : \{A_1, A_2, \ldots, A_n\}$ de domaines de définition respectifs $\{X_1, X_2, \ldots, X_n\}$. Un objet d est initialement défini dans un espace numérique/catégoriel : $d \in X_1 \times X_2 \times \ldots \times X_n$.

Un objet peut être réécrit sur la base d'un vocabulaire utilisateur en projetant chacune de ses valeurs sur les partitions qui composent ce vocabulaire.

Definition *On nomme* **vecteur de réécriture linguistique d'un objet** *d selon un vocabulaire utilisateur* $\mathcal{V}$ *la concaténation des degrés de satisfaction des valeurs de d vis-à-vis des modalités qui composent les partitions de* $\mathcal{V}$. *Un tel vecteur, noté* $RV_d^{\mathcal{V}}$, *est représenté de la façon suivante :*

$$\langle \mu_1^{A_1}(d), \mu_2^{A_1}(d), \ldots, \mu_{q_1}^{A_1}(d), \ldots, \mu_1^{A_n}(d), \mu_2^{A_n}(d), \ldots, \mu_{q_n}^{A_n}(d) \rangle,$$

où q_i *désigne le nombre de modalités qui composent la partition* P_{A_i}.

En raison des contraintes structurelles imposées sur les partitions de $\mathcal{V}$ (Sec.1.2), le vecteur de réécriture $RV_d^{\mathcal{V}}$ d'un objet est très creux car il contient au maximum $2 \times n$ composantes non nulles.

Les réécritures individuelles des objets de $\mathcal{D}$ peuvent être agrégées pour former un vecteur de réécriture global de $\mathcal{D}$. Ce vecteur, noté $RV_{\mathcal{D}}^{\mathcal{V}}$, renseigne sur la distribution des termes linguistiques issus du vocabulaire $\mathcal{V}$ par rapport aux objets du jeu de données $\mathcal{D}$.

Definition *On nomme* **vecteur de réécriture linguistique d'un ensemble de données** $\mathcal{D}$ *selon un vocabulaire utilisateur* $\mathcal{V}$ *l'agrégation des vecteurs de réécriture* $RV_d^{\mathcal{V}}, d \in \mathcal{D}$. *Un tel vecteur, noté* $RV_d^{\mathcal{V}}$, *est représenté de la façon suivante :*

$$\langle \rho_{v_1^{A_1}}(\mathcal{D}), \ldots, \rho_{v_{q_1}^{A_1}}(\mathcal{D}), \ldots, \rho_{v_1^{A_n}}(\mathcal{D}), \ldots, \rho_{v_{q_n}^{A_n}}(\mathcal{D}) \rangle,$$

où $v_j^{A_i}$ *désigne le* j^{eme} *SEF de la partition définie sur* A_i *et* $\rho_{v_j^{A_i}}(\mathcal{D}) = \frac{\sum_{d \in \mathcal{D}} \mu_j^{A_i}(d)}{|\mathcal{D}|}$.

Ce vecteur fournit une représentation synthétique dans l'espace linguistique, personnel à l'utilisateur, des données initialement décrites par des valeurs numériques et catégorielles.

2 Aide à la construction d'un vocabulaire

Subjectif et lié à un contexte applicatif particulier, un vocabulaire utilisateur doit être en adéquation vis-à-vis des données manipulées pour garantir une personnalisation cohérente des processus d'analyse. Il serait en effet peu informatif de définir un descripteur linguistique qui différencie fortement deux données très proches sans distinguer des données nettement plus éloignées. Utilisé pour former une condition de sélection dans une requête flexible, un tel descripteur conduirait à des situations problématiques telles que l'absence ou symétriquement la surabondance de résultats.

En collaboration avec Marie-Jeanne Lesot du Laboratoire Informatique Paris 6 (LIP6), nous avons travaillé sur cette problématique identifiée [116, 135], mais jusqu'alors jamais étudiée, concernant l'adéquation entre un vocabulaire utilisateur et un jeu de données, ou plus précisément leur structuration intrinsèque sous forme de classes distinguables. Nous avons tout d'abord proposé une mesure d'adéquation entre la structure intrinsèque des données et celle reflétée par un vocabulaire utilisateur (Sec. 2.2). Puis dans un second temps, nous avons travaillé sur des méthodes d'aide à la construction de descripteurs linguistiques garantissant un compromis entre interprétabilité et adéquation vis-à-vis des données (Sec. 2.3). Contrairement aux méthodes complètement automatiques de construction de partitions à partir des données [87, 62] ou de sélection des attributs à considérer [73], nous privilégions une approche coopérative pour l'élicitation de descripteurs afin de laisser l'expert exprimer sa vision des données.

2.1 Structure intrinsèque des données

Du fait de corrélations entre attributs, les données réelles qui décrivent des phénomènes non aléatoires peuvent être généralement structurées en classes, notamment à l'aide de méthodes automatiques d'apprentissage non-supervisé [70]. Le processus d'apprentissage vise à distinguer les sous-ensembles de données partageant des propriétés similaires dans l'espace initial numérique/catégoriel de description des données.

La construction de classes $\{C_1, C_2, \ldots, C_k\}$ repose sur l'utilisation d'une mesure de ressemblance entre deux objets d_i et d_j, notée ici $R(d_i, d_j)$, et d'une mesure de dissemblance notée $D(d_i, d_j)$. Le choix de ces mesures est évidemment crucial et de nombreux travaux, notamment au sein de la communauté du *soft computing*, portent sur l'étude de leurs propriétés [23, 64, 82]. La stratégie d'identification de classes mise en œuvre dans les algorithmes que nous avons utilisés [5, 78, 81] repose sur la maximisation de deux fonctions objectifs, l'une quantifiant la similarité des objets appartenant à une même classe et l'autre la dissimilarité entre les objets de classes différentes.

À partir des classes obtenues, une connaissance importante sur les objets, notamment exploitée pour l'élicitation de descripteurs (Sec. 2.3), concerne leur degré relatif de représentativité vis-à-vis de la structuration en classes des données. Cette représentativité quantifie le caractère prototypique d'un objet par rapport à sa classe d'appartenance [83]. Soit un objet d affecté à une classe C, une définition usuelle de la fonction permettant de quantifier le degré de typicité de d dans C, notée ici $\Gamma(d, C)$ est la suivante :

$$\Gamma(d, C) = \Phi[avg(R(d, d'_{d' \in C})), avg(D(d, d''_{d'' \notin C}))],$$

où Φ est un opérateur d'agrégation [118]. En plus de construire des sous-ensembles, potentiellement flous, de $\mathcal{D}$, les méthodes de clustering utilisées identifient pour chaque classe son représentant le plus prototypique. Cet élément, noté M_i, désigne le médoïde (ou centre) de la classe C_i et correspond à l'objet de C_i ayant le degré de typicité maximal : $M_i = argmax_{d \in \mathcal{D}} \Gamma(d, C_i)$.

2.2 Quantifier l'adéquation entre les données et le vocabulaire utilisateur

Comme indiqué dans la section 1.1, la formalisation d'un vocabulaire utilisateur par discrétisation des domaines de définition des attributs induit des relations d'indistinguabilité entre les objets décrits. Les objets appartenant à la même α-coupe d'un descripteur F (i.e. $\{d \in \mathcal{D}, \mu_F(d) \geq \alpha\}$)

sont indistinguables du point de vue de leur description linguistique. Un vocabulaire utilisateur $\mathcal{V}$ est dit en adéquation avec les données qu'il décrit si la relation d'indistinguabilité qu'il introduit est compatible avec la définition numérique/catégorielle de ces données. E.g, sur la figure 4, la définition des SEFs F_a et F_b rend les valeurs x, y et z indistinguables entre elles, mais linguistiquement complètement distinctes de w. Il apparaitrait plus cohérent d'associer les valeurs z et w a une réécriture linguistique proche mais distinguable de celle de x et y, surtout si x et y appartiennent à une même classe différente de celle de z et w.

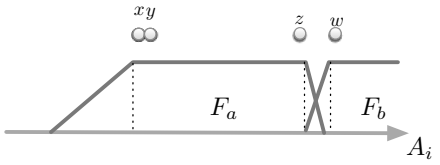

FIGURE 4 – Illustration d'un inadéquation possible entre données et vocabulaire

Dans le papier [84], deux critères sont proposés pour quantifier cette adéquation. Le premier est basé sur la comparaison entre la structure de classes obtenue en applicant un même algorithme de classification (algorithme *lfcmed-select* [81]) dans l'espace initial de définition des données et dans l'espace de description linguistique. Le second critère consiste à quantifier la capacité du vocabulaire à décrire les classes construites dans l'espace initial de définition des données.

Comparaison des structures induites

La première stratégie proposée pour quantifier l'adéquation entre la structure intrinsèque en classes des données et le vocabulaire utilisateur est d'appliquer un même algorithme de classification non supervisé dans chacun des deux espaces. Nous obtenons alors deux structurations en classes, notées respectivement $C(\mathcal{D})$ et $C(\mathcal{RD})$, qu'il s'agit de comparer. L'application d'un algorithme de partitionnement repose sur une mesure de distances entre les objets comparés. Dans l'espace linguistique, nous avons utilisé la métrique introduite dans [63] pour quantifier la proximité entre deux vecteurs de réécriture. De nombreux critères ont été proposés pour comparer des structures entre elles [2]. Nous avons utilisé dans ce travail l'indice de Rand ajusté, noté $ira(C(\mathcal{D}), C(\mathcal{RD}))$ quantifiant la proportion d'affectations concordantes au sein des deux structures. L'adéquation entre les structures en classes induites dans chacun des deux espaces est alors définie de la façon suivante :

$$ad(C(\mathcal{D}), C(\mathcal{RD})) = ira(C(\mathcal{D}), C(\mathcal{RD})).$$

Cette stratégie a l'avantage de placer notre problématique dans un cadre classique étudié par la communauté du data mining, la comparaison de structures. Cependant, elle repose sur la double application d'un algorithme de structuration, ce qui peut s'avérer coûteux notamment dans le cas où l'on souhaiterait quantifier dynamiquement cette adéquation lors d'un processus de construction de vocabulaire (Sec. 2.4).

Projection de classes sur le vocabulaire

La seconde méthode envisagée pour quantifier l'adéquation entre un vocabulaire et une structure en classes de données s'affranchit de la double application d'un algorithme de structuration. La structure obtenue dans l'espace de définition des données $C(\mathcal{D})$ est alors évaluée vis-à-vis du vocabulaire selon deux critères assez classiques : la compacité et la séparabilité. Le premier quantifie à quel point les objets d'une même classe sont proches et le second à quel point les objets de classes différentes sont distincts. Ces deux notions sont dans notre cas définies dans l'espace linguistique où nous vérifions si les centres de clusters indistinguables (resp. séparés) dans l'espace de définition le sont également une fois réécrits à l'aide du vocabulaire sur une dimension donnée. Pour combiner ces deux notions, nous utilisons l'indice de Xie et Beni [140] qui correspond au quotient de la compacité et de la séparabilité. La mesure d'adéquation utilisant cette stratégie est notée $q(\mathcal{V}, \mathcal{D})$.

Résultats expérimentaux

Ces deux approches, $ad(C(\mathcal{D}), C(\mathcal{RD}))$ et $q(\mathcal{V}, \mathcal{D})$, visant à quantifier l'adéquation entre des données et un vocabulaire ont été implémentées et testées sur des données artificielles représentant des classes elliptiques bien séparées (Fig. 5). Sur plusieurs vocabulaires prédéfinis, les approches $ad(C(\mathcal{D}), C(\mathcal{RD}))$ et $q(\mathcal{V}, \mathcal{D})$ ont été utilisées pour quantifier leur adéquation respective.

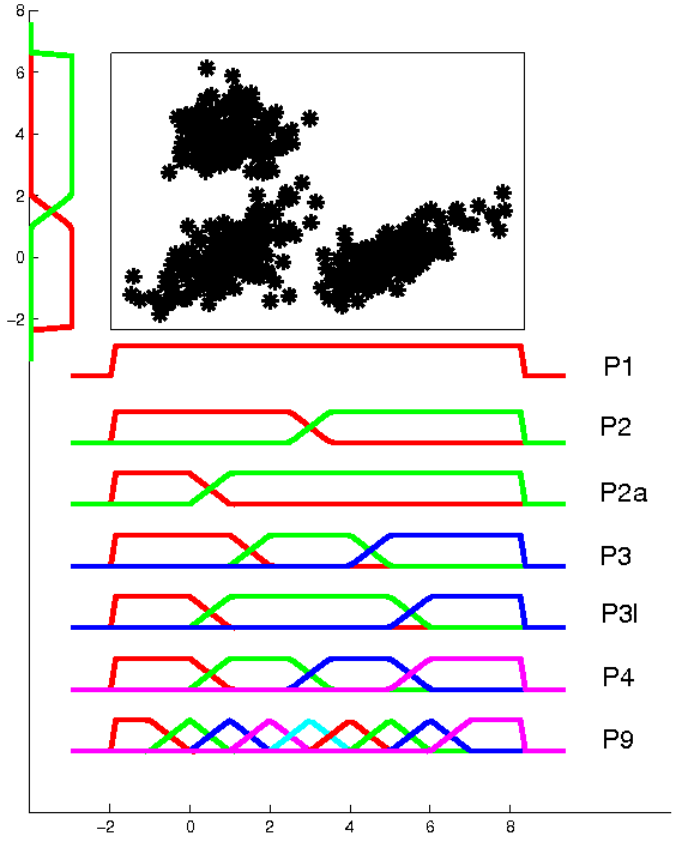

FIGURE 5 – Quantifier l'adéquation entre des données artificielles et des vocabulaires possibles

Ces expérimentations ont montré la capacité des deux approches à distinguer les vocabulaires exploitables (P_2, P_4 et P_9) des inadaptés (les autres). Les deux approches ne génèrent pas les mêmes conclusions. L'approche $ad(C(\mathcal{D}), C(\mathcal{RD}))$ est sensible aux éléments de vocabulaire décrivant les transitions entre classes, le critère $q(\mathcal{V}, \mathcal{D})$ l'étant moins en considérant le vocabulaire P_3 comme acceptable. À l'inverse, cette dernière méthode est sensible à la granularité de la partition et considère la partition P_9 comme peu adéquate du fait du manque de compacité des réécritures des deux clusters de gauche. Les expérimentations effectuées montrent donc la complémentarité de ces deux approches.

2.3 Elicitation de descripteurs à partir des données

La définition d'un vocabulaire personnel permettant une interrogation ou une analyse efficace de données est une tâche si cruciale et délicate que la définition d'indices d'adéquation n'est pas suffisante. Dans la continuité des travaux présentés dans la section 2.2, nous (M.J. Lesot, O. Pivert et moi-même) avons proposé des méthodes d'aide à la construction de vocabulaire prenant en compte la particularité des données concernées. Afin d'accélérer l'appropriation du vocabulaire par l'utilisateur en vue de son utilisation, nous considérons important que ces méthodes soient interactives et non automatiques [87, 62]. Toujours en considérant des données pouvant être structurées en classes, nous avons défini des méthodes permettant d'identifier des zones d'imprécision

correspondant à des transitions entre classes distinguables dans l'espace initial de définition des données.

Révision d'un vocabulaire pour une meilleure adéquation

Nous avons dans un premier temps considéré un vocabulaire utilisateur prédéfini et cherché à identifier des modifications permettant d'améliorer son adéquation vis-à-vis des données concernées. Pour chaque modalité du vocabulaire, les clusters dont le centre appartient au noyau de cette modalité sont identifiés. Si deux centres de clusters séparables dans l'espace numérique/catégoriel de définition sont réécrits par la même modalité, alors une scission de la modalité concernée est suggérée afin de pouvoir distinguer linguistiquement ces deux clusters. Les bornes des nouvelles modalités suggérées sont définies à partir de l'identification de zones dites de non-imprécision [87] où les objets des deux classes sont complètement séparés.
Cette stratégie de révision d'un vocabulaire a été utilisée sur les données artificielles et les vocabulaires illustrés dans la figure 5. Comme le montre la figure 6, des modifications sont suggérées pour les vocabulaires P_1, P_{2a} et P_{3l} afin de mieux distinguer sur la dimension en abscisse les clusters séparables en bas de la figure. Les expérimentations ont montré que cette stratégie permettait d'améliorer l'adéquation d'un vocabulaire au jeu de données auquel il se rapporte sans pour autant nuire à son interprétabilité.

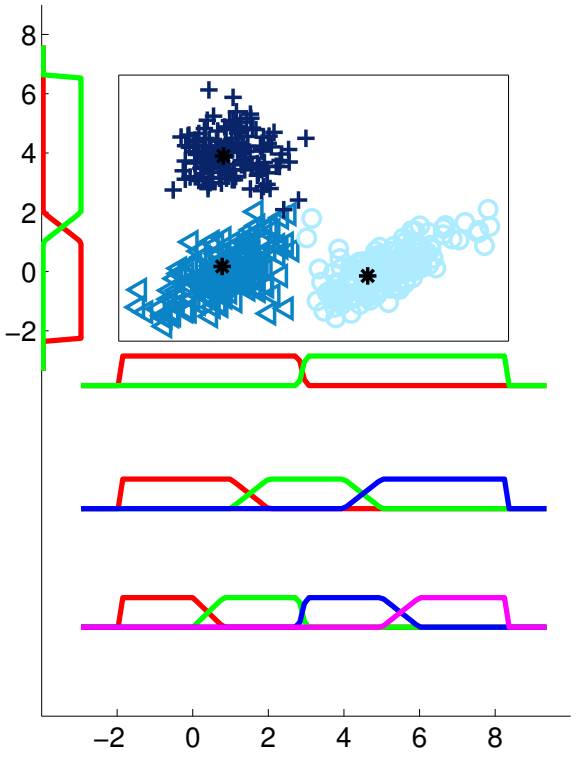

FIGURE 6 – Suggestions de découpage de modalités pour une meilleure adéquation

Élicitation d'une base de vocabulaire

De manière complémentaire à ces premiers travaux sur la révision d'un vocabulaire, nous (toujours M.J. Lesot, O. Pivert et moi-même) avons travaillé sur des stratégies visant à initier la définition d'un vocabulaire à partir de données. En nous positionnant dans la problématique de décrire linguistiquement des classes de données, nous avons défini plusieurs méthodes pour identifier, en partant d'une modalité universelle décrivant la totalité d'un domaine de définition, des zones de coupe permettant de distinguer linguistiquement des classes séparables dans l'espace initial de définition. Quatre stratégies ont été identifiées et comparées selon deux critères : la qualité des description linguistiques qu'elles génèrent et la robustesse des coupes suggérées. La première stratégie sert de base car elle s'appuie sur une recherche gloutonne de la coupe optimale d'un point de vue de la qualité des descriptions obtenues. Les trois autres stratégies sont plus efficaces et basées sur des heuristiques, comme e.g. la recherche d'une zone de non incertitude entre deux classes adjacentes. La figure 7 montre les vocabulaires suggérés. Une fois un vocabulaire suggéré, une ultime étape de vérification est appliquée pour reconsidérer des coupes jugées trop artificielles séparant linguistiquement des classes non distinguables dans l'espace initial de définition.

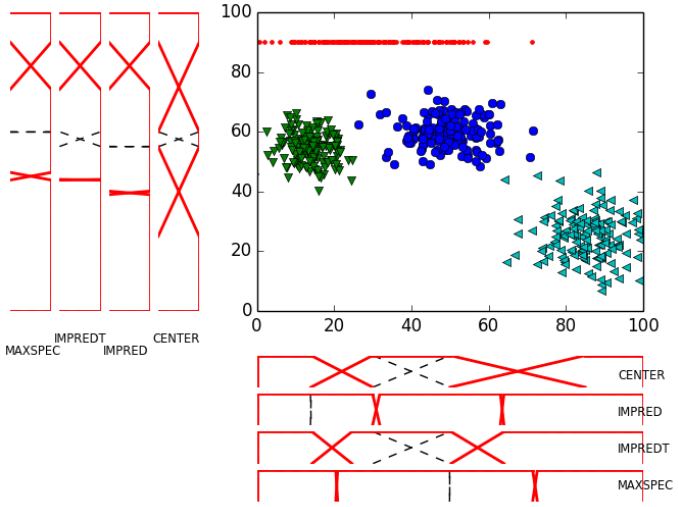

FIGURE 7 – Élicitation de vocabulaires

Les expérimentations ont montré que la stratégie basée sur la détection de zones de non incertitude des données fournissait un vocabulaire de qualité, vis-à-vis de la tâche de description linguistique des classes, quasiment aussi intéressant que l'approche gloutonne mais avec une meilleure robustesse. Nous avons également montré que cette approche permettait d'obtenir une première version d'un vocabulaire contenant les principales coupes des domaines de définition caractérisant les classes, et que l'utilisateur pouvait effectuer des coupes additionnelles pour préciser certains descripteurs sans que cela nuise à la capacité descriptive du vocabulaire.

2.4 Construction graphique d'un vocabulaire

D'un point de vue plus technique, j'ai encadré un stage court (5 semaines) d'un élève ingénieur de l'ENSSAT visant au développement d'une interface graphique de construction d'un vocabulaire utilisateur. Le processus de définition du vocabulaire est le suivant. Tout d'abord l'utilisateur téléverse un jeu de données brutes (format *csv*) contenant la description d'objets (un par ligne) selon différents attributs numériques et catégoriels (les colonnes). L'application détermine automatiquement le type de chaque attribut et son domaine de définition. L'utilisateur peut ensuite discrétiser graphiquement le domaine de définition des attributs qu'il juge intéressants à l'aide d'une partition floue (Fig. 8). Il est prévu d'intégrer rapidement dans ce prototype les méthodes coopératives d'aide à la construction de vocabulaire décrites précédemment.

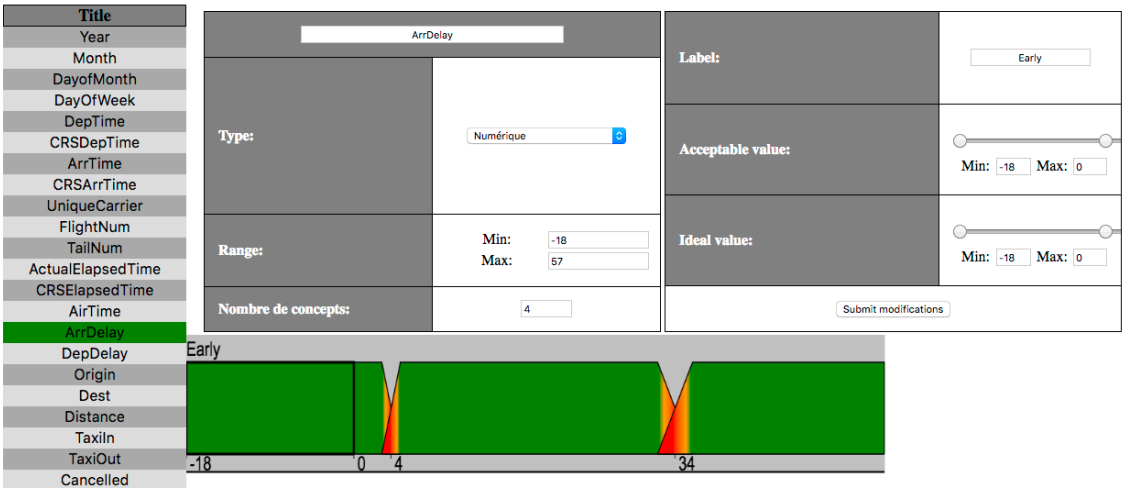

FIGURE 8 – Interface graphique de construction de vocabulaire

3 Personnalisation et enrichissement des méthodes d'accès

Un vocabulaire utilisateur, tel que défini dans la section précédente, est donc composé d'un ensemble de descripteurs linguistiques, un descripteur étant un couple associant un label et un sous-ensemble flou. La capacité descriptive d'un vocabulaire peut tout d'abord être exploitée pour construire des représentations synthétiques et personnalisées des données. Cette fonctionnalité a été très largement étudiée par la communauté du *soft computing* à travers la proposition de différentes approches dites de résumé linguistique de données. Les approches de résumé linguistique s'appuient généralement sur un patron syntaxique composé d'un quantificateur et de termes issus d'un vocabulaire utilisateur pour décrire les principales tendances observées dans un jeu de données [7] (e.g. « la majorité des jeunes employés a un salaire moyen »). Dans la seconde partie de cette section (Sec. 3.6), je présente mes contributions sur la construction de représentations synthétiques et personnalisées de données en vue de fournir des fonctionnalités d'exploration d'un jeu de données.

Comme évoqué dans la section précédente, un descripteur linguistique utilisé dans une condition de sélection floue peut être perçu comme l'expression d'une préférence sur les objets recherchés. De nombreux travaux ont montré l'apport de la prise en compte de telles préférences pour améliorer l'expressivité des langages d'interrogation en permettant la définition de prédicats flous [48, 134, 8, 100]. Cette démarche répond à un besoin exprimé par les utilisateurs de Systèmes de Gestion de Bases de Données (SGBD) pour plus de flexibilité dans la formulation de leurs besoins d'information [16]. Qualifier un système d'interrogation de données de flexible, c'est reconnaître sa capacité à s'adapter aux besoins de l'utilisateur final afin de lui retourner les informations les plus appropriées à sa demande. Dans cette première partie, je me focalise sur la personnalisation des interfaces d'interrogation (sous-section 3.2) et l'aide à la construction de requêtes floues (sous-section 3.4), deux aspects qui permettent de rendre les systèmes d'accès aux données plus flexibles. Cette démarche vers des systèmes flexibles et intelligents sera complétée par des contributions autour de l'enrichissement des réponses retournées par ces systèmes dans la partie III.

3.1 Contexte et motivations

La personnalisation des langages d'interrogation de Bases de Données (BD) par la prise en compte de préférences utilisateur est une problématique qui a suscité un grand intérêt auprès des communautés scientifiques de BD et de *soft computing* [93, 133, 30, 76, 148]. L'équipe BADINS, rebaptisée PILGRIM puis SHAMAN, est devenue un acteur majeur dans le domaine de l'interrogation flexible de BD en contribuant pendant près de trente années à la conception de stratégies d'interrogation floue.

Afin d'identifier clairement les contours de mes contributions sur cette problématique de l'interrogation floue de BD, commençons par un historique des travaux réalisés par mes collègues avant mon arrivée en 2009. Dans la continuité des travaux initiés par H. Prade et C. Testemale [112], l'équipe BADINS a défini une algèbre complète d'interrogation floue [100], puis a formalisé un langage d'interrogation nommé SQLf [17]. Basé sur la syntaxe de SQL, le langage SQLf permet la définition de conditions floues de sélection et de jointure comme illustré ci-dessous par le squelette syntaxique d'une requête SQLf :

SELECT [distinct] [α, k | k | α] <attributs projetés>
FROM R1 JOIN R2 ON <condition floue>...
WHERE <condition floue> ;

L'algèbre étendue, qui sert de base à un langage d'interrogation orienté utilisateur, constitue le fondement indispensable au développement de systèmes d'interrogation floue de BD.

3.2 Quelques contributions au langage SQLf

Groupement de tuples à partir d'une partition floue

À mon arrivée dans l'équipe PILGRIM en 2009, j'ai eu l'occasion de travailler avec Patrick Bosc et Olivier Pivert sur l'enrichissement du langage SQLf. Nous avons notamment défini une extension de la clause de groupement de SQL (clause *group by*) afin de prendre en compte des conditions de groupement flexibles [19]. Ainsi, au lieu de regrouper les tuples sélectionnés ayant une valeur commune sur un attribut donné (la condition booléenne de groupement en SQL), une clause supplémentaire (la clause *USING*) a été introduite dans *SQLf* afin de préciser les valeurs d'une variable linguistique qui serviront de conditions flexibles de groupement. La clause de sélection de groupe (clause *HAVING* de SQL) a également été étendue pour permettre d'exprimer des conditions flexibles sur :

— l'inclusion graduelle du résultat d'une fonction d'agrégation appliquée sur chaque groupe vis-à-vis d'une sous-requête Q' potentiellement floue,
HAVING count(*) $\geq$ (Q')

— la comparaison du résultat de deux fonctions d'agrégation,
HAVING $agg^1 \theta agg^2$ où θ est un opérateur de comparaison

— la satisfaction du résultat d'une fonction d'agrégation vis-à-vis d'un SEFs.
HAVING $agg\ is\ \phi$ où ϕ est l'étiquette linguistique d'un SEF

La requête ci-dessous permet de connaître la distribution des personnes habitant à Lannion selon leur catégorie d'âge :

SELECT label(age), count(*)
FROM personnes
WHERE ville ='Lannion'
GROUP BY label(age)
HAVING count(*) is 'élevé'
USING part(age) = {'enfant', 'adolescent', 'adulte', 'sénior'} ;

Ainsi, la clause de groupement est toujours utilisée pour créer une partition complète de l'ensemble des tuples sélectionnés par la clause *WHERE*, mais cette partition est désormais floue car un tuple peut appartenir, à des degrés divers mais dont la somme fait un, à plusieurs groupes.

Outre le fait d'améliorer encore l'expressivité du langage SQLf, nous avons montré dans l'article [19] que cette clause de groupement graduelle pouvait s'avérer très intéressante pour générer des résumés linguistiques des données et identifier des règles d'association floue à partir de données relationnelles.

Usage étendu des opérateurs de comparaison

Dans la clause de sélection d'une requête formulée à l'aide du langage SQLf, les descripteurs linguistiques sont initialement utilisés pour former des prédicats flous exprimés à l'aide de l'opérateur *is* : « *prix est bas* ». L'opérateur *is* constitue l'extension de l'opérateur classique d'égalité de SQL permettant de quantifier le degré de satisfaction d'une valeur vis-à-vis du descripteur. En collaboration avec Olivier Pivert et Pierre Nerzic, nous avons travaillé à la définition de comparateurs possédant une sémantique étendue et permettant de comparer un scalaire s avec un ensemble flou F [45]. Ces comparateurs permettent d'exprimer des conditions telles que $s \geq_f F$, $s \leq_f F$, $s >_f F$ ou encore $s <_f F$ qui se traduisent linguistiquement par "s est au moins F", "s est au plus F", "s est plus que F" et "s est moins que F".

Pour illustrer un usage de ces nouveaux opérateurs, la requête suivante permet d'obtenir la liste des vins de grand cru qui ne sont pas encore arrivés à maturité.

```
SELECT *
FROM maCave
WHERE type is 'grand cru' and age <f 'mature';
```

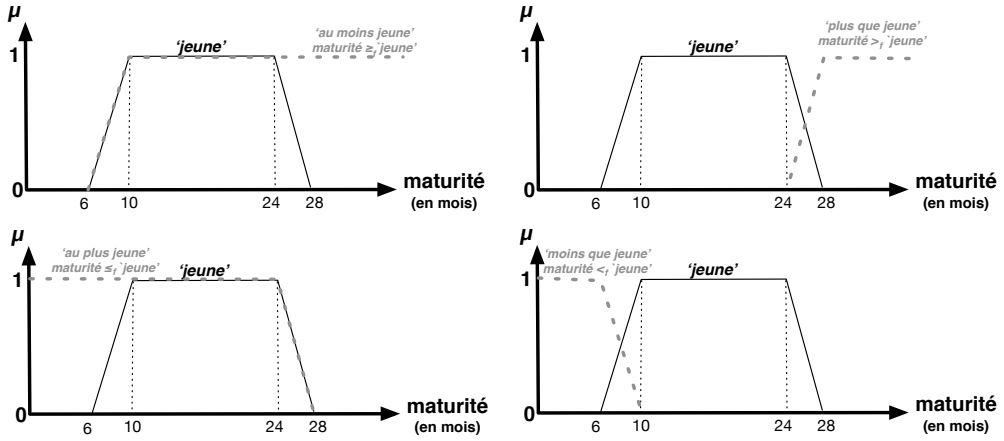

FIGURE 9 – Opérateurs de comparaison étendus aux SEFs

Nous avons dans un premier temps donné une définition intuitive de ces opérateurs illustrés par la figure 9. Ainsi, l'opérateur $s >_f F$ permettant de retourner les tuples dont la valeur sur l'attribut concerné se trouve à droite du SEF F est défini comme suit :

$$\mu_{>_f}(x, F) = \begin{cases} 1 & si\, x \geq B + b \\ 0 & si\, x \leq B \\ \frac{x-B}{b} & sinon. \end{cases}$$

Nous avons ensuite cherché à systématiser la définition de ces opérateurs ($>_f$, $<_f$, $\geq_f$ et $\leq_f$) en appliquant le principe d'extension proposé initialement par LA. Zadeh [145] puis repris par R.R. Yager [142] :

$$\mu(x\, \theta\, F) = \sup_{y \in F} \min(\mu_F(y), \mu_\theta(x, y)).$$

Cependant, l'application du principe d'extension pour les comparateurs $<$ et $>$ n'aboutit pas à la sémantique souhaitée. En effet, l'application du principe d'extension pour le comparateur $>$ (resp. $<$) retourne le même ensemble flou que pour le comparateur $\geq$ (resp. $>$) hormis la frontière gauche (resp. droite) qui est exclue. Nous avons ensuite montré que les opérateurs de comparaison étendus pouvaient être définis à travers une représentation possibiliste des évènements $s >_f F$, $s \geq_f F$, etc. Par exemple, la valeur retournée par la comparaison d'un scalaire avec un ensemble floue F

selon le comparateur $\geq$ correspond à la possibilité de l'événement $s \geq_f F$, c'est-à-dire la valeur de vérité de :

$$\exists y \in D, s \geq y \wedge \mu_F(y).$$

La possibilité de l'événement $s \geq_f F$, notée $\Pi(s \geq_f F)$ est ainsi quantifiée à l'aide de la formule suivante :

$$\Pi(s \geq_f F) = \sup_{y \in D} \min(s \geq y, \mu_F(y)).$$

La version stricte des comparateurs s'exprime comme la nécessité de l'évènement $s >_f F$ obtenue à partir de la possibilité de son évènement contraire :

$$N(s >_f F) = 1 - \Pi(s \leq_f F).$$

3.3 Interrogation flexible de données liées

Bien que les bases de données relationnelles soient encore largement utilisées, en particulier pour les applications classiques de gestion dans les entreprises, le besoin de gérer des données plus complexes s'est fait sentir depuis déjà plusieurs décennies, et a conduit à l'émergence d'autres modèles de données. Ainsi, un concept a fait son apparition ces dernières années, et a attiré l'attention de nombreux chercheurs de la communauté des bases de données, celui de base de données graphe [4]. En collaboration avec Virginie Thion, Olivier Pivert et Olfa Slama, nous avons transposé les apports de l'interrogation floue, en termes de flexibilité et d'expressivité, observés dans le modèle relationnel vers cette représentation particulière des données. Cette transposition n'est évidemment pas directe car la structuration des données en graphe, potentiellement flou [119, 88, 28], modifie la nature des éléments atomiques manipulés (passage des relations aux graphes) ainsi que des besoins d'information exprimés dans les requêtes. Une requête a en effet pour objectif d'identifier l'ensemble des sous-graphes qui satisfont des conditions exprimées sur leur structure (longueur, poids des chemins, connectivité, etc.) et/ou les valeurs attachées aux nœuds et aux arcs. Un exemple de graphe incluant des arcs flous entre nœuds est illustré par la figure 10. Le degré associé à un arc exprime l'intensité d'un lien entre deux nœud, comme e.g. à quel point deux auteurs peuvent être considérés comme co-auteurs.

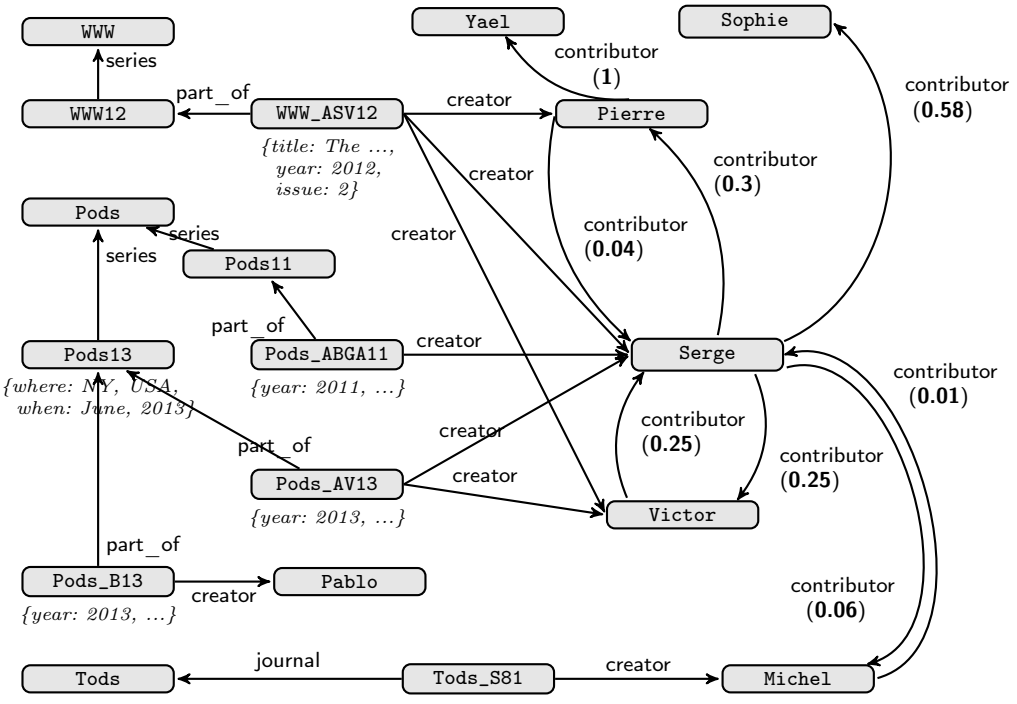

FIGURE 10 – Exemple d'un graphe de données floues

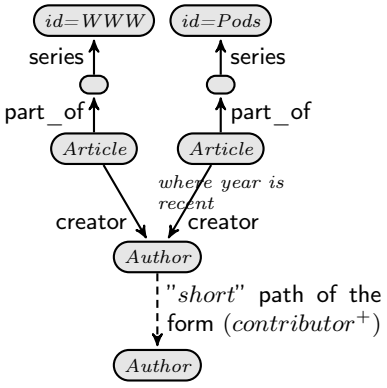

FIGURE 11 – Patron de sélection $\mathcal{P}$

Pour doter les systèmes de bases de données graphe de fonctionnalités d'interrogation floue, un formalisme de représentation de conditions floues ainsi qu'une algèbre étendant les opérateurs classiques de sélection et projection ont tout d'abord été proposés [111, 110]. La notion de patron de graphe flou a été formalisée pour exprimer des conditions sur les propriétés associées aux nœuds, arcs et chemins du graphe. La figure 11 illustre graphiquement un patron permettant de retrouver dans le graphe de la figure 10 les sous-graphes représentant des contributeurs proches ayant publié un article dans *WWW* et un autre dans *Pods*.

L'opérateur de sélection σ visant à identifier les sous-graphes d'un ensemble, potentiellement flou, de graphes $\mathcal{G}$ satisfaisant un patron structurel $\mathcal{P}$ est défini de la façon suivante :

$$\sigma_{\mathcal{P}}(\mathcal{G}) = \{\langle s, \min(d,\ \mu_{\mathcal{P}}(s))\rangle \,|\, \mu_{\mathcal{P}}(s) > 0\},$$

où s est un sous-graphe de g tel que $\langle g, d\rangle \in \mathcal{G}$ (d est le degré d'appartenance de g au SEF $\mathcal{G}$).

La seconde étape vers le développement d'un système d'interrogation floue de données graphe fut de concevoir un langage permettant la définition de descripteurs linguistiques et de conditions floues. Un langage nommé FUDGE [110] inspiré de la syntaxe de CYPHER [4] a ainsi été proposé. Le patron syntaxique de la figure 11 peut e.g. être traduit selon la syntaxe de FUDGE comme suit :

```
DEFINEDESC short AS (3,5)
DEFINEASC recent AS (2010,2014)
IN
MATCH
(ar1:Article)-[part_of.series]->(s1),
(ar2:Article)-[part_of.series]->(s2),
(ar1)-[:creator]->(au1:Author),
(ar2)-[:creator]->(au1:Author),
(au1)-[(contributor+)|Length IS short]->(au2:Author)
WHERE
s1.id=WWW AND s2.id=Pods AND ar2.year IS recent;
```

Ce travail sur la formalisation de conditions floues et personnalisées dans des requêtes soumises à des BD graphes a été ensuite étendu au cas particulier des données liées sous forme de triplets (i.e. *RDF*) et à son langage *SPARQL*. Le langage *FURQL* [102], en tant qu'extension de *SPARQL*, permet de définir des conditions floues sur les valeurs d'objets ou les prédicats associés à un sujet [27]. Un exemple d'une requête recherchant les artistes dont presque tous les albums sont faiblement notés par un ami proche est exprimée en *FURQL* ci-dessous :

```
select ?Art1
WHERE {
  ?Art1 (friend+ | Length IS short) ?Art2.
  ?Art2 creator ?Alb.
  ?Alb rating ?rate.
  ?Art1 (recommend | almostall) ?Alb.
   filter (?rate IS low).}
```

4. https://neo4j.com/developer/cypher-query-language/

3.4 Construction graphique et interactive de requêtes floues

Le développement de langages est un premier pas pour prouver la faisabilité des approches floues d'interrogation de données et montrer leur apport en termes d'expressivité et de flexibilité. Ces langages formels ne sont cependant accessibles qu'aux informaticiens et non aux experts métiers qui constituent les utilisateurs finaux les plus enclins à apprécier l'enrichissement des méthodes d'accès aux données. Un second enjeu auquel il a fallu répondre a consisté à rendre la définition de requêtes floues intuitive et montrer que la vision booléenne que les utilisateurs ont des interfaces d'interrogation n'est que le résultat des fonctionnalités limitées qui leur ont été proposées jusqu'à présent.

L'interface ReqFlex

Dans un premier temps, nous avons conçu une interface graphique de construction de requêtes floues [128]. L'objectif de ce travail était de montrer que la personnalisation et l'amélioration de l'expressivité des interfaces d'interrogation ne se traduisait pas forcément par une diminution du caractère intuitif de leur utilisation. ReqFlex est une interface graphique de construction de requêtes floues servant à interroger un SGBD PostgreSQL disposant du module d'extension PostgreSQLf (Sec. 5.2). Comme l'illustre la figure 12, l'utilisateur peut graphiquement construire ses propres descripteurs linguistiques ou bien s'appuyer sur ceux définis par d'autres utilisateurs. Graphiquement, ces descripteurs peuvent être combinés de manière conjonctive, disjonctive ou bien en utilisant des quantificateur prédéfinis (comme e.g. « la plupart », « autant que possible », « la majorité » ou « quasiment tous »), pour former une requête floue. La figure 13 montre comment un système de glisser-déposer permet de construire des requêtes combinant, selon différentes sémantiques, des conditions floues exprimées à l'aide de descripteurs linguistiques. En plus de pouvoir calibrer l'ensemble des résultats avec l'application de seuils qualitatif et/ou quantitatif, la visualisation de la relation floue retournée permet à l'utilisateur d'identifier facilement les résultats satisfaisant au mieux les critères de sa requête.

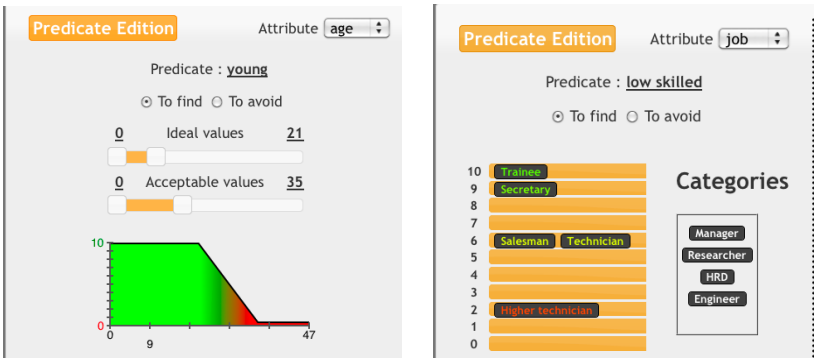

FIGURE 12 – Définition graphique de descripteurs linguistiques avec ReqFlex

Construction de requêtes floues pour BD graphe

Le développement de l'interface ReqFlex a permis de prouver, par la pratique, la maturité de l'interrogation floue de données relationnelles et également le caractère intuitif de son utilisation. Un travail analogue a ensuite été mené pour proposer une interface d'interrogation floue de données graphes. Ce travail a davantage vocation à être utilisé comme prototype de recherche pour illustrer l'apport de la logique floue sur l'expressivité des langages d'interrogation et la distinguabilité des résultats générés. En effet, l'interface SUGAR (Fig. 14) ne prend en entrée que des requêtes exprimées à l'aide du langage formel FUDGE (Sec. 3.4). Ce développement a surtout montré la possibilité de développer des interfaces d'interrogation floue au dessus de SGBD (ici Neo4J) dédiés à l'interrogation booléenne de BD graphe. Le mécanisme d'exécution de requêtes FUDGE au dessus de Neo4J est décrit dans la partie II (Sec. 5).

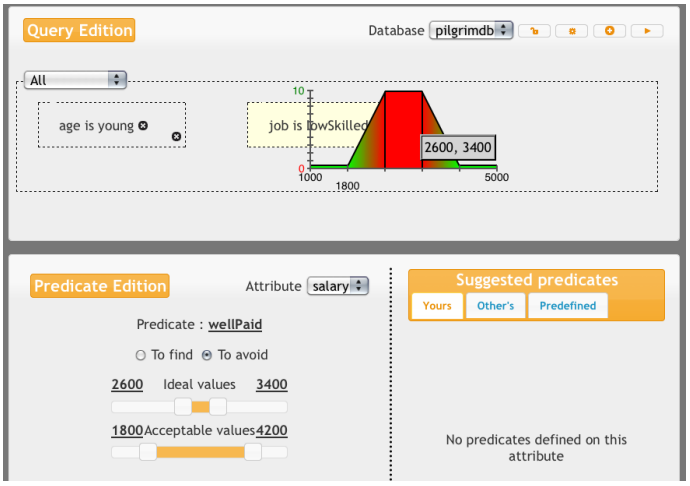

FIGURE 13 – Construction intuitive de requêtes floues avec ReqFlex

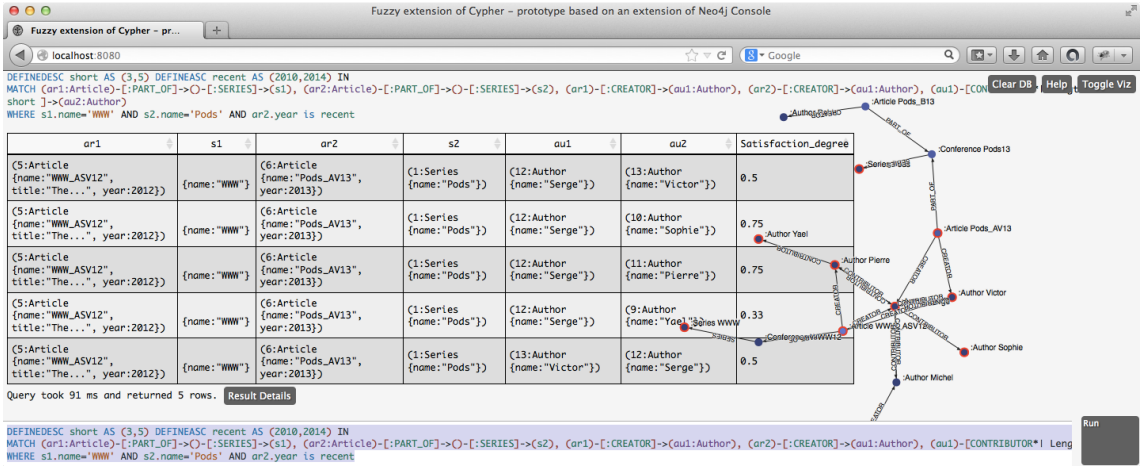

FIGURE 14 – SUGAR : interrogation et visualisation de graphes flous

3.5 Aide à la construction de requêtes par mots-clés

Pour rendre plus aisée l'expression de requêtes à l'aide de langages formels, deux stratégies ont principalement été envisagées. La première repose sur le développement d'interfaces graphiques d'interrogation (e.g. ReqFlex) et la seconde sur la définition d'un langage intermédiaire plus simple d'accès inspiré des moteurs de recherche. Les approches d'interrogation par mots-clés, largement démocratisées pour la recherche de documents sur Internet, correspondent à cette seconde stratégie. De nombreux travaux ont été réalisés par la communauté des bases de données pour transposer ces méthodes d'interrogation par mots-clés au cas particulier des données structurées selon un modèle relationnel [69], sémantique [79] ou de type entrepôt [40].

En m'appuyant sur les compétences acquises lors de mon doctorat en traitement automatique des langues, j'ai travaillé sur l'enrichissement des méthodes d'interrogation par mots-clés pour données structurées. L'objectif de ces travaux était d'améliorer l'expressivité des requêtes par mots-clés et ainsi proposer des stratégies d'interrogation à mi-chemin entre les pseudo-langages contraints et les requêtes composées d'une simple énumération non structurée de mots-clés.

Construction interactive de requêtes SPARQL par mots-clés

Dans le cadre d'un contrat de recherche externe avec la société SEMSOFT [5] spécialisée en médiation de données, j'ai eu l'occasion de travailler sur l'interrogation par mots-clés de données sémantiques. L'outil de médiation de données conçu par SEMSOFT offre un point d'accès SPARQL permettant d'interroger des sources de données distribuées dont la sémantique est unifiée par un schéma *rdfs* médiateur. L'objectif des travaux que j'ai menés était de définir une méthode intuitive et pragmatique fournissant une traduction SPARQL d'une requête exprimée à l'aide de mots-clés. Contrairement aux méthodes classiques d'interrogation par mots-clés où l'expression d'un besoin d'information est réalisé à l'aide de quelques mots non liés, l'idée avancée consiste à assister l'utilisateur dans le choix et l'enchaînement des mots-clés afin d'aboutir à une expression en pseudo-langage naturel à la fois expressive et surtout moins ambiguë. L'approche présentée dans [130] exploite un mécanisme d'auto-complétion pour permettre à l'utilisateur de naviguer dans une ontologie et construire un sous-graphe connexe particulier (arbre de Steiner) représentant une requête. Le parcours guidé de l'ontologie permet d'activer des classes, des propriétés et des valeurs décrites linguistiquement par des étiquettes. La succession de ces étiquettes linguistiques forme la requête par mots-clés enrichie définie de manière interactive par l'utilisateur. La figure 15 illustre un extrait d'ontologie dont les différents éléments peuvent être activés par auto-complétion (Fig. 16) pour par exemple former la requête « name of person at the head of company and author of article about "business intelligence" ». L'utilisateur sélectionne tout d'abord le patron de projection « name of person » pour indiquer l'information souhaitée, puis se laisse guider par le système d'auto-complétion pour former un graphe connexe partant de la classe *foaf :Person*.

Une grammaire non contextualisée a été construite pour définir des patrons syntaxiques de labels (rdfs :label) exprimant des instructions de projection ou de sélection. Ces règles syntaxiques sont à la fois utilisées pour guider le processus d'auto-complétion et également pour traduire le sous-graphe en requête SPARQL. L'application de cette grammaire sur la requête illustrée par la figure 15 permet d'aboutir à la requête SPARQL suivante :

```
SELECT DISTINCT ?name
WHERE {
    ?name rdf :type foaf :lastName.
    ?person rdf :type foaf :Person. ?person foaf :name ?name.
    ?person org :headerOf ?comp. ?person sch :author ?art.
    ?art rdf :type sch :NewsArticle. ?art sch :headline ?head.
    FILTER regex( ?head, "business intelligence") }
```

Ce système interactif de construction de requêtes à mots-clés par auto-complétion apporte une solution pragmatique et novatrice au problème de l'aide à l'interrogation de données liées. Ce premier travail sur l'interrogation par mots-clés de données structurées a donné lieu à un transfert

5. `https://www.semsoft-corp.com/fr/`

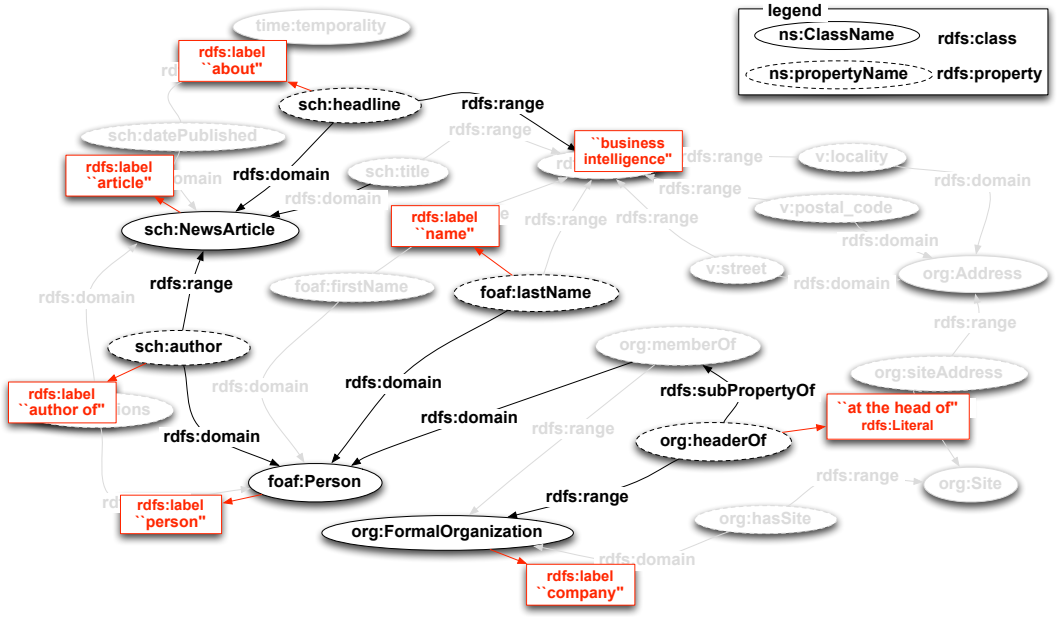

FIGURE 15 – Représentation de la requête en tant que sous-graphe de l'ontologie

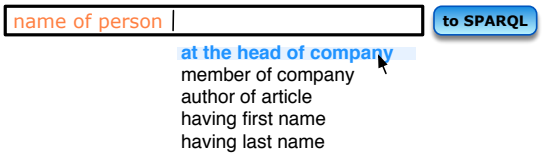

FIGURE 16 – Parcours interactif de l'ontologie et construction par auto-complétion de requête

technologique réussi depuis l'IRISA vers la société SEMSOFT à travers la commercialisation d'un outil nommé *Queries* intégré dans la suite logicielle *AGGREGO* (`https://semsoft-corp.com/fr/`).

Interprétation de requêtes par mots-clés et traduction en langage formel

L'outil *Queries* est un système d'aide à la construction de requêtes par mots-clés. Outre sa simplicité d'usage, le principal intérêt de l'outil *Queries* est qu'il garantit la conformité de la requête vis-à-vis de l'ontologie interrogée et le fait qu'elle soit traduisible en SPARQL. Ces atouts sont cependant obtenus au prix d'une absence de flexibilité, propriété pourtant propre aux approches initiales d'interrogation par mots-clés, car l'utilisateur doit spécifier, par sélection successive des labels suggérés, le sous-graphe complet correspondant à sa requête.
Permettre à l'utilisateur de construire plus librement sa requête implique la définition d'une stratégie d'interprétation des mots-clés visant à éliciter le sens de la requête caché derrière une succession de mots-clés. Cette problématique a été largement étudiée par la communauté des bases de données selon deux approches : l'une correspondant à la transposition des méthodes statistiques issues de la recherche d'information (comme e.g. TF-IDF) [114], la seconde basée sur la reconstruction d'un sous-graphe connexe à partir des éléments du modèle de données activés par les mots-clés [69].

L'approche, nommée COKE (pour COnnected KEywords) [131], suit cette seconde stratégie en cherchant à construire les sous-graphes pouvant représenter les différentes interprétations de la requête par mots-clés. La singularité du système COKE provient de l'expressivité de son langage d'interrogation positionné à l'intersection des systèmes d'interrogation en langue naturelle et des systèmes par mots-clés. En effet, la quasi-totalité des systèmes d'interrogation par mots-clés de données structurées ne prennent en compte que les mots-clés de la requête associés à des valeurs ou des descripteurs structurels du modèle de données (nom de colonne, nom de table, etc.). Le constat à l'origine du système COKE, inspiré d'une approche empirique d'analyse syntaxique [137], est que les connecteurs grammaticaux jouent un rôle très important dans l'interprétation de la structure et du sens des phrases. Ainsi, lorsqu'un utilisateur formule les requêtes « film de C. Eastwood » et « film avec C. Eastwood » la distinction sémantique entre ces deux requêtes repose sur l'interprétation des connecteurs *de* et *avec*, mots pourtant trop fréquents dans la langue française pour être pris en compte par les approches statistiques [113]. De plus, lorsqu'un utilisateur interroge une source de données structurées, dont le modèle représente sa sémantique, il est en mesure d'attendre des résultats plus précis à sa requête que lors d'une recherche de documents (non structurés) sur Internet. Ainsi, les résultats des requêtes « film de C. Eastwood » et « film avec C. Eastwood » doivent être différents, car dans le premier cas C. Eastwood est à considérer comme un producteur ou un réalisateur et dans le second cas comme un acteur.

L'approche COKE [131] s'appuie sur une modélisation en graphe étiqueté du schéma relationnel d'une base de données [77] et vise à construire des sous-graphes connexes constituant les interprétations possibles de la requête soumise par l'utilisateur. En termes d'expressivité, COKE est à la fois capable d'interpréter des requêtes composées d'une énumération de mots-clés (comme e.g. « film Eastwood »), mais également d'identifier une structure syntaxique en s'appuyant sur la sémantique de connecteurs grammaticaux. Pour ce faire, une grammaire composée de syntagmes nominaux représentant des structures de sous-graphes du schéma de la BD a été définie afin d'identifier les instructions exprimant des projections ou des sélections. Ainsi, le système COKE détermine le sens correct et précis de requêtes plus complexes, telles que « title of movies produced by Andy Warhol after 1960 », afin de retourner le résultat exact attendu. En termes de flexibilité de l'interface d'interrogation, la construction de la requête est également assistée par un mécanisme d'auto-complétion qui incite l'utilisateur à définir des instructions de projection et de sélection non ambiguës à l'aide de connecteurs grammaticaux. Cependant, par rapport à l'outil *Queries*, l'utilisateur n'est pas contraint de construire un sous-graphe connexe complet représentant sa requête, mais peut définir des instructions qui seront ensuite reliées par des jointures. La figure 17 montre comment l'interprétation de la requête « title genre productionYear of movies with Gene Hackman by C. Eastwood » conduit à l'identification de trois sous-graphes représentant une instruction de projection et deux instructions de sélection qui sont par la suite reliées par des jointures. Le prin-

cipe du plus court chemin entre les classes activées par les instructions de projection et de sélection est exploité pour décider quelles jointures sont à privilégier.

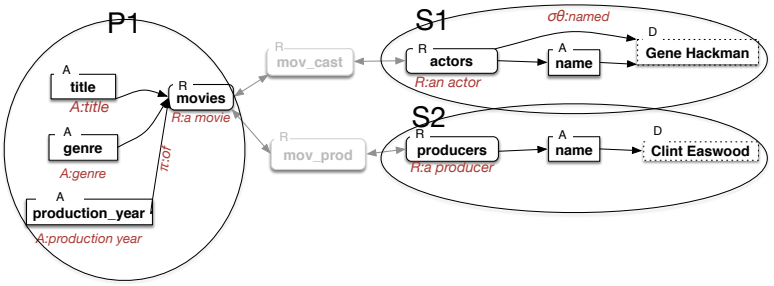

FIGURE 17 – COKE : reconstruction de la requête à partir d'instructions isolées de projection et de sélection

L'approche COKE a été implémentée pour interroger la base de données musicale MusicBrainz [6] et évaluée à partir du corpus de requêtes en langue naturelle issu de la compétition QALD [138]. L'objectif de cette expérimentation était tout d'abord de prouver l'expressivité de la grammaire proposée et de montrer que l'usage de connecteurs grammaticaux associés en tant que labels aux liens entre nœuds permettait d'exprimer des requêtes complexes. Ainsi sur les 100 requêtes en LN du corpus QALD, seulement 11 ne peuvent pas être directement traduites sous forme de requête COKE car elles correspondent à des requêtes booléennes (« Is Liz Story a person or a group ? ») non gérées actuellement par COKE. Du fait de la traduction manuelle des requêtes LN en requêtes COKE, la traduction SQL exacte est générée par COKE pour 100% des requêtes couvertes. La prise en compte de connecteurs grammaticaux, couplée à un système d'auto-complétion pour la construction des requêtes, permet de réduire considérablement le nombre d'interprétations possibles et donc symétriquement d'atteindre des temps de traitement très faibles (0,65s. en moyenne pour COKE sur les 89 requêtes et 31,6s. pour l'approche la plus efficace ayant concouru à la compétition QALD).

En m'appuyant sur les retours utilisateurs concernant l'outil *Queries* et le bilan des expérimentations menées sur le système COKE, j'ai ensuite travaillé, avec l'aide d'un post-doctorant (K. Dramé), au développement d'une approche d'interrogation par mots-clés de données liées. Ce travail s'inscrit dans le cadre d'un projet PME financé par le pôle Image & Réseau en collaboration avec les entreprises Semsoft et Predicsis [7]. La particularité de l'approche développée [42] repose sur un découpage en étapes successives du processus interactif de construction de la requête. L'idée est d'utiliser au départ un vocabulaire d'interrogation limité composé uniquement des labels associés aux différentes classes de l'ontologie ainsi qu'aux valeurs. Également assisté par un mécanisme d'auto-complétion, l'utilisateur sélectionne les principales notions (i.e. mots-clés) qui interviennent dans sa requête (e.g. « track memberOf The Cure 5minutes »). La seconde étape, réalisée automatiquement, consiste à identifier tous les chemins reliant les classes activées par les mots-clés de la requête initiale, puis à utiliser ces chemins pour former des arbres de Steiner qui constituent les différentes interprétations de la requête initiale. Ces interprétations sont traduites dans un pseudo-langage naturel afin que l'utilisateur puisse choisir précisément l'interprétation correcte et lever toute ambiguïté (Fig. 18).

Lors d'une dernière étape, l'utilisateur peut raffiner sa requête en précisant les propriétés à projeter ainsi qu'introduire des comparateurs ou des fonctions d'agrégation (Fig. 19). Un ensemble de règles de traduction a été défini pour passer d'un arbre de Steiner enrichi (avec des comparateurs et fonctions d'agrégation) à la requête SPARQL qui retournera le résultat final. Cette approche a été implémentée pour former un prototype de recherche nommé iKeys [124] et expérimentée sur la base MusicBrainz. Des expérimentations ont ensuite été réalisées à l'aide du corpus QALD. Ce prototype est actuellement en phase de transfert technologique avec l'entreprise Semsoft.

Globalement, les expérimentations menées pour évaluer la pertinence de ces approches d'in-

6. `http://musicbrainz.org`
7. `https://predicsis.ai/`

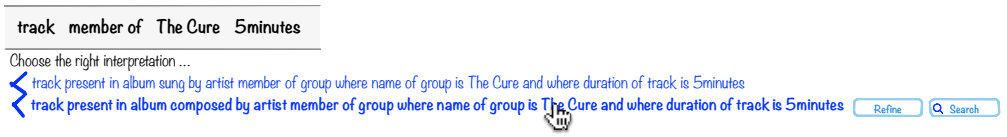

FIGURE 18 – iKeys : suggestion des interprétations possibles d'une requête initiale par mots-clés

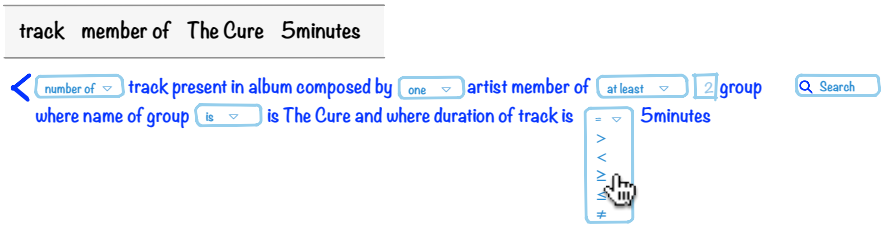

FIGURE 19 – iKeys : raffinement de la requête

terrogation par mots-clés montre que contraindre/assister l'utilisateur lors de la définition de ses requêtes permet à la fois de clarifier la recherche d'information et également d'obtenir des résultats plus précis. Ceci montre qu'améliorer l'expressivité des interfaces d'interrogation ne se traduit pas forcément par une diminution de la précision des résultats, mais ceci à condition de garder un certain contrôle sur le processus de construction des requêtes. C'est ce compromis que recherchent notamment les approches d'interrogation basées sur des langages naturels contrôlés.

3.6 Exploration de données

L'extraction de connaissances à partir de données n'est plus l'apanage des informaticiens, mais constitue désormais une activité quotidienne de nombreux professionnels (assureurs, responsables d'entreprises, élus, communicants, etc.). Les données que ces experts ont à analyser ne sont généralement pas structurées en relations ou classes liées, mais issues de fichiers bruts de type texte, CSV ou Excel. L'analyse autonome de données brutes étant visiblement un enjeu actuel crucial, j'ai étudié dans quelle mesure la modélisation de connaissances subjectives sur les données à l'aide de vocabulaire de description (Sec. 1.2) pouvait améliorer le processus d'appropriation des données par un expert métier. Deux approches d'exploration des données sont présentées dans cette section. La première repose sur un mécanisme de navigation par facettes dans les données où un vocabulaire prédéfini est utilisé pour réécrire des données numériques/catégorielles en labels linguistiques. La seconde exploite également un vocabulaire utilisateur pour générer un résumé linguistique des données, ce résumé pouvant ensuite être utilisé pour explorer les données.

Exploration par facette

Mes premiers travaux sur l'exploration intuitive de données ont porté sur la définition d'un système de navigation par facettes [65]. Ces interfaces de navigation proposent une alternative aux classiques formulaires de construction de requêtes en proposant à l'utilisateur d'atteindre un sous-ensemble d'objets intéressants en sélectionnant successivement les propriétés recherchées (e.g. $\text{type} = \text{monospace} \longmapsto \text{année} \in [2012 - 2017] \longmapsto \text{km} \leq 50000 \longmapsto \text{couleur} \in \{gris, blanc, noir\} \longmapsto \text{prix} \leq 15000$ €). Dans l'article [126], nous avons proposé une approche floue de la navigation par facettes basée sur l'exploitation d'un vocabulaire linguistique prédéfini (Sec. 1.2). La recherche s'effectue alors par sélection successive de descripteurs linguistiques. Outre le fait de simplifier l'interface graphique d'interrogation en proposant pour chaque facette (i.e. attribut) une liste restreinte de descripteurs linguistiques à la place d'un ensemble de valeurs numériques/catégorielles possibles, nous avons défini des mesures permettant de qualifier l'intérêt d'un descripteur selon :

— sa corrélation ou symétriquement son caractère surprenant vis-à-vis des descripteurs précédemment sélectionnés,

— ou sa sélectivité, c'est-à-dire sa capacité à sélectionner un nombre d'objets de bonne qualité aussi proche que possible d'un seuil quantitatif k fixé par l'utilisateur.

Pour quantifier ces différentes notions, l'approche proposée s'appuie sur un résumé des données réécrites à l'aide du vocabulaire, ce résumé étant composé de cardinalités floues [44]. Une cardinalité floue formellement notée $F_{\mathcal{D}}^{P} = \{n_1/\sigma_1, n_2/\sigma_2, \ldots n_f/\sigma_f\}$ indique combien d'éléments de l'ensemble de données $\mathcal{D}$ satisfont la condition P aux degrés σ_1, σ_2, etc., ces degrés appartenant dans notre cas à une échelle ordinale $\sigma_1 = 1 > \sigma_2 > \ldots > \sigma_f = 0^+$. Davantage de détails sont donnés sur la construction de ce résumé à base de cardinalités floues dans la section 6.2.

La quantification du degré de corrélation et de surprise d'un descripteur, disons v, par rapport à une conjonction de descripteurs déjà sélectionnés, disons Q, s'appuie sur le score de confiance de la règle d'association entre Q et v. Ce score de confiance noté $conf(Q \Rightarrow v)$ quantifie à quel point les objets satisfaisant les propriétés de Q satisfont également v. Le degré de corrélation quantifiant à quel point v est sémantiquement lié aux propriétés de Q est calculé de la façon suivante : $\mu_{cor}(v, Q) = \min(conf(Q \Rightarrow v), conf(v \Rightarrow Q))$. Symétriquement, le caractère surprenant de v par rapport à Q est défini comme suit : $\mu_{surp}(v, Q) = \min(1 - conf(Q \Rightarrow v), 1 - conf(v \Rightarrow Q))$. L'autre information, plus quantitative, associée à chaque descripteur suggéré concerne sa capacité à sélectionner un nombre d'objets proche du seuil k fixé par l'utilisateur. Cette sélectivité, notée $\mu_{sel}(Q \wedge v)$, est quantifiée par la formule :

$$\mu_{sel}(Q \wedge v) = sup_{0 \leq \alpha \leq 1} \min(1 - \frac{|\,|\Sigma_{Q \wedge v}^{\alpha}| - k\,|}{max(k, |\Sigma_{Q}^{\alpha}| - k)}, \alpha),$$

où Σ_{Q}^{α} est l'ensemble des objets satisfaisant Q à un degré supérieur ou égal à α. L'intérêt de maintenir un résumé des données sous forme de cardinalités floues est de disposer en temps constant de toutes les informations nécessaires au calcul des degrés de corrélation, de surprise et de sélectivité sans avoir à parcourir de nouveau les données.

La figure 20 illustre l'interface graphique d'un démonstrateur de navigation par facettes que nous avons implémenté au dessus de données réelles décrivant des voitures d'occasions. L'interface de navigation est découpée en cinq zones. La zone 1 décrit linguistiquement les descripteurs déjà sélectionnés (Q), la zone 2 liste les descripteurs corrélés à Q, la zone 3 les descripteurs surprenants, la zone 4 tous les autres descripteurs possibles et la zone 5 offre un aperçu des résultats de la requête actuelle Q.

Exploration à partir d'un résumé personnalisé des données

Lorsque le nombre de dimensions sur lesquelles les données sont décrites est important (≥ 10), l'utilisabilité des systèmes de navigation par facettes et la clarté de leur interface deviennent problématiques. En collaboration avec Olivier Pivert et Ronald R. Yager, nous avons travaillé à une approche d'exploration basée sur une représentation synthétique du contenu d'un jeu de données. L'utilisateur en charge de l'analyse d'un jeu de données construit tout d'abord son vocabulaire personnel de description des données. Bien que des méthodes existent pour construire automatiquement un vocabulaire à partir de données [86, 62, 125], nous considérons dans cette approche que l'utilisateur définit manuellement son propre vocabulaire notamment à l'aide d'interfaces graphiques intuitives [128]. Cette étape de définition manuelle des termes, bien que non triviale, est importante car elle permet à la fois à l'utilisateur de s'approprier la définition de chaque terme et également de sélectionner les attributs à considérer dans le résumé et pas uniquement ceux apparaissant comme statistiquement les plus informatifs [73].

Le jeu de données à analyser est ensuite réécrit selon les termes du vocabulaire utilisateur $\mathcal{V}$ pour obtenir sa représentation linguistique personnalisée sous forme d'un vecteur de réécriture, noté $RV_{\mathcal{D}}^{\mathcal{V}}$ (Sec. 1.3). Un algorithme de réécriture efficace et distribué est proposé dans la section 6.1.

Se pose alors la question de la restitution à l'utilisateur de cette réécriture afin qu'il dispose d'une vue synthétique des données. La construction d'un résumé de données est une problématique largement étudiée et maîtrisée par la communauté du *soft computing* [7]. Les approches existantes s'appuient généralement sur un protoforme syntaxique pour générer un ensemble d'assertions linguistiques décrivant les tendances dans les résumés (e.g. « la majorité des vols en soirée

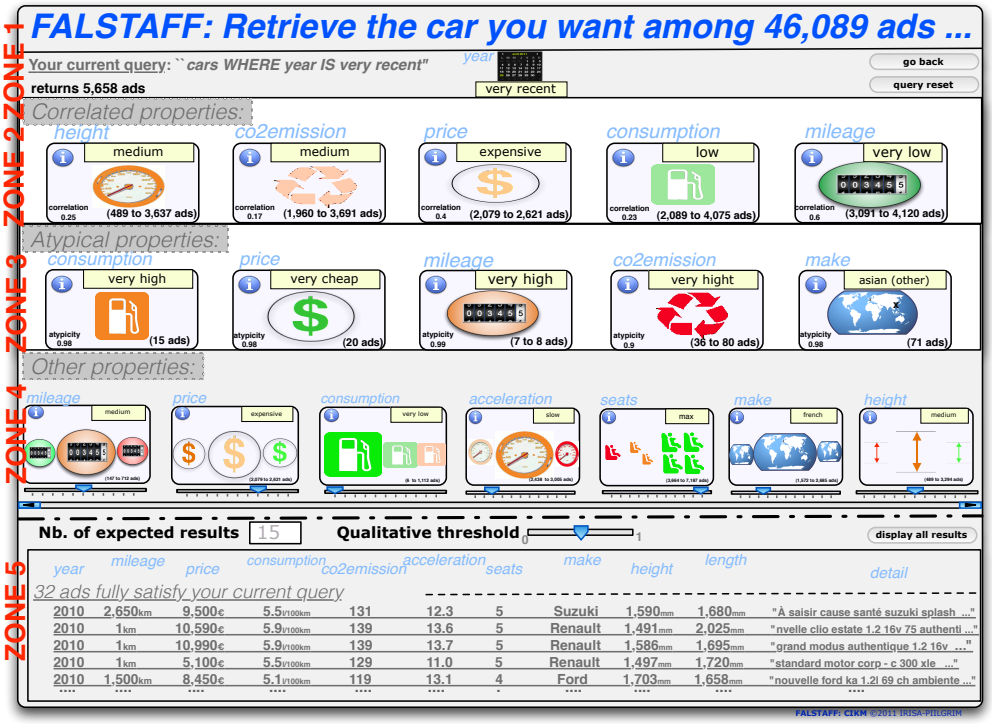

FIGURE 20 – Interface d'interrogation par facettes exploitant un vocabulaire utilisateur

ont du retard ») et une mesure qualifiant la véracité de l'assertion. Nos travaux, s'inscrivant dans un contexte applicatif d'informatique décisionnelle autonome (self-service business intelligence), privilégient une restitution graphique du résumé des données. La figure 21 suggère une restitution graphique possible du vecteur $\mathcal{RV}_{\mathcal{D}}$ avec un regroupement des termes portant sur le même attribut et une taille de police proportionnelle à la couverture de $\mathcal{D}$ par le terme (i.e. $\rho_v(\mathcal{D})$).

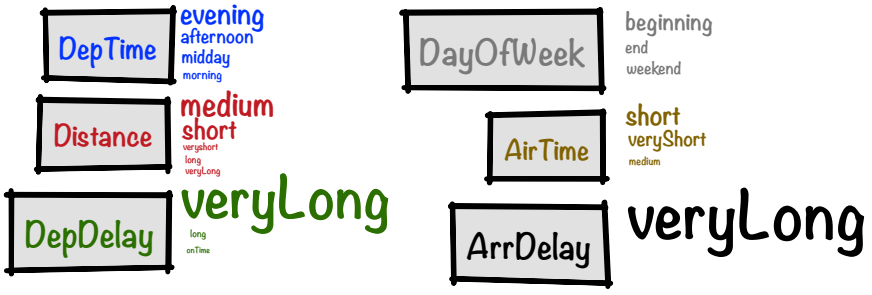

FIGURE 21 – Visualisation du vecteur de réécriture $\mathcal{RV}_{\mathcal{D}}^{veryLong\ ArrDelay}$ synthétisant les caractéristiques des vols retardés

Outre le fait d'offrir une vue synthétique et personnalisée des données, cette représentation peut également être utilisée comme interface d'exploration des données. La sélection (par clic) d'un terme, disons v, conduit à la récupération du sous-ensemble des objets $\mathcal{D}^v$ satisfaisant v : $\{d \in \mathcal{D}, \mu_v(d) > \alpha\}$, où α est un seuil de filtrage pouvant être ajusté par l'utilisateur. Ce sous-ensemble est ensuite réécrit à son tour pour obtenir le vecteur $\mathcal{RV}_{\mathcal{D}^v}$ qui est lui même représenté graphiquement (Fig. 21) afin que l'utilisateur puisse prolonger l'exploration des données. Les vecteurs de réécriture occupant une place infime en mémoire (quelques kilo-octets), il est aisé de stocker les différents vecteurs construits successivement pour proposer une exploration rétro-active. L'applicabilité de la fonctionnalité d'exploration des données repose sur la mise en œuvre de stratégies

efficaces de stockage et d'indexation des vecteurs de réécriture des tuples qui sont décrites dans la section 6.3.

3.7 Autres modèles de représentation des préférences

La modélisation de préférences est un domaine de recherche extrêmement vaste [120] et leur utilisation pour enrichir et rendre flexible les interfaces d'interrogation de BD a également été largement étudiée [16]. L'approche quantitative de modélisation de préférences basée sur des SEF a évidemment été prépondérante dans mes travaux et occupe également une majeure partie de ce document. À mon arrivée dans l'équipe PILGRIM, j'ai eu également l'occasion de travailler avec P. Bosc et O. Pivert sur d'autres modèles de préférence pouvant être utilisés pour l'interrogation flexible de BD. Lors de mon doctorat en traitement automatique des langues écrites, j'ai étudié en particulier un modèle de préférence qualitatif. Brièvement, mon sujet de thèse portait sur la définition de stratégies de contrôle des résultats, intermédiaires et finaux, générés par une chaîne de TAL séquentielle [67]. Ce contrôle repose sur l'agrégation de multiples critères d'évaluation, hétérogènes par nature et souvent non commensurables (statistique, morpho-syntaxique, sémantique, etc.), permettant de privilégier un sous-ensemble des interprétations candidates. L'approche que j'avais proposée [123] reposait sur la formalisation du contrôle en tant que problème de décision multi-critère basé sur l'utilisation de relations de surclassement [121]. Ma première contribution relative à l'interrogation flexible de BD a consisté à transposer mes connaissances sur le modèle de préférences par surclassement pour la sélection d'objets satisfaisant une requête à préférences.

Les travaux résumés dans cette section ont pour objectif de montrer que des modèles de préférence, potentiellement plus riches sémantiquement et plus simples à exploiter que le modèle *skyline* [9] prépondérant en BD, peuvent être utilisés pour proposer des fonctionnalités d'interrogation flexible.

Utilisation des relations de surclassement pour l'interrogation flexible de BD

Soit un ensemble d'objets $\mathcal{D}$ à comparer selon n préférences atomiques ($\mathcal{G} : \{G_1, G_2, \ldots, G_n\}$) définies sur un ensemble $\mathcal{A} : \{A_1, A_2, \ldots, A_n\}$ d'attributs. Un modèle de préférence qualitatif vise à construire entre deux objets, e.g. d et $d\prime$, une relation notée $d \succ d\prime$ (resp. $d \succeq d\prime$) exprimant le fait que d'après $\mathcal{G}$, d est préférable à (resp. au moins aussi intéressant que) $d\prime$. Le principe d'une approche par surclassement est de quantifier la proportion des préférences atomiques validant (resp. invalidant) une telle assertion de préférence. Une préférence $d \succ d\prime$ (resp. $d \succeq d\prime$) est établie entre d et $d\prime$ si les préférences atomiques concordantes (resp. et indifférentes) avec cette assertion sont majoritaires et si les préférences discordantes ne sont pas trop importantes. Dans l'article [21], nous avons proposé deux approches, une stricte et une graduelle inspirée de [98], permettant de déterminer si une préférence atomique est concordante, discordante ou indifférente vis-à-vis d'une relation de préférence $d \succ d\prime$. La figure 22 illustre la façon dont est calculé le degré de concordance, indifférence et discordance d'une préférence atomique G_i par rapport à la valeur (notée $d.A$) prise par l'objet d sur l'attribut concerné A. Les trois zones sont définies à partir de la valeur $d.A$ et dépendent d'un seuil de préférence noté q_i associé à G_i.

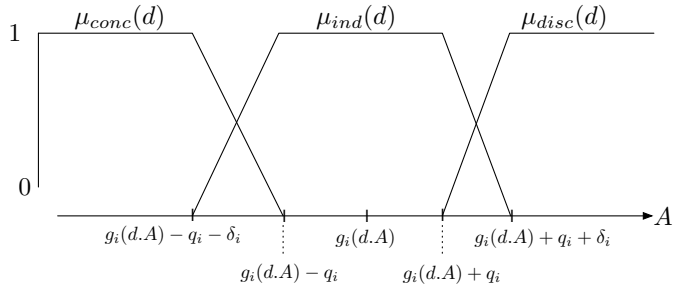

FIGURE 22 – Situations de concordance, indifférence et discordance d'une préférence atomique

Soit $\mathcal{G}^C$, $\mathcal{G}^I$, $\mathcal{G}^D$ les ensembles de préférences atomiques qui sont respectivement concordantes, indifférentes et discordantes avec l'assertion $d \succ d\prime$, le degré de surclassement de d sur $d\prime$ peut alors être calculé de la façon suivante (la version stricte du calcul du degré de surclassement ainsi que l'évaluation de l'assertion $d \succeq d\prime$ peuvent être trouvées dans l'article [21]) :

$$out(d, d\prime) = \top(conc(d, d\prime), 1 - disc(d, d\prime)),$$

où $conc(d, d\prime)$ (resp. $disc(d, d\prime)$) est le poids, prenant en compte le degré de satisfaction du SEF $\mu_{conc}(d)$ (resp. $\mu_{disc}(d)$) (cf. Fig. 22), des préférences concordantes (resp. discordantes).

Une évaluation globale de chaque tuple permet de quantifier le fait qu'un objet surclasse beaucoup d'autres objets sans être lui même fortement surclassé :

$$\mu_3(d) = \top(\mu_1(d), 1 - \mu_2(d)),$$

où $\mu_1(d) = \frac{\sum_{d\prime \in \mathcal{D} \setminus \{d\}} out(d, d\prime)}{|\mathcal{D}|-1}$ et $\mu_2(d) = \frac{\sum_{d\prime \in \mathcal{D} \setminus \{d\}} out(d\prime, d)}{|\mathcal{D}|-1}$.

Nous avons ensuite proposé une syntaxe et une stratégie d'exécution pour prendre en compte dans une requête *SQL* la définition des préférences atomiques ainsi que les paramètres nécessaires au calcul des relations de surclassement (poids w et seuils de préférence q).

Afin de réduire la complexité quadratique de la comparaison paire-à-paire de chaque objet, une approche basée sur une étape préliminaire de filtrage a été proposée [105]. Cette stratégie se fonde sur une idée et des propriétés étudiées dans le cadre de l'aide multicritère à la décision [41] pour réduire la complexité initiale des méthodes ELECTRE [53].

Autres travaux autour de l'interrogation par préférences de BD

Afin de simplifier à l'utilisateur la phase de définition de son modèle de préférence, nous avons, avec P. Bosc et O. Pivert, proposé un langage d'interrogation flexible basé sur la formalisation de préférences à l'aide d'une échelle symbolique de scores : e.g. $\omega_0 = rejected$, $\omega_1 = very\,weak$, $\omega_2 = weak$, $\omega_3 = medium$, $\omega_4 = good$, $\omega_5 = very\,good$, $\omega_6 = ideal$. Une préférence consiste à associer à tout ou partie de ces scores symboliques des intervalles ou ensembles de valeurs d'un attribut concerné. Afin de proposer un langage d'interrogation de BD prenant en compte cette définition particulière des préférences [104], nous avons tout d'abord défini un ensemble d'opérateurs permettant de combiner de telles préférences (négation, conjonction, disjonction, pondération, etc.). E.g., l'évaluation du score d'un objet d vis-à-vis d'une combinaison conjonctive pondérée de préférences symboliques de la forme « $(w_1)P_1\ and\ (w_2)\,P_2\ and\ \ldots\ and\ (w_n)\,P_n$ » (les w_is sont les poids des préférences P_is) est réalisé de la façon suivante :

$$\omega(d) = \min_i \max(rv(w_i), \omega_{P_i}(d)),$$

où $rv(\omega_i)$ est la négation par inversement de l'échelle de préférence du score ω_i : $rv(\omega_i) = \omega_{k-1-i}$.

Nous avons ensuite étendu l'algèbre relationnel et les opérateurs ensemblistes pour former un langage d'interrogation dédié. L'opération de sélection des objets satisfaisant une préférence symbolique ψ est e.g. définie comme suit :

$$\sigma_\psi(r) = \{\omega/u \mid u \in support(r)\ \wedge \omega = min(\omega_r(u),\ \omega_\psi(u))\}.$$

Une définition complète de l'algèbre étendue et des stratégies d'interprétation des requêtes est disponible dans l'article [15].

Fusion d'ordre locaux

Dans l'esprit de la célèbre méthode de Borda utilisée en théorie du choix social [122], nous avons, toujours avec P. Bosc et O. Pivert, étudié une stratégie d'interrogation flexible de BD, où

les préférences sont formalisées par des ordres locaux. L'idée est de demander à l'utilisateur de classer, sur les dimensions qui l'intéressent, des valeurs par ordre de préférence. Pour chaque préférence atomique exprimée, e.g. $bleu \succ rouge \succ blanc \succ jaune$, les objets à comparer sont classés par ordre de préférence, les bleus avant les rouges et ainsi de suite. L'évaluation globale d'un objet vis-à-vis d'un ensemble de telles préférences est alors calculé par rapport aux nombres d'objets qui le précèdent ou qui le suivent dans ces différents ordres locaux. Le calcul de ce degré global de satisfaction est calculé à l'aide de l'algorithme suivant :

1 **pour chaque** objet d **faire**
2 $\quad \sigma_1 \leftarrow 0\,;\ \sigma_2 \leftarrow 0\,;$
3 $\quad$ **pour chaque** attribut A_i **faire**
4 $\quad\quad \sigma_1 = \sigma_1 +$ nb. objets avant d sur $A_i\,;$
5 $\quad\quad \sigma_2 = \sigma_2 +$ nb. objets après t sur $A_i\,;$
6 $\quad$ **fait** ;
7 $\quad \sigma(d) = \frac{1}{2} \cdot \left(\frac{\sigma_2 - \sigma_1}{p \cdot (n-1)} + 1 \right)$
8 **fait** ;
9 classement des objets par ordre décroissant de leur valeur σ associée ;

Dans le papier [20], nous avons montré que ce modèle raffine l'ordre de Pareto classiquement utilisé pour identifier l'ensemble Skyline de réponses les plus satisfaisantes sans entraîner de surcoût calculatoire. Une extension de ce modèle a également été proposée pour prendre en compte des scores attachés à chaque valeur prenant part à la définition d'une préférence.

4 Bilan sur l'enrichissement des méthodes d'accès aux données

Cette première partie du document synthétise mes contributions sur l'amélioration de la flexibilité, l'expressivité et l'intuitivité des interfaces d'accès aux données. La flexibilité est définie dans nos travaux comme l'adaptation des systèmes aux besoins et compétences des utilisateurs. Dans la continuité des travaux réalisés par les équipes BADINS, PILGRIM puis SHAMAN, j'ai travaillé sur la modélisation de connaissances subjectives par des SEFs, mais j'ai également apporté une expertise sur d'autres modèles de préférences [21, 20, 104]. La modélisation des connaissances expertes, notamment sous forme de vocabulaire, est au cœur d'un bon nombre de mes travaux visant à fournir *in fine* des méthodes personnalisées et efficaces d'accès aux données. C'est pourquoi j'ai ensuite cherché à étudier les propriétés qui permettaient de garantir qu'un vocabulaire utilisateur pouvait être exploité efficacement pour interroger/explorer/décrire des données.

Afin de constituer une réelle solution pragmatique aux enjeux de notre ère numérique, les systèmes d'accès aux données ne peuvent se contenter d'être intelligents, flexibles, expressifs et intuitifs, ils doivent également être efficaces. La prochaine partie du document est consacrée aux stratégies d'exécution de requêtes flexibles et aux méthodes efficaces d'accès aux données.

Deuxième partie

Traitement efficace de requêtes flexibles et réécriture linguistique de données

Traitement *[nom masculin]*[8]
— Ensemble des opérations réalisées par des moyens automatiques, relatif à la collecte, l'enregistrement, l'élaboration, la modification, la conservation, la destruction, l'édition de données et d'une façon générale leur exploitation.
Efficient *[adjectif]*
— Qui produit, dans de bonnes conditions et sans autre aide, l'effet attendu..

La première partie de ce document était consacrée à mes contributions relatives à l'enrichissement des méthodes d'interrogation de données. Cet enrichissement passe notamment par la prise en compte de descripteurs linguistiques subjectifs pour former des conditions floues de sélection ou bien pour guider un processus d'exploration de données. L'amélioration de l'expressivité des langages d'interrogation ne doit cependant pas se faire au détriment de leur efficacité. Lorsque j'ai eu, en 2009, l'opportunité de travailler avec l'équipe PILGRIM sur l'interrogation flexible de bases de données, je me suis donné comme objectif de prouver que des systèmes opérationnels d'interrogation floue pouvaient être conçus et développés en offrant un compromis intéressant entre expressivité et efficacité. Le passage de la théorie à la pratique est une condition *sine qua non* pour envisager des transferts de technologie vers l'industrie. Ces récents travaux m'ont notamment permis de montrer que des systèmes flexibles et intuitifs pouvaient être développés et également former des solutions pragmatiques et innovantes répondant à des besoins fonctionnels exprimés par les acteurs industriels. Le support d'acteurs industriels des BD relationnelles (Sec. 5.2) pour mener à bien ces développements et les collaborations qui en résultent avec l'entreprise SEMSOFT et la Direction Générales des Armements (DGA) témoignent de l'enjeu crucial que constitue cette thématique.

La première partie de ce document décrit les travaux réalisés autour du développement de stratégies d'exécution de requêtes floues adressées à un SGBD relationnel et graphe. La seconde partie décrit mes travaux très récents sur les enjeux sous-jacents au développement de stratégies efficaces de résumé et d'exploration de données brutes. Mes contributions sur ce dernier point ont permis d'aboutir à un système efficace d'exploration personnalisée de données massives.

5 Exécution de requêtes floues

La définition d'un langage d'interrogation floue, tel que *SQLf* ou *FUDGE*, n'est qu'une première étape vers le développement de stratégies et de systèmes d'interrogation flexible et coopérative (partie III). Les SGBD commerciaux n'étant pas pourvus de fonctionnalités d'interrogation floue, la seconde étape, tout aussi cruciale, concerne la définition de stratégies efficaces d'exécution de requêtes floues [47].

5.1 Stratégies d'implémentation

Dans la continuité des travaux initiés par Urratia et al. dans l'article [136], j'ai étudié les différentes stratégies envisageables pour doter un SGBD hôte d'un module offrant des fonctionnalités d'interrogation floue. Trois niveaux d'imbrication entre le SGBD et le module d'interrogation floue peuvent être envisagées, on parle alors de couplages fort, modéré et faible.

Comme l'illustre la figure 23 (en haut à gauche), le couplage fort consiste à modifier l'analyseur syntaxique et le moteur d'exécution de requêtes du SGBD hôte pour intégrer la reconnaissance de requêtes floues et leur traitement. Malgré le fait que les SGBD *OpenSource* permettent ce genre de modification, cette stratégie de développement est à la fois complexe et non pérenne

8. Source : TLFi : Trésor de la langue Française informatisé, `http://www.atilf.fr/tlfi`, ATILF - CNRS & Université de Lorraine.

car une évolution du SGBD pourrait remettre en cause ces fonctionnalités. Des implémentations réalisées dans le SGBD PostgreSQL nous ont permis de constater l'efficacité de cette stratégie d'implémentation, le surcoût du calcul du degré de satisfaction étant minime, mais également la difficulté de maintenir les performances initiales du SGBD pour les requêtes booléennes. À partir de ce constat, nous nous sommes focalisés sur l'étude des deux autres stratégies de couplage afin d'aboutir à un meilleur compromis entre pérennité et efficacité des développements.

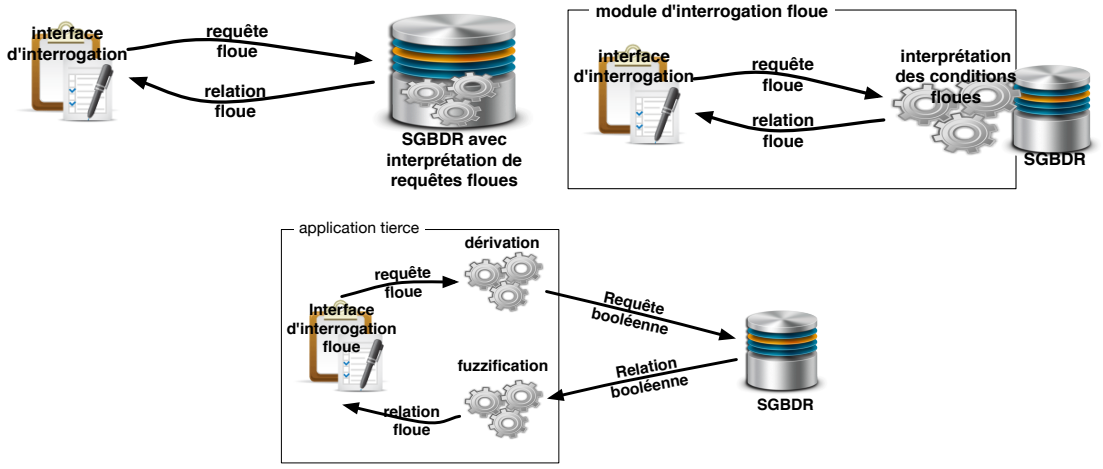

FIGURE 23 – Stratégies de couplage fort (en haut à gauche), modéré (en haut à droite) et faible (en bas)

5.2 Couplage modéré avec le SGBD hôte

La seconde stratégie d'implémentation consiste à utiliser des procédures stockées implémentées avec un langage procédural, tel que PL/SQL pour les BD relationnelles, afin d'étendre les fonctionnalités du SGBD. L'exécution de requêtes floues et la génération des résultats, sous forme de relations floues, sont alors directement réalisées par le SGBD sans que cela n'altère ses performances initiales.

Depuis la version 9.0, le SGBD PostgreSQL intègre un gestionnaire de modules permettant de greffer « à chaud » des extensions syntaxiques et des fonctionnalités à son moteur d'exécution. Pour tester leur système d'extension, l'association francophone [9] de développement du SGBD relationnel PostgreSQL a lancé un appel à projet en 2012. J'ai alors proposé de coordonner un projet visant à implémenter un module d'interrogation floue pour PostgreSQL. Avec l'aide de Thomas Girault, un ingénieur de recherche freelance que j'ai supervisé, j'ai défini une adaptation du langage SQLf conforme aux contraintes syntaxiques imposées par PostgreSQL puis des opérateurs permettant d'interpréter des conditions floues de sélection et de les combiner de manière conjonctive, disjonctive ou bien à l'aide de quantificateurs graduels.

Comme l'illustre la figure 24, ce travail a permis de doter le système PostgreSQL d'une extension de son langage d'interrogation, extension nommée PostgreSQLf, permettant notamment :

— la déclaration de descripteurs linguistiques,

select newTrapezoidalFuzzySet('ancienneté', 'élevée', 10, 13, 22, 25) ;

select newDiscreteFuzzySet('posteOccupé', 'aFortesResponsabilités', ['chef d"équipe', 'chercheur', 'doctorant', 'chef de projet', 'programmeur', 'resp. des ventes', 'vendeur', 'PDG'], [0.8, 0.7, 0.4, 0.7, 0.3, 0.5, 0.3, 1]) ;

— l'expression et l'utilisation de conditions de sélection floues,

select * **from** *employes*

where *ancienneté* ∼= 'vraiment élevée' **or** *posteOccupé* ∼= 'aFortesResponsabilités' ;

9. `http://www.postgresqlfr.org/`

select * **from** *employes*
where laPlupart(*ancienneté* ∼= 'vraiment élevée', *posteOccupé* ∼= 'aFortesResponsabilités', *dep* = 'd3') ;

```
reqflexdb=# select create_predicate('elevee', 9, 13, 22, 25);
 create_predicate
------------------

(1 row)

reqflexdb=# select *, get_mu() as mu
from employees
where anciennete ~= 'elevee';
 ide |  nom   | anciennete | dep | posteoccupe |  mu
-----+-------+------------+-----+-------------+------
 e2  | Lucas |         10 | d3  | chercheur   | 0.25
 e8  | Marie |         19 | d4  | PDG         |    1
(2 rows)
```

FIGURE 24 – Le langage d'interrogation PostgreSQLf

L'interface ReqFlex présentée dans la section 3.4 transmet ainsi directement au SGBD des requêtes floues, construites graphiquement puis traduites dans la syntaxe de PostgreSQLf, et récupère en résultat une relation floue.

5.3 Couplage faible avec le SGBD hôte

Le couplage faible consiste à externaliser les fonctionnalités d'interrogation floue à travers une interface (API) implémentée dans un langage tierce. En étant décorrélée du SGBD, cette interface doit à la fois gérer la validation syntaxique de la requête floue mais également procéder à la traduction de la requête floue dans le langage reconnu par le SGBD hôte. Cette étape, dite de dérivation, consiste à traduire une requête floue en une version booléenne aussi proche que possible, c'est-à-dire retournant uniquement les éléments pouvant satisfaire les conditions floues.

Le principe de dérivation, introduit par Olivier Pivert pour le modèle relationnel [99], a ensuite été repris pour définir, en collaboration avec Olfa Slama et Virginie Thion, un système d'interrogation floue de données sémantiques [102] et graphe [103]. Ce système, nommé *SUGAR*, est couplé de manière faible avec le moteur d'exécution de requêtes graphe *neo4J* comme l'illustre la figure 25. Dans le système *SUGAR*, l'étape de dérivation est réalisée par le module *SUGAR Transcriptor* afin d'aboutir à une requête *CYPHER* directement exécutable par *neo4J*. Les graphes retournés comme résultats de cette requête dérivée sont ensuite transmis au module *SUGAR Score Calculator* qui calcule, pour chacun d'eux, leur degré de satisfaction vis-à-vis des conditions floues exprimées dans la requête initiale. Cette stratégie offre évidemment une portabilité intéressante d'un SGBD à un autre et est la plus simple à implémenter.

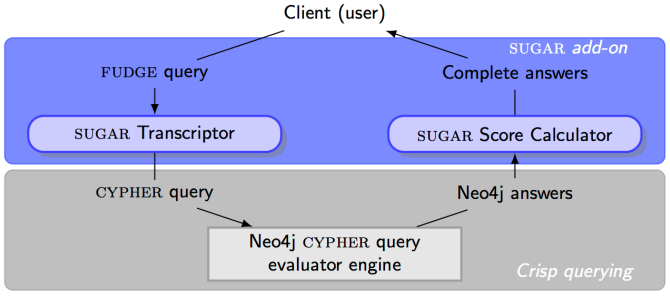

FIGURE 25 – Couplage faible entre *SUGAR* et *Neo4J*

5.4 Comparaison des stratégies d'exécution

Les différents travaux réalisés autour de l'interrogation floue de données relationnelles ou graphes m'ont ensuite permis de mener des expérimentations pour comparer l'efficacité et l'intérêt des différentes stratégies de couplage [128]. En utilisant une relation peuplée (avec 100 000 tuples) de données générées de manière pseudo-aléatoire et des requêtes de sélectivité variable, différentes stratégie d'exécution ont été comparées :

- *LCND* (Loose Coupling No Derivation) qui consiste à parcourir de manière séquentielle les tuples de la relation pour identifier ceux satisfaisant la condition de sélection,
- *LCD* (Loose Coupling Derivation) qui repose sur une dérivation de la condition floue en condition booléenne afin de calculer le degrés de satisfaction uniquement pour les tuples pertinents,
- *MCND* (Mild Coupling No Derivation) utilisant l'extension PostgreSQLf pour directement exécuter les requêtes floues,
- *MCD* (Mild Coupling Derivation) utilisant l'extension PostgreSQLf pour exécuter les requêtes floues au sein desquelles une étape de dérivation est intégrée.
  (e.g. la requête **select** * **from** *R* **where** *A* ∼= *'low';* est dérivée en **select** * **from** *(***select** * **from** *R* **where** *A* **between** *0* **and** *100) R1* **where** *R1.A* ∼= *'low';*).

Pour illustrer le comportement observé de ces quatre stratégies d'exécution, prenons trois requêtes significatives de sélectivité variable. La première requête comporte une condition floue et atomique de sélection, la seconde combine de manière conjonctive deux conditions et la troisième repose sur l'utilisation d'un quantificateur :

- Q1 : **select** * **from** *randomtable* **where** *attnum* ∼= 'low' ;
  dérivée en
  select * **from** *randomtable* **where** *attnum* **between** 0 **and** 15 ;
- Q2 : **select** * **from** *randomtable* **where** *attnum* ∼= 'low' and *atttext* ∼= 'around_10' ;
  dérivée en
  select * **from** *randomtable* **where** *attnum* **between** 0 **and** 15 **and** *atttext* **in** ('eight', 'nine', 'ten', 'eleven', 'twelve')) ;
- Q3 : **select** * **from** *randomtable* **where** most(*attnum* ∼= 'low', *atttext* ∼= 'around_10') ;
  dérivée en
  select * **from** *randomtable* **where** *attnum* **between** 0 **and** 15 **or** *atttext* **in** ('eight', 'nine', 'ten', 'eleven', 'twelve').

La figure 26 montre le temps nécessaire à chacune de ces quatre stratégies pour construire le résultat, sous forme de relation floue, de ces trois requêtes. Cet extrait des expérimentations menées montre clairement l'apport d'une étape de dérivation pour réduire le nombre de calculs de degrés de satisfaction à effectuer, ceci quelle que soit la stratégie de couplage employée. Le gain, en terme d'efficacité, d'un couplage moyen (MCND utilisant *PostgreSQLf*) par rapport à une implémentation dans un langage tierce (ici Java) décorrélée du SGBD (LCND ou LCD) n'est pas si flagrant. Ceci vient du fait que, malgré un calcul des degrés de satisfaction et une construction de la relation floue résultat dans le SGBD, *PostgreSQL* ne propose pas de mécanisme efficace pour utiliser les index disponibles sur les attributs concernés par la condition de sélection floue. Pour déterminer si un tuple satisfait une condition floue, *PostgreSQL* applique de manière séquentielle la fonction associée au prédicat flou et regarde si cette fonction retourne un nombre strictement positif, c'est-à-dire si le tuple concerné satisfait la condition floue. Les résultats les plus intéressants en terme d'efficacité sont ainsi obtenus en intégrant dans la clause **from** de la requête *PostgreSQLf* la version dérivée de la requête floue.

Les travaux réalisés autour de l'implémentation du module *PostgreSQLf* et du système *SUGAR* ont tout d'abord permis de disposer d'une base de comparaison des différentes stratégies d'implémentation. Le couplage modéré reste la stratégie la plus pertinente et pérenne pour le développement de fonctionnalités d'interrogation floue au dessus d'un SGBD existant. Cependant, en intégrant une étape de dérivation, des applications dédiées (telles que *SUGAR*) peuvent fournir également des fonctionnalités efficaces d'interrogation floue. À partir de ces résultats, deux perspectives de développement ont été définies : tout d'abord porter le langage *SQLf* sur *Oracle* et deuxièmement, rendre *ReqFlex* (Sec. 3.4) indépendant de PostgreSQLf en intégrant une étape de dérivation dans l'application pour transmettre à n'importe quel SGBD une requête *SQL* « clas-

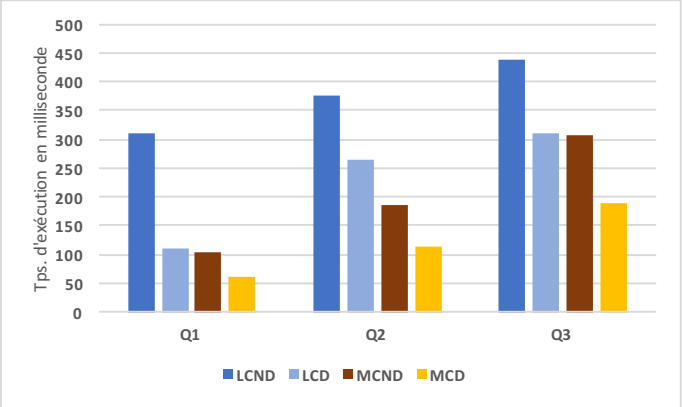

FIGURE 26 – Comparaison des temps d'exécution de requêtes floues par différentes stratégies d'implémentation

sique » et quantifier ensuite la satisfaction de chaque tuple retourné vis-à-vis des conditions floues exprimées.

6 Réécriture efficace de données

Dans la première partie de ce document (Part. I), nous avons vu que l'expressivité des langages d'interrogation de BD pouvait être améliorée par la prise en compte de conditions de sélection floue. La section précédente (Sec. 5) était, quant à elle, dédiée aux stratégies permettant de retourner, à l'émetteur d'une telle requête, les éléments les plus satisfaisants de la BD concernées. Les données qui gravitent autour de nos activités professionnelles ou personnelles ne sont pas toujours structurées selon un modèle particulier (relationnel, graphe, hiérarchique, rdf, NoSQL, etc.), mais au contraire de plus en plus stockées et transmises dans un format brut (i.e. tabulaire). Un tel jeu de données $\mathcal{D}$ contient alors la description d'un ensemble d'objets $\mathcal{D} : \{d_1, d_2, ..., d_m\}$ chacun défini par les valeurs prises sur un ensemble $\mathcal{A}$ d'attributs ($\mathcal{A} : \{A_1, A_2, ..., A_n\}$) de nature numérique ou catégorielle. Un des enjeux cruciaux auxquels les acteurs universitaires et industriels en gestion de données doivent répondre concerne la construction de méthodes d'analyse et d'exploration de données brutes.

La communauté du *soft computing* a évidemment un rôle crucial à jouer sur cette problématique en proposant des stratégies intuitives de manipulation de données brutes et d'extraction de connaissances interprétables. Une des réponses apportées par le *soft computing* au problème de la gestion de grandes quantités de données concerne la génération de résumés linguistiques [7]. Le dénominateur commun de ces approches de résumé linguistique consiste en la génération d'un ensemble de phrases décrivant les tendances observées dans les données. Ces phrases sont structurées selon un protoforme syntaxique incluant un quantificateur et des descripteurs linguistiques provenant d'un vocabulaire utilisateur, e.g. *la plupart des jeunes employés ont un salaire moyen* où *la plupart* décrit linguistiquement une proportion et les termes *jeunes* et *moyen* sont des descripteurs linguistiques issus du vocabulaire utilisateur.

Dans cette section, je présente une nouvelle approche visant également à fournir une vue synthétique d'un jeu de données, vue également basée sur des termes linguistiques issus d'un vocabulaire utilisateur. Dans un contexte applicatif particulier, celui du *Business Intelligence* (BI), les experts métier s'attendent à disposer de représentations graphiques des données. Outre le fait de répondre à un besoin pragmatique actuel, les travaux présentés dans cette section ont également été motivés par une demande de l'entreprise Semsoft (Rennes, France) avec laquelle SHAMAN collabore de longue date. Semsoft a développé une architecture logicielle de médiation permettant à une entreprise de centraliser et d'unifier l'ensemble de ses jeux de données. Cependant, l'intégration d'un nouveau jeu de données dans cette architecture logicielle n'est pas aisée et repose sur un travail d'ingénie-

rie informatique. Les experts métier, souvent non-informaticiens, souhaitent disposer d'outils leur permettant d'analyser le contenu et la structure d'un jeu de données avant de commander son intégration dans le système d'information. De tels outils permettant de rendre autonomes les experts dans l'analyse de données métiers sont désignés comme des solutions de BI ou plus précisément dans notre cas *Self Service BI* (SSBI) et visent à fournir des vues graphiques, nommés tableaux de bord, contenant des représentations des données et des connaissances extraites. Le SSBI est un domaine abordé principalement par les grands groupes industriels de traitement des données (e.g. Amazon avec QuickSight, Pentaho Agile BI, Microsoft avec PowerBi, etc.) mais suscite un intérêt croissant chez les acteurs scientifiques [52, 61, 149, 31, 29]. Dans le cadre d'un contrat de recherche externe (2013) puis de l'encadrement d'une thèse CIFRE (étudiant Toan Ngoc Duong, 2016-2019) financés par Semsoft, j'ai travaillé à la définition d'une approche de SSBI basée sur les techniques issues du *soft computing* et notamment la représentation linguistique et personnalisée des données. Ces travaux constituent une application parfaite du paradigme *Computing with Words* défini par L.A. Zadeh dans [147]. Cette section est consacrée à la représentation linguistique et personnalisée des données brutes et sera complétée dans la section 10 par des méthodes d'extraction de connaissances à partir de ces représentations synthétiques des données.

Qu'il s'agisse de la génération de résumés linguistiques ou bien le développement d'approches coopératives de manipulation de données (comme nous le verrons dans la partie III), une étape centrale et commune à ces différents travaux concerne la réécriture de données à l'aide de termes linguistiques issus du vocabulaire utilisateur. Cette réécriture permet de « passer » les données de l'espace numérique/catégoriel dans lequel elles sont initialement définies à un espace linguistique personnalisé dans lequel elles seront décrites et manipulées [147].

Dans cette section, je présente les travaux réalisés sur cette étape de traduction linguistique des données et fournis une réponse possible aux questions suivantes :

— Comment réécrire efficacement de grandes quantités de données (Sec. 6.1) ?
— Comment représenter formellement la distribution des données selon les termes du vocabulaire (Sec. 6.2) ?
— Quelles stratégies de stockage et mécanismes d'indexation sont les plus adaptés pour permettre un retour de l'espace linguistique de description personnalisée à l'espace initial numérique/catégoriel de définition des données (Sec. 6.3) ?

6.1 Réécriture linguistique de grandes quantités de données

Le terme *Computing with Words* désigne le fait que les données manipulées, initialement définies dans un espace numérique/catégoriel, sont transposées dans un espace linguistique en s'appuyant sur des descripteurs personnalisés (Sec.1.1). En considérant un jeu de données brutes $\mathcal{D}$ et un vocabulaire utilisateur $\mathcal{V}$ matérialisé par une discrétisation des domaines de définition des attributs $\mathcal{A}$, cette sous-section décrit les aspects algorithmiques du passage de l'espace numérique/catégoriel à l'espace linguistique personnalisé. L'objectif de cette étape de « traduction » est de produire à la fois un vecteur de réécriture, noté $RV_d^{\mathcal{V}}$ (Sec. 1.3), pour chaque objet d du jeu de données $\mathcal{D}$ et également un vecteur global pour l'ensemble du jeu de données, noté $RV_{\mathcal{D}}^{\mathcal{V}}$ (Sec. 1.3), et représentant la distribution des termes linguistiques par rapport aux données. Dans un article (soumis pour la session spéciale de Fuzzy Sets and Systems sur la gestion de données massives à l'aide de techniques issues du soft computing), nous (en collaboration avec R.R. Yager, P. Nerzic et O. Pivert) avons proposé deux algorithmes pour construire le vecteur de réécriture $RV_{\mathcal{D}}^{\mathcal{V}}$. Le premier, très simple, utilise une stratégie séquentielle de réécriture de chaque objet pour chaque dimension sur laquelle un vocabulaire utilisateur est défini (Alg. 1). Ce processus de complexité initiale linéaire en fonction des données et du vocabulaire (au pire cas $|\mathcal{D}| \times |\mathcal{V}|$ calculs de degrés de satisfaction) peut être réduit en complexité sous-linéaire en prenant aléatoirement un sous-ensemble des données pour obtenir un premier aperçu du jeu de données.

Une stratégie séquentielle centralisée de réécriture des données n'est envisageable que si les données ne sont pas trop volumineuses et peuvent être stockées et traitées par une seule unité de calcul. Afin de pouvoir réécrire de grandes quantités de données, une stratégie de calcul parallélisée et distribuée sur une architecture de type map-reduce [35] a également été définie.

Données : Jeu de données $\mathcal{D}$, Vocabulaire $\mathcal{V}$
Résultat : Vecteur de réécriture $RV_{\mathcal{D}}^{\mathcal{V}}$

$RV_{\mathcal{D}}^{\mathcal{V}} \leftarrow \emptyset$;
pour tous les $x \in \mathcal{D}$ **faire**
  $RV_x^{\mathcal{V}} \leftarrow \emptyset$
  pour tous les $v \in \mathcal{V}$ **faire**
    calculer $\mu_v(x)$
    ajouter $\mu_v(x)$ dans $RV_x^{\mathcal{V}}$
    si $v \notin RV_{\mathcal{D}}^{\mathcal{V}}$ **alors**
      $RV_{\mathcal{D}}^{\mathcal{V}}[v] = 0.0$
    fin
    $RV_{\mathcal{D}}^{\mathcal{V}}[v] = RV_{\mathcal{D}}^{\mathcal{V}}[v] + \mu_v(x)$
  fin
  $stocker(x, RV_x^{\mathcal{V}})$
fin
pour tous les $v \in \mathcal{V}$ **faire**
  $RV_{\mathcal{D}}^{\mathcal{V}}[v] = RV_{\mathcal{D}}^{\mathcal{V}}[v]/|\mathcal{D}|$
fin
retourner $RV_{\mathcal{D}}^{\mathcal{V}}$

Algorithme 1 : Réécriture linguistique séquentielle d'un jeu de données

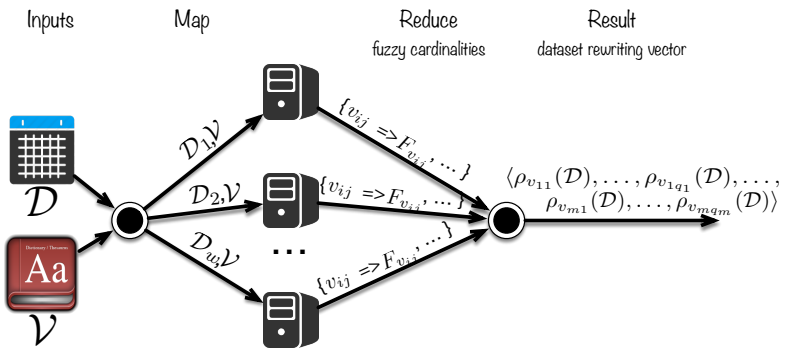

FIGURE 27 – Distribution du processus de réécriture des données

Comme l'illustre la figure 27, le stockage et le calcul des réécritures est distribuée dans une grappe de w unités de calcul, où chaque unité est en charge de la réécriture d'un sous-ensemble $s \subseteq \mathcal{D}$. La stratégie de calcul distribué repose sur trois étapes :

1. la réécriture de chaque objet $d \in s$:
$$\{RV_d^{\mathcal{V}}\} = map(lambda\, d : rewrite(d, \mathcal{V}), s),$$
où $rewrite(d \in s, \mathcal{V})$ est une fonction $(d \in s \times \mathcal{V} \rightarrow [0,1]^{|\mathcal{V}|})$ qui retourne le vecteur $(RV_d^{\mathcal{V}})$ de d par rapport à $\mathcal{V}$,
2. l'agrégation des vecteurs de réécriture des objets pour obtenir le vecteur non normalisé de s
$$RV_s'^{\mathcal{V}} = reduce(lambda\, d_i, d_j : map(lambda\, \mu, \mu' : \mu + \mu', d_i, d_j), \{RV_x^{\mathcal{V}}\}),$$
3. la normalisation finale du vecteur de réécriture de $\mathcal{D}$.
$$RV_{\mathcal{D}}^{\mathcal{V}} = map(lambda\, \rho_i : \rho_i/|\mathcal{D}|, RV_{\mathcal{D}}'^{\mathcal{V}}).$$

Seules l'agrégation des vecteurs de réécriture des sous-ensembles ($s \subseteq \mathcal{D}$) et la normalisation finale sont centralisées sur le nœud maître de la grappe, l'étape 2 étant en effet partiellement effectuée de manière distribuée à l'aide de l'instruction *combiner* des environnements map-reduce. Le nombre de dimensions sur lesquelles les données sont décrites étant généralement de taille réduite (< 100) et de taille négligeable par rapport à celle des données, la stratégie distribuée a une complexité linéaire en fonction de $|\mathcal{D}|$, et effectue donc $\frac{|\mathcal{D}|}{w}$ opérations fondamentales (la réécriture d'une valeur en fonction du vocabulaire dans notre cas). Des expérimentations menées sur des données réelles (descriptions des 127 millions de vols domestiques aux Etats-Unis de 1987 à 2008 [10] stockées sur des volumes *HDFS* et un vocabulaire composé de 75 termes définis sur 16 attributs) et une grappe de calcul *Hadoop* [11] montrent cette efficacité et l'intérêt d'une approche distribuée. Les figures 28, 29 et 42 montrent respectivement le dépendance linéaire de la réécriture par rapport aux données, au vocabulaire (pour 10 millions de vols) et au nombre d'unités dans la grappe de calcul (également pour 10 millions de vols).

L'objectif de ces expérimentations n'était pas d'éprouver les capacités de notre modeste grappe de calcul, mais bien d'étudier le comportement du processus de réécriture en fonction de la taille des données, du vocabulaire et des ressources de calcul disponibles. Lors de ces expérimentations, nous avons pu quantifier que près de 98% du processus de réécriture était complètement distribué sur les w unités de calcul. Ainsi, sur la base des mesures relevés pour résumer 10 millions de vols sur 75 termes en utilisant notre grappe de 4 unités, le tableau 5 avance les projections de temps de traitement en utilisant plus de ressources distribuées.

TABLE 5 – Résumé de 10 millions de vols en fonction de la taille du cluster (w)

w	4	8	16	32
temps en s.	27	14	7	4

6.2 Représentation de la distribution des données à l'aide de cardinalités floues

Les vecteurs de réécriture, soit d'objets soit de jeux de données, représentent la distribution des valeurs sur un ensemble de termes. Cette connaissance n'est pas toujours suffisante car il est parfois nécessaire de disposer d'informations précises sur la distribution des données pour différentes combinaisons de propriétés. Cette section décrit un travail réalisé avec P. Bosc, O. Pivert et A. Hadjali en 2011 sur la représentation de la distribution de données vis-à-vis d'un vocabulaire utilisateur composé de descripteurs linguistiques (cf. Sec. 1.2).

10. http ://stat-computing.org/dataexpo/2009/
11. 5 machines dont 4 esclaves Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz-4 cores with 6GB of RAM

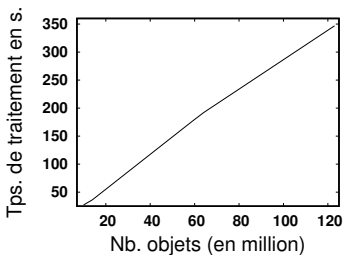

FIGURE 28 – Réécriture selon la taille de $\mathcal{D}$ ($|\mathcal{V}| = 75\,termes$)

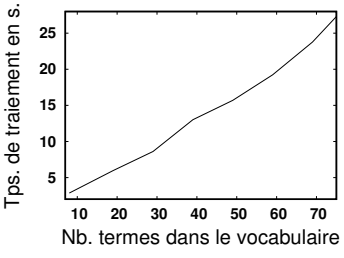

FIGURE 29 – Réécriture de 10 millions de vols en fonction de $|\mathcal{V}|$

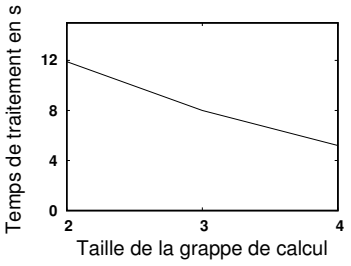

FIGURE 30 – Réécriture en fonction du nombre d'unités de traitement

Cardinalité floue

Soit P un prédicat flou formé par la conjonction de termes issus d'un vocabulaire $\mathcal{V}$ (e.g. âge est 'jeune' et salaire est 'élevé' et position est 'permanente'), et $\mathcal{D}$ un ensemble de données. Les cardinalités floues [43, 10] offrent un formalisme adéquat pour représenter la distribution des objets de $\mathcal{D}$ vis-à-vis du prédicat P. La cardinalité floue associée à P est alors formalisée comme une distribution de possibilités :

$$\mathcal{F}_P = 1/0 + \ldots 1/(n-1) + 1/n + \lambda_1/(n+1) + \ldots + \lambda_k/(n+k) + 0/(n+k+1) + \ldots,$$

où $1 > \lambda_1 \geq ... \geq \lambda_k > \lambda_{k+1} = 0$, indiquant que $n+k$ objets satisfont possiblement le prédicat P à un degré supérieur ou égal à λ_k. Nous utilisons une représentation plus compacte basée sur une échelle ordonnée de degrés de satisfaction $1 = \alpha_1 > \alpha_2 > ... > \alpha_f > 0$ définie comme suit :

$$\mathcal{F}_P = 1/c_1 + \alpha_2/c_2 + ... + \alpha_f/c_f,$$

où $c_i, i = 1..f$ est le nombre d'objets de $\mathcal{D}$ qui satisfont P à un degré au moins égal à α_i.
Lorsque P n'est pas atomique et concerne une conjonction de propriétés (e.g. $\mathcal{F}_{P^a \wedge P^b \wedge ... \wedge P^q}$), la norme min (e.g. $min(\mu_{P^a}(d), \mu_{P^b}(d), ..., \mu_{P^q}(d))$) est dans notre cas appliquée sur les degrés de satisfaction individuels pour obtenir la satisfaction globale de l'objet d sur P.

Algorithme de calcul des cardinalités floues

Afin de disposer d'une représentation complète des données selon le vocabulaire utilisateur, il faut calculer la cardinalité floue de chaque partie de l'ensemble des termes du vocabulaire. Ceci revient à construire un treillis des conjonctions possibles de termes (Fig. 31), et à associer une cardinalité floue à chaque nœud de cette structure. Le calcul des cardinalités repose sur un parcours du treillis à l'aide d'un algorithme à la *apriori*, où pour chaque objet son vecteur de réécriture ($RV_d^{\mathcal{V}}$) est construit puis utilisé pour mettre à jour les cardinalités.

L'algorithme de construction des cardinalités floues est évidemment de complexité linéaire par rapport aux données mais dépend de manière exponentielle du nombre de termes dans le vocabulaire. Pour chaque objet du jeu de données à traiter, l'ensemble du treillis n'est pas parcouru puisque deux critères peuvent être utilisé pour arrêter le parcours d'une branche de treillis :

- si un objet ne satisfait pas une conjonction de termes, disons P, alors il ne satisfera pas non plus toutes les combinaisons de termes incluant P,
- du fait des contraintes structurelles imposées sur le vocabulaire utilisateur (Sec. 1.2) un objet ne peut satisfaire plus de deux termes d'une même partition. Dès que la somme des degrés de satisfaction d'un terme est égale à un, il n'est plus nécessaire d'envisager les conjonctions de termes qui incluent les autres termes de la dimension concernée.

Le processus de calcul des cardinalités floues repose sur un traitement séquentiel des objets et un comptage des objets satisfaisant les combinaisons de termes selon différents degrés. Cette procédure est donc complètement distribuable dans une architecture *map/reduce* e.g. où chaque unité de traitement retourne des associations clé/valeur, la clé étant l'association d'une conjonction de termes et d'un degré de satisfaction (α_i) et la valeur le nombre d'objets du sous-ensemble traité

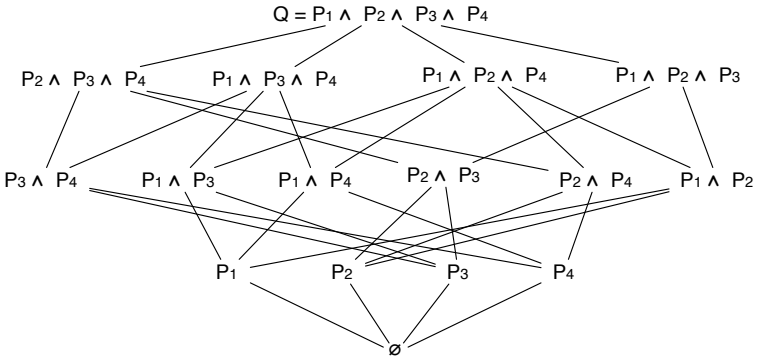

FIGURE 31 – Treillis des combinaisons conjonctives possibles des termes de l'ensemble $\{P_1, P_2, P_3\}$

satisfaisant la conjonction au degré α_i. L'opération finale de *reduce* consiste alors à agréger ces comptages.

Expérimentations et observations sur le calcul de cardinalités floues

Ce travail assez technique sur l'utilisation d'un formalisme existant, les cardinalités floues, pour représenter la distribution de données selon un vocabulaire utilisateur, nous a surtout permis de mener des réflexions sur la réécriture linguistique de données à partir d'un vocabulaire utilisateur. La projection de valeurs sur un ensemble flou F introduit une relation d'indistinguabilité entre ces valeurs puisque, d'un point de vue de la satisfaction de F, toutes les valeurs appartenant à la même α-coupe ($F^{\alpha}_{\mathcal{D}} = \{d \in \mathcal{D}, \mu_F(d) \geq \alpha\}$) sont indistinguables. De plus, dans de nombreux contextes applicatifs, des corrélations existent entre les attributs utilisés pour décrire les objets. Ces deux propriétés, dont la présence et la fréquence dépendent du contexte applicatif, ont un impact important sur le calcul des cardinalités floues. En effet, bien que le nombre théorique de combinaisons possibles de descripteurs linguistiques issus du vocabulaire utilisateur soit de $2^{|\mathcal{V}|}$, le nombre réel de combinaisons observées, et donc à calculer et stocker, est nettement inférieur. En effet, en plus des propriétés structurelles des partitions du vocabulaire qui réduisent déjà le nombre de combinaisons possibles, la relation d'indistinguabilité entre les valeurs d'une même α-coupe factorise les réécritures et les corrélations entre attributs font que certaines combinaisons de termes ne sont pas observables.

Nous avons mené des expérimentations sur deux jeux de données réelles. Le premier décrit 46 089 voitures d'occasion sur 10 attributs {*age*, *km*, *prix*, *marque*, *longueur*, *hauteur*, *nbSieges*, *accéleration*, *consommation*, *emissionCO2*}. Un vocabulaire de sens commun composé de 59 descripteurs linguistiques a été défini sur ces 10 attributs. Nous avons ensuite appliqué un algorithme (version séquentielle non distribuée) de calcul des cardinalités floues sur ces données. La figure 32 (gauche) montre bien la dépendance linéaire du processus vis-à-vis des données et la dépendance exponentielle par rapport à la taille du vocabulaire, et donc également par rapport au nombre de dimensions prises en compte (Fig. 32 à droite). L'observation la plus intéressante est donnée par la figure 33, qui montre que, quelque soit la taille du jeu de donnée, les corrélations entre attributs font que le nombre de combinaisons observables de termes converge très rapidement. Ce phénomène a également été observé et quantifié sur le jeu de données, plus volumineux, des vols d'avions (Sec. 6.1). Pour décrire la distribution de sept millions de vols sur un vocabulaire composé de 75 termes définis sur 16 attributs, 700 000 combinaisons de termes, et donc cardinalités floues, sont nécessaires, ce qui correspond à une factorisation de $\frac{1}{10}$. Le nombre de combinaisons observées sur la totalité du jeu de données, soit un peu plus de 127 millions de vols, est d'environ 6 millions, soit une factorisation de $\frac{1}{17}$, ce qui est bien loin des $2^{75} \approx 3,8 \times 10^{22}$.

Une telle représentation des données par cardinalités floues peut être construite *a priori* et aisément maintenue de manière incrémentale suite à des évolutions des données.

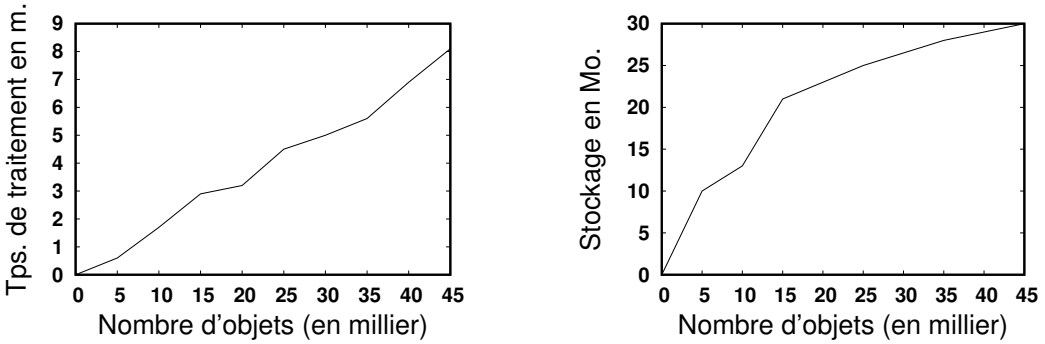

FIGURE 32 – Temps de calcul des cardinalités (gauche) et espace de stockage utilisé (droite) vs. taille du jeu de données

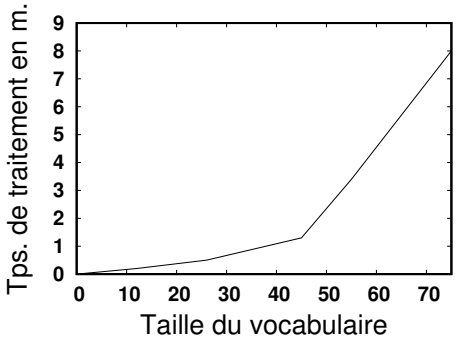

FIGURE 33 – Temps de calcul des cardinalités vs. nombre de descripteurs linguistiques

L'intérêt des cardinalités floues et leur capacité à représenter précisément la distribution de données vis-à-vis de termes linguistiques imprécis ont été démontrés pour de nombreuses tâches allant de l'extraction de règles d'association floue [18] à l'évaluation de résumés linguistiques [36] en passant par l'évaluation de la pertinence d'une source de données [101, 106]. Ce travail a été pour nous un préalable à la définition de fonctionnalités de réponses coopératives [129] qui seront décrites dans la partie suivante (Part. III).

6.3 Stockage des réécritures

Réécrire efficacement de grandes quantités de données selon un vocabulaire utilisateur est évidemment un préalable au développement de fonctionnalités d'exploration personnalisée comme celles décrites dans la section 3.6. Pour que cette exploration soit fluide, et afin de permettre à l'utilisateur de visualiser les données satisfaisant une condition atomique ou conjonctive composée de termes issus de son vocabulaire, il faut également définir des mécanismes d'indexation des réécritures. Dans un contexte d'interrogation flexible (Sec. 3) les conditions de sélection floue ne sont pas connues à l'avance, c'est pourquoi les mécanismes de dérivation sont indispensables pour identifier rapidement les valeurs, puis les objets, à sélectionner et ainsi utiliser les index « classiques » fournis par les SGBD. Dans un système d'exploration personnalisée de données où le vocabulaire utilisateur est défini *a priori* et est relativement stable, il devient alors possible de définir des stratégies de stockage et d'indexation des vecteurs de réécritures associés aux tuples du jeu de données. En complément de l'algorithme distribué de réécriture d'un jeu de données, nous avons envisagé trois stratégies possibles de stockage et d'indexation des vecteurs de réécriture pour des objets stockés dans une table *HDFS*. Nous disposons donc pour chaque objet d'un identifiant de ligne dans la table *HDFS* noté *rowid*.

Stratégie clé/valeur (*CV*)

La première stratégie exploite les capacités des systèmes NoSQL à gérer de grandes quantités d'associations clé/valeur, et consiste à associer à chaque terme du vocabulaire utilisateur (référencé par un identfiant noté *termid*), formant la clé, la liste des *rowid*s des objets satisfaisant ce terme, formant la valeur associée à la clé. Cette liste potentiellement très longue doit être subdivisée (méthode des *buckets* en BD) et la clé dupliquée (cf. Tab. 6).

TABLE 6 – Stratégies de stockage *CV*

clés (termes)	**valeurs (identifiants)**
$medium\_horsePower\_1$	$\langle rowID_1, \ldots, rowID_i, \rangle$
$medium\_horsePower\_2$	$\langle rowID_j, \ldots, rowID_k, \rangle$
...	...

Stratégie bitmap (*BM*)

La seconde stratégie repose sur la construction d'un vecteur binaire de réécriture (i.e. bitmap) pour chaque objet d indiquant, pour chaque terme, si d le satisfait (valeur à 1) ou non (valeur à 0). Une colonne supplémentaire est ajoutée à la table des objets afin de stocker son bitmap (Tab. 7). Un bitmap contient autant de bits qu'il y a de termes dans le vocabulaire. Cependant, du fait des propriétés des partitions qui forment le vocabulaire, le bitmap ne contient qu'au maximum $2m$ bits à 1. Des stratégies intéressantes de compression peuvent ainsi être utilisées [6] pour lui la place prise par le stockage des vecteurs de réécriture sur la ressource de stockage.

TABLE 7 – Stratégies de stockage *CV-BM*

rowID	A_1	...	A_m	**bitmap**
t_1	$t_1.A_1$	...	$t_1.A_m$	$\langle 011010001100010 \rangle$
t_2	$t_2.A_1$	...	$t_2.A_m$	$\langle 110010001000010 \rangle$
...	...	...	...	...

Stratégie hybride (*CV-BM*)

Nous avons vu dans la section 6.2 que la description linguistique de données à l'aide de SEFs induisait une relation d'indistinguabilité entre les objets satisfaisant un terme à un même degré. Nous avons également vu que l'existence de corrélations entre attributs dans la plupart des contextes applicatifs réduisait le nombre de combinaisons de réécritures possibles. La troisième stratégie de stockage et d'indexation des vecteurs de réécriture exploite ces observations. Cette stratégie repose sur une redéfinition des *rowid*s des objets de la table de sorte que les objets ayant une réécriture linguistique commune (même vecteur bitmap de réécriture) aient des *rowid*s consécutifs. Le stockage et l'indexation des réécritures sont réalisés dans une table annexe de type clé/valeur, où les clés sont les différents bitmaps observés et à chacun d'eux est associé l'intervalle des identifiants d'objets qui se réécrivent de cette façon (Tab. 8).

TABLE 8 – Stratégies de stockage : *CV-BM*

bitmap	**intervalles d'identifiants**
$\langle 011010001100010 \rangle$	$[1, i]$
$\langle 110010001000010 \rangle$	$[i_{+1}, j]$
$\langle 110010001000011 \rangle$	$[j_{+1}, k]$
...	...

6.4 Comparaison des stratégies de stockage des réécritures

Les stratégies de stockage des réécritures linguistiques ont été implémentées puis évaluées selon trois critères, i) l'espace utilisé pour le stockage des vecteurs, ii) l'efficacité de sélection d'un sous-ensemble de tuples satisfaisant un terme ou une conjonction de termes, et iii) leur capacité à s'adapter aux modifications du vocabulaire ou des données.

Pour mener ces expérimentations, nous avons utilisé le cluster *Hadoop* composé de 5 unités de stockage décrit dans la section 6.1 et le même jeu de données de 11Go décrivant des vols d'avion aux Etats-Unis. Les données, stockées dans une base HBASE sont décrites linguistiquement par un vocabulaire défini sur 16 attributs (*DayOfWeek, DepTime, AirTime, ArrDelay, DepDelay, Distance, Month, DayOfMonth, TaxiIn, TaxiOut, CarrierDelay, WeatherDelay, SecurityDelay, LateAircraftDelay, Origin, Dest*) avec un total de 75 termes.

Espace de stockage

Pour la première stratégie *CV*, si quatre octets sont utilisés pour représenter les identifiants d'objets et 1 octet pour chaque terme du vocabulaire, alors le stockage des réécritures avec la stratégie *CV* utilise $|\mathcal{D}| \times 4 \times 2 \times |\mathcal{V}| + |\mathcal{V}|$. Quant à la stratégie *BM*, l'association d'un bitmap à chaque objet conduit à une augmentation de l'espace de stockage utilisé de $\frac{|\mathcal{V}|}{8} \times |\mathcal{D}|$ octets. Finalement, pour la stratégie hybride *CV/BM*, si H est le nombre de vecteurs distincts de réécritures d'objets, alors l'espace de stockage utilisé par la stratégie *CV-BM* est de $H \times \frac{|V|}{8} \times 2 \times 4$.

Le tableau 9 montre que l'approche *CV-BM* est bien la plus efficace en terme d'espace mémoire utilisé pour stocker les réécritures.

Table 9 – Comparaison de l'espace mémoire utilisé par les stratégies *CV*, *BM* et *CV-BM*

Strategie	*CV*	*BM*	*CV-BM*
Espace occupé en Mo	9000	300	67

Sélection d'un sous-ensemble d'objets

Associée à une fonction de hachage calculée sur les identifiants de termes linguistiques, la stratégie *CV* permet d'identifier en temps constant la liste des identifiants d'objets satisfaisant un terme. Cependant, en cas de conjonction de propriétés le résultat de l'intersection est à construire en mémoire, ce qui est rédhibitoire. Pour la stratégie *BM*, la sélection des identifiants d'objets satisfaisant une propriété ou une conjonction de propriétés repose sur un parcours séquentiel, pouvant être distribué, des bitmaps sur lesquels des opérations binaires (conjonction avec un masque) sont appliquées efficacement. La sélection à partir de propriétés (atomiques ou conjonctives) pour la stratégie *CV-BM* repose également sur un parcours séquentiel distribuable durant lequel des opérations binaires sont effectuées. Du fait du nombre potentiellement réduit de vecteurs de réécritures stockés (uniquement les vecteurs distincts), ce parcours est au moins aussi rapide que celui de la stratégie *BM*.

Le tableau 10 montre le temps moyen observé pour 30 opérations de sélection conjonctive ayant des degrés de sélectivité variables.

Table 10 – Comparaison de l'efficacité des stratégies *CV*, *BM* et *CV-BM* pour sélectionner un sous-ensemble d'objets

Strategie	*CV*	*BM*	*CV-BM*
Tps. d'exécution en s.	< 1	60	20

Adaptabilité aux évolutions des données et du vocabulaire

Le principal avantage de la stratégie *CV* est sa capacité à faire évoluer les réécritures stockées en fonction de modifications effectuées sur les données et dans certaines mesures sur le vocabulaire. La stratégie *BM* supporte également facilement les modifications des objets car il suffit de reconstruire les bitmaps concernés, mais pas forcément les évolutions du vocabulaire. Quand à la stratégie *CV-BM*, toute modification des données ou du vocabulaire implique une redéfinition des identifiants des objets (i.e. *rowid*) et une reconstruction complète de la table stockant les réécritures distinctes.

Bilan sur les stratégies de stockage des réécritures

Il ressort de ces expérimentations que la stratégie hybride *CV-BM* est très intéressante de par la concision de sa structure de stockage mais également la rapidité avec laquelle elle permet de sélectionner un sous-ensemble des objets satisfaisant un terme. Cette stratégie ad-hoc est donc à privilégier lorsque les données et le vocabulaire ne varient pas.

7 Bilan sur l'exécution de requêtes floues et la réécriture linguistique de données

Cette seconde partie du document peut paraître assez technique mais adresse des problématiques cruciales pour la valorisation de stratégies d'interrogation floue ainsi que pour les approches personnalisées d'exploration de données.

Les travaux décrits dans la section 5 sur l'implémentation de stratégies d'interrogation floue ont constitué une étape importante et fournissent des résultats qui ont eu une forte influence sur la visibilité de l'équipe SHAMAN et mes perspectives de recherche (Sec. 13). L'interface *ReqFlex* couplée avec l'extension *PostgreSQLf* ont constitué une démonstration de la faisabilité et de l'intérêt de la logique floue pour l'enrichissement des méthodes d'interrogation de données [128]. Ce passage réussi de la théorie à la pratique a constitué également un argument influent pour établir des collaborations avec des acteurs professionnels des données tels que Semsoft, DCBrain et l'association francophone PostgreSQL. D'un point de vue personnel, ce travail a été également très enrichissant car il s'agit du résultat d'un projet au cours duquel j'ai pu superviser un ingénieur de recherche et trois étudiants stagiaires. En capitalisant sur l'héritage scientifique des équipes BADINS puis PILGRIM et sur ces derniers résultats techniques et expérimentaux, l'équipe SHAMAN constitue désormais un protagoniste majeur de l'interrogation floue de données structurées. Un projet de maturation logicielle est en cours de montage avec la Société Accélératrice de Transfert Technologique (SATT) afin de compléter et valoriser l'interface *ReqFlex*. L'idée est d'intégrer les stratégies de dérivation dans *ReqFlex* et ainsi s'affranchir de *PostgreSQLf* pour offrir des capacités d'interrogation floue de n'importe quelle source de données.

Les travaux récents sur la réécriture linguistique de données massives apportent une solution novatrice et pragmatique à un enjeu actuel auquel les acteurs scientifiques de gestion des données et connaissances doivent répondre : fournir efficacement des représentations interprétables des données assorties d'explications [3]. Grâce au processus linéaire de réécriture linguistique personnalisée décrit plus haut, nous pouvons désormais envisager la construction de stratégies d'extraction de connaissances à partir de résumés des données et ainsi apporter des fonctionnalités intéressantes d'exploration et d'analyse de données (Sec. 10.3).

Troisième partie
Enrichir la restitution des résultats

Enrichir *[verbe transitif]*[12]
— En ajoutant un élément nouveau qui augmente la valeur de l'ensemble.

Introduction aux réponses coopératives

Le terme *réponses coopératives* est apparu au début des années 80 dans les systèmes de question-réponse en langue naturelle [74]. Au sens large, on appelle stratégie de réponse coopérative tout mécanisme visant à enrichir l'informativité des résultats retournés par un système suite à l'exécution d'une requête utilisateur. Pour la communauté des BDs, un système d'interrogation doté d'un comportement coopératif doit :

- expliquer à l'utilisateur, le cas échéant, les raisons pour lesquelles aucune réponse ne peut être fournie à sa requête et proposer une version corrigée de la requête initiale,
- aider l'utilisateur à comprendre un ensemble de réponses,
- guider l'utilisateur pour identifier, dans un ensemble potentiellement pléthorique de résultats, les objets les plus pertinents/intéressants,
- assister l'utilisateur lors de la construction de sa requête. En ce sens les approches décrites dans la section 3.5 constituent des approches coopératives d'aide à la construction de requête (SQL et SPARQL) à l'aide de mots-clés.

Ces problématiques ont tout d'abord été étudiées dans le cadre « classique » de l'interrogation booléenne de BD relationnelles [33, 59, 94, 115, 32]. Dans la continuité des travaux initiés par l'équipe PILGRIM au milieu des années 2000, je me suis pour ma part attaché à les transposer et à les résoudre dans le contexte d'interrogation flexible de BD.

8 Expliquer les réponses à une requête

En réponse à une requête utilisateur, un SGBD retourne l'ensemble des objets (i.e. plus précisément des tuples qui forment une relation) satisfaisant les conditions exprimées. Lorsque la clause de sélection de la requête comporte au moins un prédicat flou, les objets retournés sont associés à un degré de satisfaction (i.e. passage d'une relation à une relation floue) permettant à l'utilisateur d'identifier plus rapidement les objets les plus intéressants. Une fois le résultat de la requête généré, une stratégie de réponse coopérative peut être appliquée pour fournir davantage d'information et ainsi permettre à l'utilisateur de mieux comprendre le contenu de la relation résultat de sa requête.

Dans cette section, deux stratégies coopératives sont décrites. La première (Sec. 8.1) concerne la génération d'explications permettant de comprendre la disparité des objets présents dans une relation résultat. Cette approche a été définie dans le cadre du master recherche puis de la thèse d'Aurélien Moreau que j'ai co-encadrés avec O. Pivert. La seconde décrit une contribution importante réalisée avec O. Pivert et A. Hadjali sur l'identification des raisons originelles de la vacuité d'un ensemble résultat (Sec. 8.3).

8.1 Caractérisation des réponses

Un volet du projet *ODIN* financé par le dispositif *RAPID-DGA* (Annexe A) concerne l'enrichissement des réponses retournés par un SGBD, sujet que nous avons notamment étudié à travers la thèse d'A. Moreau (Annexe C). Dans les articles [89, 91], nous avons présenté une approche coopérative générant un ensemble d'explications distinctives permettant de mieux comprendre le contenu d'un ensemble, potentiellement volumineux, de réponses. Afin de rendre ces explications

12. Source : TLFi : Trésor de la langue Française informatisé, `http://www.atilf.fr/tlfi`, ATILF - CNRS & Université de Lorraine.

les plus informatives et interprétables possibles, elles sont exprimées à l'aide de descripteurs linguistiques issus d'un vocabulaire pré-défini sur le domaine concerné (Sec. 1.2).

Considérant une requête Q visant à retourner les objets d'une relation R (R pouvant être le résultat d'une jointure sur plusieurs relations) décrite sur un ensemble $\mathcal{A}$ d'attributs, nous distinguons trois sous-ensembles d'attributs de A. On désigne par $A_\pi \subset \mathcal{A}$ les attributs sur lesquels les résultats de Q sont projetés, $A_\sigma \subset \mathcal{A}$ les attributs concernés par la clause de sélection de Q et $A_\omega = \mathcal{A} \setminus \{A_\pi \bigcup A_\sigma\}$. L'approche d'explication des résultats Σ_Q de la requête Q repose sur trois étapes :

1. l'identification de la structure intrinsèque de Σ_Q en appliquant un algorithme de classification non-supervisée [81],
2. la **description** des propriétés distinctives des classes identifiées à l'étape précédente en utilisant les attributs A_π sur lesquels les résultats initiaux étaient projetés,
3. puis la **caractérisation** des classes à travers la découverte de propriétés distinctives supplémentaires sur les attributs A_ω, attributs initialement non présents dans la relation retournée par l'exécution de Q.

L'utilisation d'une étape de classification non-supervisée vise à identifier des groupes de résultats partageant des propriétés communes au sein d'une classe mais distinctives entre classes. La démarche suivie dans ce travail consiste à accompagner les résultats de Q d'explications linguistiques sur les propriétés structurelles de ces différents groupes de résultats. L'utilisation de cette étape de structuration des résultats constitue une différence majeure par rapport à l'approche présentée dans l'article [34] dont l'objectif est pourtant identique.

Les explications, que nous distinguons sous les termes de description et de caractérisation pour souligner le fait qu'elles décrivent des propriétés présentes ou non dans la relation résultat Σ_Q, sont formalisées de la même façon. Une description (resp. caractérisation) d'une classe C est formée par un ensemble de couples $(attribut, SEF\,de\,descripteurs)$ associant à chaque attribut de A_π (resp. A_ω) l'ensemble des descripteurs qui décrivent C, où chaque descripteur est associé à un degré de couverture (i.e. Σ-count). Une représentation linguistique est générée en interprétant de manière conjonctive les différents SEF associés à une classe. Un exemple de description linguistique sur les attributs $A_\pi = \{anne, km\}$ d'une classe C regroupant des voitures d'occasion est donnée ci-dessous :

"Pour la classe C_1, année est récente $\langle 1 \rangle$ et (km est faible $\langle 0.8 \rangle$ ou moyen $\langle 0.2 \rangle$)";

Alors que les descriptions de classes sont identifiées en projetant simplement les objets de chaque classe sur les partitions du vocabulaire définies sur A_π, les compléments d'informations que sont les caractérisations doivent posséder des propriétés particulières pour être réellement informatives. En effet, considérant l'ensemble des combinaisons possibles de descripteurs pouvant être construites par la projection des objets d'une classe C sur les partitions des attributs A_ω, nous avons défini un algorithme permettant d'identifier les caractérisations qui sont à la fois spécifiques et minimales. La spécificité est une mesure quantifiant le caractère couvrant et discriminant d'une explication par rapport aux autres classes. La minimalité est relative au nombre de descripteurs incluses dans une explication et vise, pour un degré de spécificité égal, à favoriser les explications les plus concises. Les classes identifiées par l'algorithme de classification ne sont pas toujours parfaitement séparables et il est donc parfois impossible de trouver des caractérisations intéressantes rendant compte de l'intégralité d'une classe. En cas de faible séparabilité des classes, et à des fins de robustesse, nous avons défini une stratégie visant à fournir tout de même des explications sur un sous-ensemble, aussi grand que possible, des objets les plus représentatifs des classes.

Des expérimentations menées sur des données réelles issues d'une BD d'annonces de voitures d'occasion montrent la plus value informationnelle de ces descriptions des résultats de requêtes et de leur enrichissement par des caractérisations complémentaires. Le coût de cette étape coopérative a été quantifié et est illustré par la figure 8.1. Ces expérimentations montrent que l'identification des classes au sein de l'ensemble des résultats est l'étape la plus coûteuse de l'approche. Il est

important de souligner que les données sur lesquelles cette stratégie coopérative est appliquée correspondent aux résultats d'une requête et non à une source de données entière, on peut donc faire l'hypothèse que sa taille est assez limitée. La dépendance exponentielle par rapport au nombre d'attributs considérés pour la génération des résultats induit évidemment une limite sur le nombre de dimensions à considérer. En ce sens, cette approche pourrait être utilisée de manière complémentaire à une approche statistique de type *Principal Component Analysis* [73] dont l'objectif est d'identifier efficacement les dimensions discriminantes pour un ensemble de données. L'approche de description et de caractérisation des résultats serait alors uniquement appliquée sur ces dimensions pertinentes.

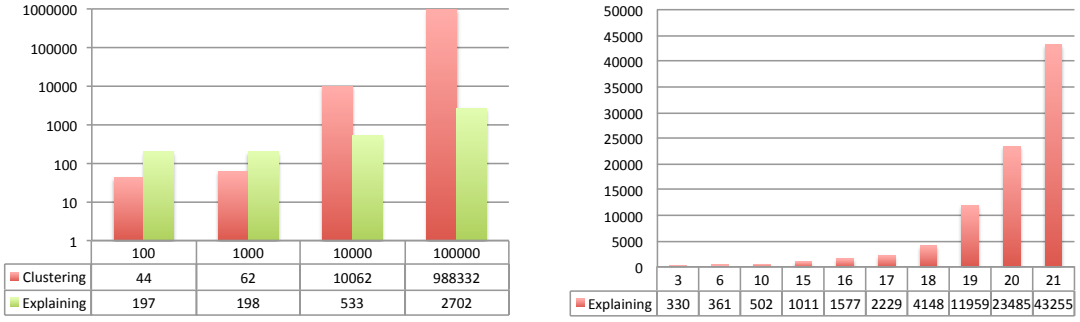

FIGURE 34 – Coût de la génération des explications de réponses en fonction des données (gauche) et du vocabulaire (droite)

8.2 Résumé de la structure intrinsèque d'un ensemble de réponses

En complément de l'approche décrite dans la section précédente, nous avons défini une méthode visant à générer des résumés linguistiques imbriqués décrivant le contenu d'un ensemble de classes. Ce travail décrit dans l'article [127] peut être utilisé pour résumer le résultat d'une requête et ainsi en donner un aperçu à l'utilisateur. Résumer un ensemble de données à l'aide de descripteurs issus d'un vocabulaire utilisateur est une problématique largement étudiée par la communauté du soft computing [7]. La particularité de l'approche présentée ici réside dans le fait que nous cherchons à résumer un ensemble de réponses à travers sa structure intrinsèque en classes. Tout comme dans l'approche décrite précédemment, il est considéré que l'identification de groupes d'objets partageant des propriétés communes, distinctives de celles des autres classes, constitue déjà un élément d'information aidant à l'appropriation d'un ensemble de résultats.

Dans de nombreux cas, les classes obtenues par un processus de classification automatique non supervisée ne sont pas toujours complètement séparées et il est alors difficile de trouver des caractérisations distinctives telles que celles recherchées par l'approche précédente. Cette seconde approche d'explication d'un ensemble de réponses à une requête vise à produire de manière robuste des résumés linguistiques de chaque classe en se focalisant sur ses éléments les plus représentatifs.

Explication et coupes par typicité

Formellement, une explication E d'un ensemble S d'objets selon un vocabulaire $\mathcal{V}$ est composée d'une conjonction de termes issus de $\mathcal{V}$. La mesure de Σ-count est utilisée pour quantifier la validité de E pour décrire S :

$$\mu_E(S) = \Sigma_E^{count}(s) = \frac{\sum_{x \in S} \mu_E(x)}{|S|}.$$

Les objets d'une classe peuvent être caractérisés par leur représentativité vis-à-vis des propriétés structurelles de la classe. Cette représentativité est quantifiable par un indice de typicité [83] exprimant le fait que d'une part l'objet partage de nombreuses propriétés communes avec les

autres membres de la classe et d'autre part tout en étant distincts des membres des autres classes. Le degré de typicité d'un objet d par rapport à une classe C est noté $typ_C(d)$. En s'appuyant sur ce degré de typicité associé à chaque élément d'une classe, il est possible de construire des sous-ensembles imbriqués correspondant à des coupes de typicité comme l'illustre la figure 8.2. Pour un degré α pris dans une échelle prédéfinie, la coupe de typicité d'une classe C est définie comme suit :

$$FC^{\alpha} = \{d \in C, typ(d) \geq \alpha\}.$$

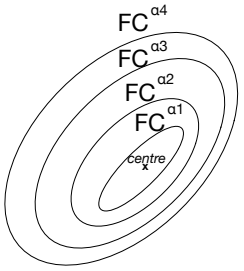

FIGURE 35 – Coupes de typicité imbriquées d'une classe

Génération des résumés linguistiques

L'algorithme de génération des résumés imbriqués d'une classe traite tout d'abord la coupe de typicité la plus élevée, c'est à dire celle correspondant aux données prototypiques de la classe ($FC^{\alpha=1}$). L'objectif étant de générer des explications des données pour des coupes de typicité, les explications à étudier sont constituées de toutes les combinaisons de termes qui décrivent le prototype de la classe. Ainsi, pour chaque explication candidate E, l'algorithme détaillé dans le papier [127] identifie l'ensemble maximal de données de la classe, par intégration successive de coupes de typicité de rangs inférieurs, qui est expliqué par E :

Definition *Soit E une explication (combinaison conjonctive de termes issus du vocabulaire $\mathcal{V}$) pour une coupe de typicité FC^{α}.* $\boldsymbol{FC^{\alpha}}$ *est un* **ensemble maximal expliqué par** $\boldsymbol{E}$ *ssi.* $\nexists\, \alpha' < \alpha$ *tel que E explique également $FC^{\alpha'}$.*

Pour illustrer ce processus de génération d'explications, la figure 36 donne un exemple jouet de deux classes de données, C_1 et C_2, décrites sur deux dimensions discrétisées par les partitions $\{P_{11}, P_{12}, P_{13}\}$ et $\{P_{21}, P_{22}\}$. Le tableau 11 nous indique la réécriture des objets selon le vocabulaire ainsi que le degré de typicité de chaque objet vis-à-vis de sa classe d'appartenance.

La table 12 montre comment à partir des explications candidates initiales, celles décrivant le prototype de chaque classe, sont identifiés les ensembles maximaux d'éléments associés à chaque explication.

La structure intrinsèque des données identifiée par un algorithme de classification non supervisée regroupe des objets dans une même classe parce qu'ils partagent des propriétés communes mais qui les différencient également des autres classes. Les explications linguistiques associées aux classes doivent également rendre compte de cette séparabilité des classes. Nous utilisons alors une mesure appelée représentativité pour quantifier le caractère distinctif de chaque explication validée. Cette mesure, considérée comme une quantification du caractère informatif d'une explication linguistique, s'apparente sémantiquement à une mesure de spécificité [143], tout en reprenant les deux composantes des mesures duales de typicité [83].

La table 13 donne les explications trouvées pour les deux classes de l'exemple jouet illustré par la figure 36.

Ainsi, quatre indicateurs numériques sont associés à chaque explication E :

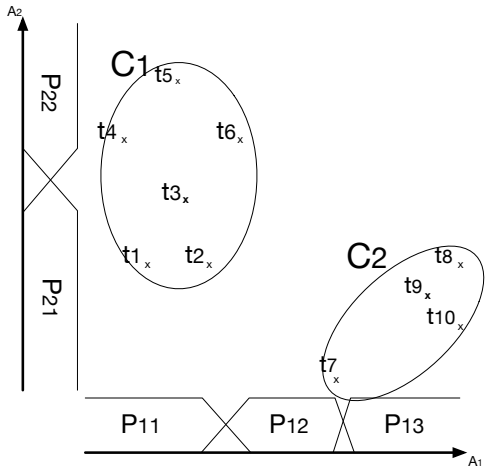

FIGURE 36 – Classes et vocabulaire

t	(x, y)	$\mu_{P_{11}}$	$\mu_{P_{12}}$	$\mu_{P_{13}}$	$\mu_{P_{21}}$	$\mu_{P_{22}}$	$typ(t)$
t_3	$(3, 6)$	1	0	0	.9	.1	.65
t_6	$(3, 10)$	.3	.7	0	0	1	.634
t_2	$(4, 4)$	.9	.1	0	1	0	.628
t_4	$(1, 8)$	1	0	0	0	1	.627
t_1	$(2, 4)$	1	0	0	1	0	.614
t_5	$(5, 8)$	1	0	0	0	1	.585
t_{10}	$(12, 1)$	0	0	1	1	0	.679
t_9	$(12, 2)$	0	0	1	1	0	.677
t_8	$(11, 3)$	0	0	1	1	0	.674
t_7	$(8, 0)$	0	.9	.1	1	0	.514

TABLE 11 – Réécritures des objets classés par ordre décroissant de leur typicité

α-coupe de C_1 $\downarrow$ *typicit*	explications candidates	explications validées
$S^1 = FC_1^{.65} = \{t_3\}$	$\{P_{11}, P_{21}, P_{22}, P_{11} \wedge P_{21}, P_{11} \wedge P_{22}\}$	$\emptyset$
$S^2 = FC_1^{.634} = S^1 \cup \{t_6\}$	$\{P_{11}, P_{22}, P_{11} \wedge P_{22}\}$	$\{(P_{21}, S^1), (P_{11} \wedge P_{21}, S^1)\}$
$S^3 = FC_1^{.628} = S^2 \cup \{t_2\}$	$\{P_{11}\}$	$\{(P_{22}, S^2), (P_{11} \wedge P_{22}, S^2)\}$
$S^3 = FC_1^{.627} = S^3 \cup \{t_6\}$	$\{P_{11}\}$	
$S^3 = FC_1^{.614} = S^3 \cup \{t_1\}$	$\{P_{11}\}$	
$S^3 = FC_1^{.585} = S^3 \cup \{t_5\}$	$\{P_{11}\}$	$\{(P_{11}, S^3)\}$
α-coupe de C_2 $\downarrow$ *typicit*	explications candidates	explications validées
$S^{1'} = FC_2^{.679} = \{t_{10}\}$	$\{P_{13}, P_{21}, P_{13} \wedge P_{21}\}$	$\emptyset$
$S^{1'} = FC_2^{.677} = S^{1'} \cup \{t_9\}$	$\{P_{13}, P_{21}, P_{13} \wedge P_{21}\}$	$\emptyset$
$S^{1'} = FC_2^{.674} = S^{1'} \cup \{t_8\}$	$\{P_{13}, P_{21}, P_{13} \wedge P_{21}\}$	$\emptyset$
$S^{1'} = FC_2^{.514} = S^{1'} \cup \{t_7\}$	$\{P_{13}, P_{21}, P_{13} \wedge P_{21}\}$	$\{(P_{13}, S^{1'}), (P_{21}, S^{1'}), (P_{13} \wedge P_{21}, S^{1'})\}$

TABLE 12 – Identification des ensembles maximaux pour chaque explication candidate initiale

Explication	Représentativité
P_{21} for $S^1 = \{t_3\}$	0
$P_{11} \wedge P_{21}$ for $S^1 = \{t_3\}$	0.9
$P_{11} \wedge P_{22}$ for $S^2 = \{t_3, t_6\}$	0.55
P_{22} for $S^2 = \{t_3, t_6\}$	0.55
P_{11} for $S^3 = C_1$	0.87
P_{13} for $S^{1'} = C_2$	0.775
P_{21} for $S^{1'} = C_2$	0.1
$P_{13} \wedge P_{21}$ for $S^{1'} = C_2$	0.775

TABLE 13 – Représentativité des explications validées

— $\frac{|S|}{|C|}$ la proportion d'élements de C couverte par E où S est l'ensemble des objets de C décrit par E,
— α le seuil de typicité de la coupe, $\alpha = min_{d \in S} typ(d)$,
— $\mu_E(S)$ la validité de E pour le sous-ensemble $S \subseteq C$,
— et $\tau_E(S)$ la représentativité de E par rapport à S.

Les indicateurs de représentativité ainsi que le seuil de typicité sont fusionnés pour former un score :

$$score(E, FC^{\alpha}) = \alpha \times \tau_E(S),$$

les deux autres indicateurs étant traduits linguistiquement et/ou graphiquement.

Représentation linguistique puis graphique des résumés

Les explications générées sur l'ensemble des objets retournés par une requête sont transmises à l'utilisateur sous deux versions. La première est linguistique où chaque explication est traduite sous forme d'un syntagme nominal. La table 14 donne la traduction linguistique des explications ainsi que leur score et degré de vérité [7] calculé comme suit :

$$\Psi(Q \text{ elements of } C \text{ are } E) = \mu_Q(\frac{1}{|C|} \sum_{d \in S} \mu_E(d)),$$

où Q est un quantificateur flou (e.g. {*a few*, *some*, *most of*, *all of*}).

Explanation	Q	Score	Truth	Linguistic translation
$P_{11} \wedge P_{21}$ for $S^1 = \{t_3\}$	few	0.65	0.15	few of the elements from C_1 are P_{11} and P_{21}
P_{11} for $S^3 = C_1$	all	0.585	1	all the elements from C_1 are P_{11}
$P_{11} \wedge P_{22}$ for $S^2 = \{t_3, t_6\}$	some	0.55	0.07	some of the elements from C_1 are P_{11} and P_{22}
P_{22} for $S^2 = \{t_3, t_6\}$	some	0.55	1	some of the elements from C_1 are P_{22}
P_{13} for $S^{1'} = C_2$	all	0.514	0.8	all of the elements from C_2 are P_{13}
$P_{13} \wedge P_{21}$ for $S^{1'} = C_2$	all	0.514	1	all of the elements from C_2 are P_{13} and P_{21}
P_{21} for $S^{1'} = C_2$	all	0.1	0.8	all of the elements from C_2 are P_{21}

TABLE 14 – Version linguistique des explications

La seconde version est graphique comme l'illustre la figure 37 où chacun des quatre indicateurs associés aux explications est traduit par une propriété graphique (taille, opacité, couleur, position).

De par sa complexité linéaire vis-à-vis du nombre d'objets à expliquer mais de sa complexité exponentielle vis-à-vis du nombre de termes expliquant chaque centre de classe, cette approche coopérative peut être utilisée efficacement pour décrire un ensemble volumineux de données décrit sur un nombre de dimensions modeste (≈ 10). Ce travail a également constitué une première contribution à la représentation graphique de connaissances pour un système d'interrogation coopérative de données, contribution que nous avons ensuite reprise pour aboutir à un système plus complet d'exploration graphique de données (Sec. 10).

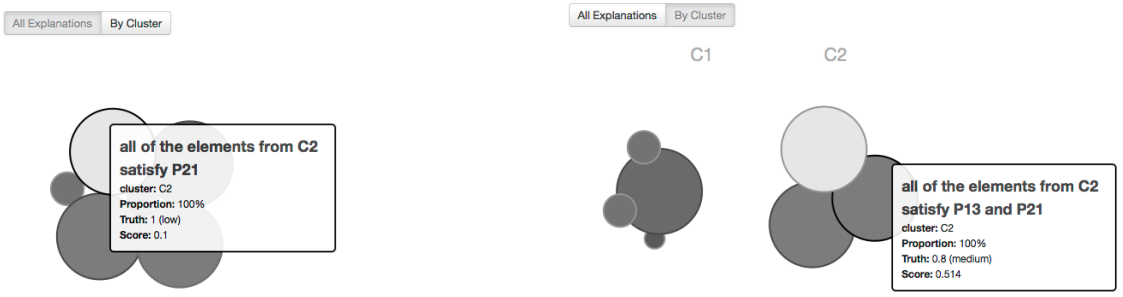

FIGURE 37 – Explications des données regroupées (à gauche), séparées par classe (à droite)

8.3 Expliquer les raisons d'un ensemble vide de réponses

Comme énoncé en introduction, l'objectif d'une approche coopérative est d'éviter que l'utilisateur se retrouve dans une situation non informative lors de ses interactions avec un SGBD. Une de ces situations que les approches coopératives tentent d'éviter concerne le cas où le SGBD retourne un ensemble vide de réponses à une requête utilisateur. Pour pallier la vacuité d'un ensemble vide de réponses, deux approches coopératives ont été envisagées. Au lieu d'indiquer laconiquement à l'utilisateur « Pas de résultat pour votre requête », une approche coopérative va chercher à identifier les raisons de la vacuité de l'ensemble résultat (absence de données quelque peu satisfaisantes, conflits dans la requête, etc.), puis suggérer des requêtes alternatives (Sec. 9.2). Ces requêtes alternatives correspondent à des versions moins restrictives de la requête initiale, proches sémantiquement de cette dernière, dont on sait qu'elles retourneront un ensemble non-vide de résultat.

La notion de MFS

En partant d'une requête initiale Q composée d'une conjonction de prédicats dont l'ensemble résultat est vide ($\Sigma_Q = \emptyset$), la première étape de cette approche coopérative repose sur l'identification des sous-requêtes de Q responsables de la vacuité de l'ensemble résultat Σ_Q. En effet, l'échec de Q est nécessairement dû à un support vide (vis-à-vis de l'état courant de la BD) pour au moins l'une des sous-requêtes de Q.

Définition Soit Q une requête de la forme $Q = P_1 \wedge P_2 \wedge \ \ldots \ \wedge P_k$, et soit S et S' deux sous-ensembles de prédicats tels que $S' \subset S \subseteq \{P_1,\ P_2,\ \ldots\ ,\ P_k\}$. Une conjonction d'éléments de S (resp. S') est une *sous-requête* (resp. *sous-requête stricte*) de Q.

Ordonné par une relation d'inclusion, l'ensemble de toutes les sous-requêtes d'une requête conjonctive, e.g. $Q = P_1 \wedge P_2 \wedge P_3 \wedge P_4$, forme un treillis (voir figure 38), avec Q comme borne haute et $\emptyset$ comme borne basse.

Définition Une *sous-requête conflictuelle minimale (MFS)* d'une requête $Q = P_1 \wedge P_2 \wedge \ldots \wedge P_n$ est toute sous-requête Q' de Q telle que i) $\Sigma_{Q'} = \emptyset$ et ii) pour toute sous-requête stricte Q'' de Q', on a $\Sigma_{Q''} \neq \emptyset$.

L'identification des MFS repose sur le parcours en profondeur d'abord du treillis afin d'identifier la frontière haute des sous-requêtes retournant un ensemble vide de résultats.

Les aspects algorithmiques de cette problématique ont tout d'abord été étudiés dans [60] où l'auteur a montré que, bien qu'il soit aisé d'identifier une MFS quelconque, le calcul de *toutes* les raisons minimales d'un échec est un problème NP-difficile qui conduit à exécuter un nombre exponentiel de sous-requêtes. Alors que les bases de données deviennent de plus en plus volumineuses, il n'est pas concevable d'exécuter de nombreuses requêtes pour établir les causes de l'échec d'une requête initiale. Dans [72] Jannach propose une technique pour identifier les MFS basée sur un unique parcours séquentiel de la BD durant lequel une matrice binaire est construite. Cette

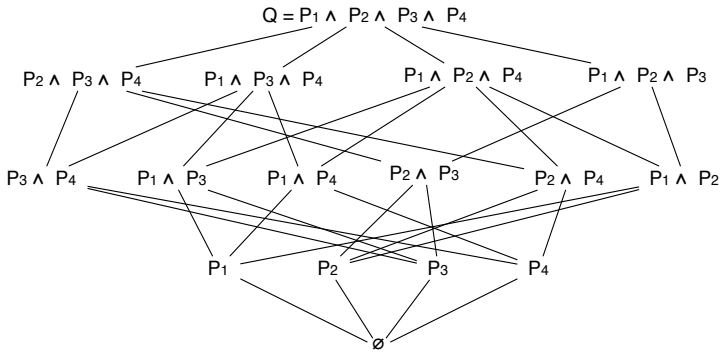

FIGURE 38 – Treillis des sous-requêtes de $Q = P_1 \wedge P_2 \wedge P_3 \wedge P_4$

matrice contient autant de lignes qu'il y a d'objets, et autant de colonnes qu'il y a de prédicats dans la requête, ce qui apparaît peu réaliste quand la relation concernée est très volumineuse.

Identification des MFS pour les requêtes flous

Dans l'article [107], nous avons proposé une généralisation du concept de "Minimal Failing Subquery" (MFS) [60].

Du fait de la satisfaction graduelle des objets vis-à-vis d'une condition imprécise, le résultat d'une requête floue peut ne pas être vide, stricto sensu, mais sans pour autant comporter des tuples vraiment satisfaisants. Il convient alors d'identifier pour différents niveaux de satisfiabilité (seuils α pris dans une échelle $\alpha_1 = 1 > \alpha_2 > \ldots > \alpha_f = 0^+$) les raisons minimales à cause desquelles des tuples plus intéressants ne sont pas retournés.

Définition Une *sous-requête conflictuelle minimale* d'une requête floue $Q = P_1 \wedge P_2 \wedge \ldots \wedge P_n$ pour un α donné est toute sous-requête Q' de Q telle que i) $\Sigma^{\alpha}_{Q'} = \emptyset$ et ii) pour toute sous-requête stricte Q'' de Q', on a $\Sigma^{\alpha}_{Q''} \neq \emptyset$.

Propriété Du fait de la monotonie de la satisfiabilité des α-coupes par inclusion, on a $\Sigma^{\alpha_i}_{Q} \subseteq \Sigma^{\alpha_j}_{Q}$ si $\alpha_i \geq \alpha_j$. En conséquence, une requête Q qui échoue pour un α_j donné échoue aussi pour les seuils plus élevés $\alpha_i > \alpha_j$. On n'a cependant pas cette propriété de monotonie pour la minimalité des sous-requêtes conflictuelles. En effet, une sous-requête Q' peut être une MFS de Q pour un α_j donné sans toutefois être minimale pour un seuil $\alpha_i > \alpha_j$ puisqu'une sous-requête stricte de Q' peut échouer au niveau α_i et pas au niveau α_j.

Pour savoir quelles sous-requêtes de Q échouent et à partir de quel degré α, nous avons proposé dans l'article [107] d'utiliser les cardinalités floues, à la place d'une matrice binaire [72], pour construire un résumé de la base de données. Comme indiqué dans la section 6.2, les cardinalités floues offrent un formalisme intéressant pour matérialiser la distribution de données selon un vocabulaire utilisateur. Nous avons alors défini un algorithme utilisant les propriétés des MFS et le résumé, sous forme de cardinalités floues, de la BD pour filtrer le treillis des conjonctions possibles et ainsi identifier les MFS pour chaque degré de satisfaction $\alpha_1 = 1 > \alpha_2 > \ldots > \alpha_f = 0^+$. Cet algorithme parcourt le treillis de bas en haut, comme illustré par la figure 39, et vérifie dans le résumé de la BD si des objets satisfont la conjonction concernée. Si tel n'est pas le cas, la propriété énoncée précédemment est utilisée pour filtrer le treillis.

Voici, ci-dessous un exemple d'explications générées en cas d'échec d'une requête incluant la clause de sélection $A \wedge B \wedge C \wedge D \wedge E$:

Aucun élément ne satisfait simultanément A, B, C, D et E pour les raisons suivantes ...
— *aucun élément ne satisfait E,*

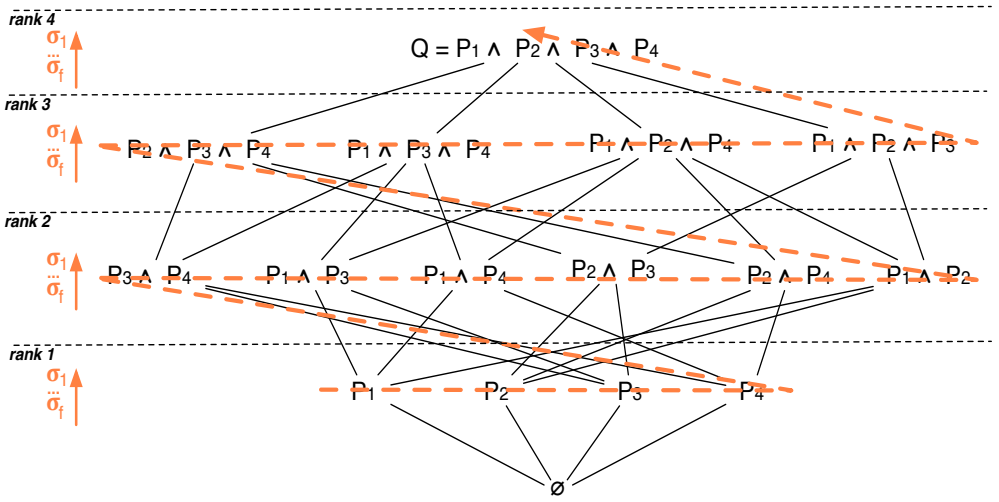

FIGURE 39 – Parcours bas-haut en largeur du treillis pour identifier les MFS

— *aucun élément ne satisfait A et D, ainsi que B ou C à un degré* ≥ 0.2,
— *aucun élément ne satisfait A ainsi que B ou C à un degré* ≥ 0.4,
— *aucun élément ne satisfait A à un degré* ≥ 0.6 ;
— *aucun élément ne satisfait B et C à un degré* ≥ 0.8,
— *aucun élément ne satisfait pleinement B*.

Ces informations sont très riches car elles permettent d'identifier les combinaisons non présentes dans la BD ou constituant des non-sens (e.g. $A \wedge B \wedge C$), et également les descripteurs potentiellement trop restrictifs (e.g. B). L'utilisateur sait alors que son critère de recherche B ne pourra jamais être pleinement satisfait et qu'un compromis est alors à accepter.

Expérimentations et bilan

L'approche a été implémentée puis expérimentée sur des données réelles (92 178 descriptions de voitures d'occasion stockées dans une relation de schéma {*prix*, *marque*, *kilométrage*, *année*, *longueur*, *hauteur*, *nbsièges*, *accélération*, *consommation*, *émissionco2*}) [13]. Ce travail sur l'identification des MFS a été l'occasion de mener de nouvelles expérimentations sur la construction d'un résumé de la BD à l'aide de cardinalités floues. Le figure 40 confirme la dépendance linéaire du processus de résumé (en temps de calcul) par rapport à la taille des données et la figure 41 illustre le phénomène de convergence du nombre de cardinalités à stocker, et donc de l'occupation mémoire du résumé, grâce aux corrélations entre attributs. L'apport d'une stratégie d'identification des MFS s'appuyant sur un résumé de la base est confirmé par la figure 43, qui compare notre approche avec une stratégie naïve où toute MFS candidate donne lieu à l'exécution d'une requête. Enfin, la figure 42 montre, qu'une fois le résumé de la partie de la BD concernée par la requête réalisé, c'est-à-dire des tuples satisfaisant la version disjonctive de la requête intiale, la détection des MFS graduelles est très efficace.

9 Enrichir les réponses

Après avoir présenté dans la section précédente (Sec. 8) mes contributions relatives à l'explication du résultat, potentiellement vide, d'une requête, cette section aborde un aspect complémentaire des systèmes coopératifs : l'enrichissement des résultats d'une requête. Cet enrichissement peut prendre plusieurs formes. Il peut tout d'abord s'agir de réduire la taille d'un ensemble de résultats

13. Expérimentations réalisées sur un Intel Core 2 Duo 2.53GHz avec 4Go 1067 MHz de ram DDR3

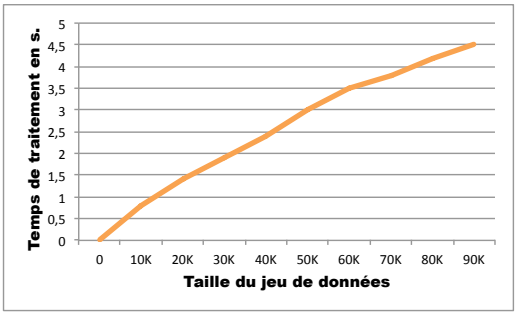

FIGURE 40 – MFS : calcul du résumé / taille de la base

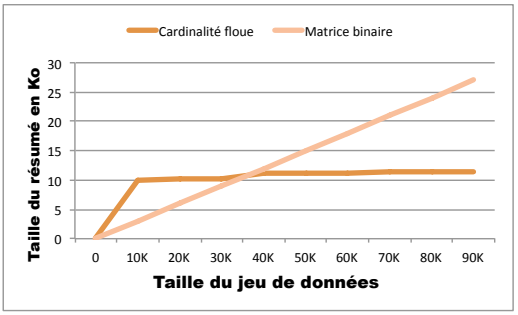

FIGURE 41 – MFS : taille du résumé / taille de la base

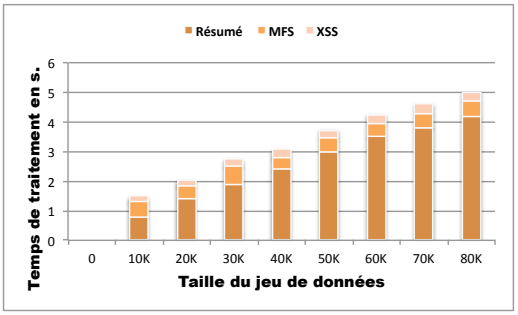

FIGURE 42 – MFS : temps de calcul des diverses étapes (résumé, explication, réparation)

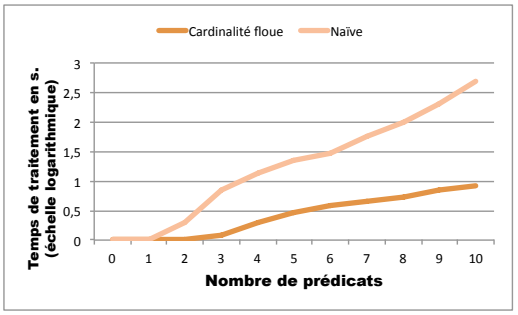

FIGURE 43 – MFS : approche cardinalité floue vs. approche naïve (échelle logarithmique)

trop volumineux pour être appréhendé efficacement par l'utilisateur et ainsi l'aider à identifier le sous-ensemble des objets les plus intéressants (Sec. 9.1). À l'opposé, en cas d'échec, nous avons vu qu'une première étape importante concerne l'explication des raisons de l'échec, ceci afin de permettre à l'utilisateur de mieux comprendre les données manipulées. Les explications de l'échec sont ensuite utilisées pour suggérer des versions moins restrictives de la requête initiale garantissant à l'utilisateur d'obtenir des résultats (Sec. 9.2). Les deux premières approches coopératives présentées dans cette section s'insèrent dans la continuité de travaux initiés par l'équipe PILGRIM avant mon arrivée. Cette section se termine par la présentation de travaux réalisés avec A. Moreau, dans le cadre de son Master recherche puis de sa thèse, sur la recommandation de résultats indirects. Les systèmes de recommandation jouent désormais un rôle important dans les interfaces d'accès aux données, et nous avons étudié l'apport de stratégies relevant du soft computing pour cette problématique (Sec. 9.3).

9.1 Intensifier une requête par ajout de prédicats corrélés

Lorsqu'une requête retourne un ensemble trop volumineux de résultats, un système dit coopératif se doit de fournir des stratégies permettant à l'utilisateur d'identifier les objets les plus intéressants. Soit Q une requête conjonctive $P_1 \wedge P_2 \wedge \ldots \wedge P_3$ retournant un ensemble pléthorique d'objets Σ_Q. Deux stratégies sont alors envisageables pour réduire la cardinalité de Σ_Q : appliquer une mesure de discrimination sur les objets retournés (en prenant e.g. en compte des préférences supplémentaires non présentes dans la requête) ou rendre plus sélectifs certains prédicats de la requête. L'équipe PILGRIM a proposé dans [11] une approche appartenant à la seconde stratégie, approche basée sur l'intensification de la requête initiale par application d'une opération d'érosion sur les prédicats flous impliqués dans la requête. Cette stratégie traite de manière indifférenciée les prédicats de la requête. Dans l'article [14], nous avons proposé une stratégie visant à intensifier une requête à travers l'ajout de nouveaux prédicats. Ces prédicats supplémentaires sont choisis de sorte à ce qu'ils réduisent le nombre de résultats tout en respectant autant que faire se peut la

sémantique de la requête initiale.

Présentation du problème

On considère tout d'abord des requêtes non floues de la forme :

$$Q : SELECT\ *\ FROM\ R\ WHERE\ Z_1 = z_1\ AND\ Z_2 > z_2\ AND\ \ldots\ AND\ Z_k\ IN\ (z_{k1}, z_{k2}, ..., z_{ks});,$$

où plus généralement des requêtes composées d'une conjonction de prédicats d'égalité, inégalité, d'intervalle, d'appartenance, etc. Si l'ensemble des résultats de Q a une cardinalité telle qu'il est difficile pour un utilisateur d'en étudier le contenu, nous proposons d'intégrer de nouveaux prédicats à la requête initiale Q pour former une requête Q' plus sélective et ainsi obtenir un ensemble $\Sigma_{Q'}$, tq. $\Sigma_{Q'} \subset \Sigma_Q$.

Nous supposons définie sur chaque attribut, disons A_i, de la relation interrogée une partition (non floue) $P_i = \{P_{i1}, P_{i2}, ..., P_{is_i}\}$, où chaque élément de la partition est associé à une étiquette linguistique L_i. Tels des histogrammes maintenus par un SGBD pour connaître la distribution des tuples sur un attribut, nous considérons disponible pour chaque élément de partition la cardinalité de l'ensemble des objets décrits par l'intervalle concerné. Comme l'illustre la figure 44, ces partitions strictes constituent des cas particuliers de partitions floues.

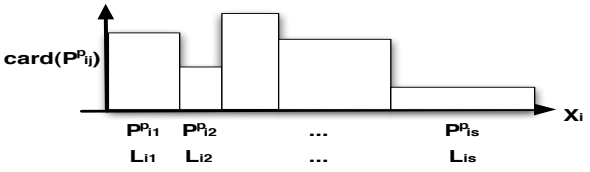

FIGURE 44 – Partition et étiquetage d'un attribut

Les prédicats ajoutés pour intensifier la requête initiale Q sont sélectionnés dans ces partitions et utilisés pour former des prédicats d'intervalle de la forme A_i BETWEEN $bmin(P_{ij})$ AND $bmax(P_{ij})$ si A_i est numérique et A_i IN $elem(P_{ij})$ [14] si A_i est un attribut catégoriel.

Choix des prédicats corrélés

Soit une requête Q et Σ_Q l'ensemble des objets retournés. Les prédicats formés à partir des éléments de partitions pour intensifier Q sont sélectionnés pour leur lien sémantique avec les prédicats présents dans Q et leur capacité à réduire le nombre de résultats.

Soit P un prédicat candidat pour l'intensification de Q et B l'ensemble des objets satisfaisant ce prédicat. Le degré de corrélation entre P et Q, noté $cor(P, Q)$, repose sur la comparaison de Σ_Q avec B :

$$cor(P, Q) = \frac{card(\Sigma_Q \cap B)}{card(\Sigma_Q \cup B)}.$$

Afin d'éviter d'utiliser un prédicat complètement corrélé à Q qui ne permettrait pas de réduire la cardinalité de l'ensemble des résultats, une fonction d'appartenance triangulaire sur l'intervalle $[0, 1]$ est utilisée pour favoriser les prédicats offrant un compromis entre corrélation et capacité de réduction. Cette fonction, notée $\mu_{corMod} : [0, 1] \rightarrow [0, 1]$, est définie par un seul point prédéfini γ caractérisant le degré de corrélation de référence comme l'illustre la figure 45. Le résultat de l'application de cette fonction sur un degré de corrélation $cor(P, Q)$ est appelé degré de corrélation modifié noté $corMod(P^p_{ij}, Q)$. Le point γ permet d'influer sur le comportement du processus d'intensification. En favorisant les degrés de corrélation élevés, les prédicats prédéfinis les plus corrélés

14. $bmin(P_{ij})$ et $bmax(P_{ij})$ sont les bornes de l'intervalle définissant l'élément P_{ij} et $elem(P_{ij})$ désigne l'ensemble des catégories inclues dans P_{ij}.

seront privilégiés au détriment de leur capacité de réduction. Les degrés les plus faibles permettront une réduction plus rapide de l'ensemble initial des résultats mais entraîneront l'ajout de prédicats moins corrélés.

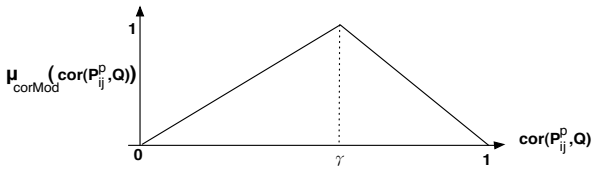

FIGURE 45 – Fonction de calcul du degré de corrélation modifié

Les prédicats utilisés pour l'intensification étant prédéfinis, une matrice de corrélation entre prédicats est utilisée. Nous effectuons tout d'abord une projection des prédicats définis dans Q, non connus à l'avance, sur les partitions prédéfinies pour identifier à quels prédicats ils se rapportent. La matrice de corrélation nous permet ensuite d'identifier les prédicats les plus corrélés à Q, non présents dans Q, afin de les ajouter pour obtenir une requête Q' dont l'ensemble résultat est exploitable par l'utilisateur. Les prédicats corrélés sont ajoutés successivement pour réduire l'ensemble Σ_Q (algorithme disponible dans l'article [14]).

Extension de l'approche aux requêtes floues

L'approche d'intensification de requêtes présentée précédemment a ensuite été étendue pour prendre en compte des requêtes floues [13]. Cette extension passe tout d'abord par l'utilisation de partitions floues pour définir le vocabulaire d'interrogation. Les connaissances sur la distribution des données par rapport à ce vocabulaire d'interrogation sont dans ce cas matérialisées par des cardinalités floues (Sec. 6.2).

La corrélation entre un prédicat, disons P, du vocabulaire d'interrogation et une requête initiale Q est quantifiée par le degré de confiance des règles d'association $Q \rightarrow P$ et $P \rightarrow Q$ comme suit :

$$\mu_{cor}(P^p_{i,j}, Q) = \top(\textit{confidence}(Q \Rightarrow P^p_{i,j}), \textit{confidence}(P^p_{i,j} \Rightarrow Q))$$

où le degré de confiance est obtenu de la façon suivante : $\textit{confidence}(P^a \Rightarrow P^b) = \frac{\Gamma_{P^a \wedge P^b}}{\Gamma_{P^a}}$, Γ_{P^a} (resp. $\Gamma_{P^a \wedge P^b}$) étant la cardinalité scalaire associée au prédicat P^a (resp. à la conjonction $P^a \wedge P^b$).

Cette mesure est utilisée pour identifier les η prédicats les plus corrélés à la requête initiale. Ces η prédicats sont ensuite triés de nouveau pour prendre en compte leur capacité à réduire l'ensemble des résultats de la requête Q. Pour ce faire, nous utilisons la cardinalité floue associée à la combinaison conjonctive de Q et du prédicat candidat P pour vérifier dans quelle mesure cette intensification candidate permet de s'approcher d'un ensemble de K résultats, K étant un seuil quantitatif défini par l'utilisateur dans sa requête (comme dans les approches *top-k queries*). Un score final d'intensification $\mu_{stren}(F_{Q \wedge P})$ quantifie la capacité du prédicat P à conduire à un résultat de K objets :

$$\mu_{stren}(F_{Q \wedge P^{c^r}_Q}) = sup_{1 \leq i \leq f} \top(1 - \frac{|c_i - K|}{max(K, |\Sigma_Q| - K)}, \alpha_i),$$

où les α_i sont des seuils de satisfaction pris dans une échelle ordonnée et c_i est le nombre de tuples satisfaisant la conjonction concernée à un degré supérieur ou égal à α_i.

Les η prédicats les plus corrélés à Q sont donc ensuite suggérés à l'utilisateur par ordre décroissant de leur capacité de réduction μ_{stren}.

Le prototype CORTEX

Afin de montrer la cohérence d'une approche d'intensification de requête basée sur la corrélation entre prédicats, nous avons implémenté un prototype de recherche nommé CORTEX [12] que nous

avons intégré dans une interface d'interrogation de données réelles. Ce prototype permet à l'utilisateur de construire des requêtes floues en sélectionnant des termes issus d'un vocabulaire prédéfini (Fig. 46). Un résumé des données sous forme de cardinalités floues est utilisé pour connaître la distribution des données vis-à-vis du vocabulaire d'interrogation et ainsi quantifier la corrélation entre prédicats flous.

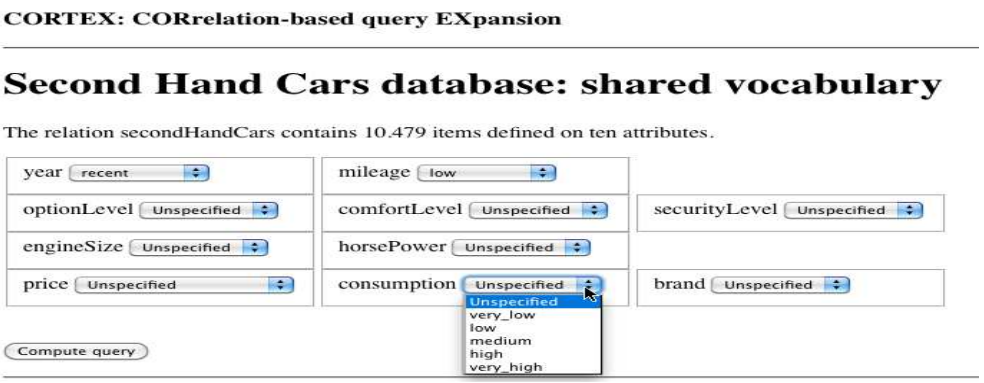

FIGURE 46 – Interface simplifiée d'interrogation du prototype Cortex

Une fois la requête exécutée, l'utilisateur peut demander au système CORTEX de lui suggérer des prédicats corrélés pour intensifier sa requête initiale (Fig. 47). Les expérimentations menées ont montré la pertinence des prédicats suggérés et que leur intégration permettait progressivement de réduire l'ensemble initial des résultats. L'utilisation d'un résumé des données à l'aide de cardinalités floues permet de disposer en temps constant de toutes les informations nécessaires à l'identification des prédicats corrélés qui seront suggérés pour l'intensification. Comme toutes les autres stratégies basées sur des cardinalités floues, cette approche d'intensification de requêtes à résultats pléthoriques peut être appliquée dans de nombreux contextes applicatifs manipulant des données d'assez grande taille à condition d'utiliser un vocabulaire prédéfini et figé.

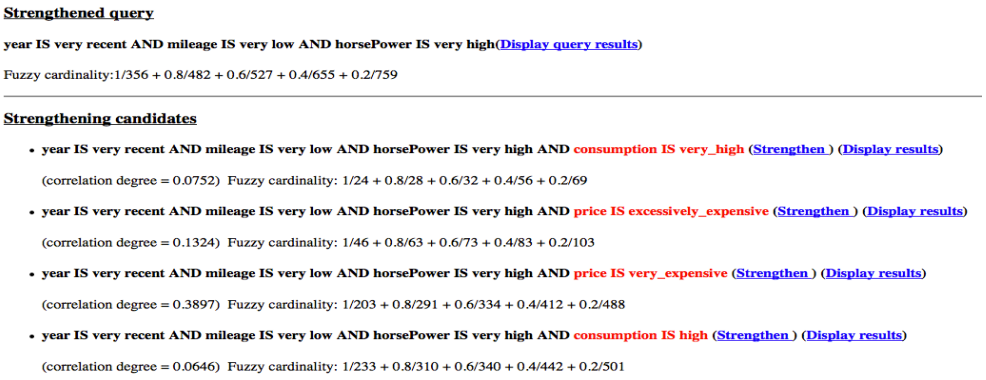

FIGURE 47 – Suggestion de requêtes intensifiées

9.2 Suggérer des sous-requêtes en cas d'échec

Pour aider l'utilisateur à comprendre pourquoi sa requête floue ne retourne pas de résultat, nous avons vu dans la section 8.3 une approche coopérative permettant d'identifier les raisons originelles de la vacuité de l'ensemble résultat. Les informations produites par cette approche permettent à l'utilisateur à la fois de découvrir le contenu de la BD mais également de s'apercevoir de contre-sens

dans sa requête (e.g. voiture dont la cylindrée est 'grosse' et l'émission de CO^2 est 'neutre'). Cependant, une approche coopérative doit également suggérer, en complément des raisons de l'échec, des requêtes dont l'exécution produira un ensemble non vide de résultats et qui soient sémantiquement les plus proches possibles de la requête initiale.

Par rapport aux premiers travaux traitant du problème des requêtes à résultat vide [60, 71], l'originalité de notre approche est qu'elle s'intéresse au cas des requêtes floues. Du fait de la satisfaction graduelle des tuples de la relation interrogée vis-à-vis de la requête initiale, les raisons originelles de l'échec varient selon le degré de satisfaction attendu par l'utilisateur. Il en est de même pour la suggestion de requêtes retournant un ensemble non nul de résultats, où, pour différents seuils de satisfaction, nous indiquons à l'utilisateur les propriétés satisfaites par des tuples de la relation interrogée.

La notion de XSS

Définition Soit une requête Q qui échoue pour un seuil de satisfaction α. Une *requête à résultat non nul* (nommée par la suite XSS pour maXimal Succeeding Subquery) pour un seuil de satisfaction $\alpha' \leq \alpha$ est une sous-requête $Q' \subseteq Q$ telle que $|\Sigma_{Q'}^{\alpha'}| > 0$, et telle qu'il n'existe pas d'autre sous-requête Q'', $Q' \subset Q''$ retournant également un ensemble non nul de résultat au degré α'.

Propriété Une sous-requête Q' d'une requête Q retournant un ensemble résultat vide qui est une XSS à un degré α retourne au moins autant de résultats à un seuil $\alpha' < \alpha$ du fait du caractère convexe des SEFs utilisés. Cependant, une sous-requête Q' peut constituer une XSS maximale à un seuil α sans pour autant être maximale pour un seuil $\alpha' < \alpha$ étant donné qu'une sous-requête $Q'', Q' \subset Q''$ peut retourner des résultats pour un seuil α' et non pour un seuil $\alpha > \alpha'$.

Identification des XSSs

La réparation d'une requête conjonctive Q s'effectue après l'identification des raisons minimales de l'échec pour différents seuils de satisfaction $\alpha_1 = 1 < \alpha_2 < \ldots < \alpha^{0^+}$. Nous avons vu dans la section 8.3 que l'identification des MFS pour différents seuils de satisfaction reposait sur la construction au préalable d'un résumé d'une partie des données, celles concernées un tant soit peu par la requête initiale. L'identification des XSS pour les différents seuils α repose également sur un parcours du treillis des sous-requêtes de Q du haut vers le bas et en largeur d'abord comme l'illustre la figure 48. On commence par examiner les sous-requêtes les plus contraignantes (celles comportant le plus de prédicats) en considérant le degré de satisfaction le plus élevé α_1. Pour déterminer si une sous-requête Q' est une XSS de Q pour un seuil α, il n'est pas nécessaire d'accéder aux résumés des données, i.e. aux cardinalités floues, puisqu'il suffit simplement de vérifier que la sous-requête ne comporte pas de MFS identifiée pour le seuil α concerné. La propriété décrite précédemment sur la monotonie de la non vacuité de l'ensemble résultat d'une requête pour des seuils moins restrictifs est utilisée pour propager une XSS aux autres seuils. Pour propager une XSS à d'autres seuils, il faut tout de même vérifier la propriété de maximalité de la XSS.

Restitution des XSSs

Ainsi, e.g., suite à l'exécution d'une requête floue dont la clause de sélection comporte une conjonction de cinq prédicats flous $Q : P^1 \wedge P^2 \wedge P^3 \wedge P^4 \wedge P^5$ et dont l'ensemble résultat est vide, l'approche coopérative que nous avons construite génère le genre d'informations suivantes :

Aucun élément ne satisfait simultanément A, B, C, D et E pour les raisons suivantes ...
- *aucun élément ne satisfait E,*
- *aucun élément ne satisfait A et D, ainsi que B ou C à un degré* ≥ 0.2,
- *aucun élément ne satisfait A ainsi que B ou C à un degré* ≥ 0.4,
- *aucun élément ne satisfait A à un degré* ≥ 0.6 *;*
- *aucun élément ne satisfait B et C à un degré* ≥ 0.8,

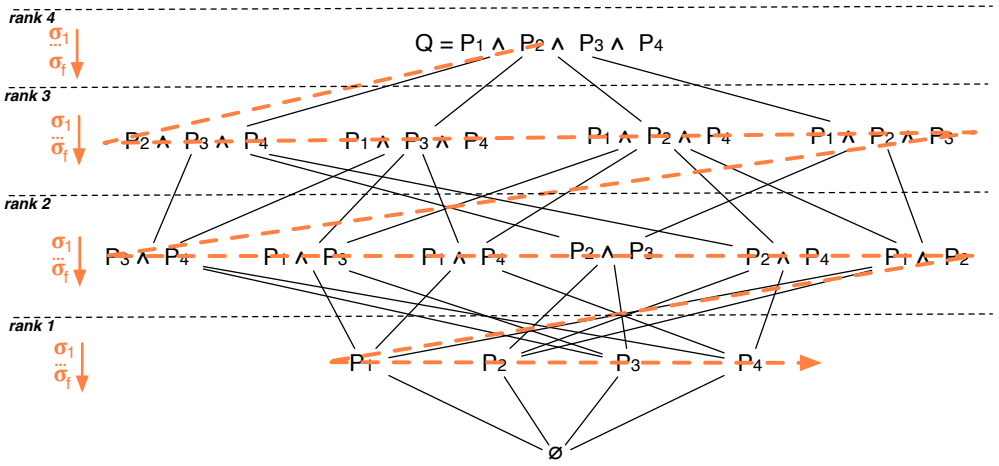

FIGURE 48 – Parcours en profondeur du treillis des sous-requêtes pour identifier les XSSs

— *aucun élément ne satisfait pleinement* B.

... mais la base contient :

— k_1 *éléments satisfaisant pleinement* P^3 *et* P^4,

— k_2 *éléments satisfaisant* P^4, *avec* P^2 *ou* P^3 *à un degré* ≥ 0.8,

— k_3 *éléments satisfaisant* P^2, P^3 *et* P^4 *à un degré* ≥ 0.6,

— k_4 *éléments satisfaisant* P^1, P^2, P^3 *et* P^4 *à un degré* > 0.$\diamond$

Contrairement à un message laconique « Pas de réponse pour votre requête », les informations générées par notre approche indiquent clairement à l'utilisateur ce qu'il peut espérer trouver dans la BD en exploitant les prédicats définis dans sa requête initiale.

Bilan sur l'approche proposée de gestion des requêtes floues retournant un ensemble vide de résultats

L'approche coopérative de gestion du problème des requêtes floues retournant un ensemble résultat vide repose donc sur trois étapes :

1. le résumé de la partie de la BD concernée par la requête,
2. l'identification des raisons minimales de l'échec,
3. et finalement la suggestion de sous-requêtes retournant un ensemble résultat non vide.

Comme l'a montré la figure 42, la première étape de résumé est de loin la plus coûteuse. Les parcours du treillis des sous-requêtes possibles de la requête initiale, de bas en haut pour l'identification des MFSs et de haut en bas pour l'identification des XSSs sont négligeables en temps tant que la requête contient un nombre raisonnable de prédicats (une dizaine au maximum), ce qui est le cas dans la plupart des contextes applicatifs.

Ce travail a constitué la première approche coopérative d'explication de la vacuité du résultat d'une requête floue et de sa réparation. Ce travail a ensuite été réutilisée par des collègues de la communauté BD dans le cadre de données sémantiques [56] et des données incertaines [37].

9.3 Recommandation de réponses indirectes

La stratégie de réponse coopérative la plus connue par les utilisateurs « grand public » et la plus étudiée par les acteurs industriels et universitaires concerne la recommandation de réponses indirectes à une recherche d'information. Principalement intégrées dans des sites de commerce en ligne et dans les portails d'accès aux données, ces stratégies de recommandation visent à suggérer

des objets de la BD interrogée possédant un lien avec les objets directement visés par les critères de recherche exprimés par l'utilisateur [117]. Des objets sont recommandés pour leur similarité avec les caractéristiques des objets retournés par la recherche d'information, on parle alors de recommandation basée sur le contenu [96], ou bien parce que d'autres utilisateurs ont créé des liens entre les objets (par achat ou recherche combinée), on parle alors de recommandation par filtrage collaboratif [68]. Outre l'existence d'approches hybrides exploitant à la fois des similarités entre objets et des indications collaboratives, une autre stratégie consiste à exploiter des connaissances démographiques sur les utilisateurs pour suggérer des contenus pertinents pour la communauté d'appartenance de l'utilisateur [139].

Nous avons étudié cette problématique pouvant être généralisée à la création de liens entre objets d'une BD selon deux axes. Tout d'abord, dans un cadre d'interrogation de BD relationnelles [108] où nous avons proposé une stratégie permettant d'établir des liens de similarité entre tuples à partir des propriétés structurelles du schéma relationnel sous-jacent. L'idée est d'utiliser les contraintes d'intégrité référentielle définies entre les tables pour identifier des liens entre tuples, comme e.g. le fait que deux employés sont reliés au même département. Puis dans le cadre de la thèse d'Aurélien Moreau, thèse financée par la DGA pour le projet ODIN, nous avons proposé une stratégie coopérative visant à exploiter des données démographiques et des relations de typicité pour identifier des réponses indirectes pouvant être pertinentes pour un utilisateur donné [92]. La particularité de ce travail par rapport aux nombreuses approches existantes est d'utiliser des données démographiques et des techniques issues du *soft computing* pour construire des profils de caractéristiques typiques associés aux objets et utilisateurs.

Construction du profil des caractéristiques typiques

L'approche proposée consiste à associer à chaque objet de la base, dans notre cas à chaque film, un vecteur décrivant ses propriétés les plus représentatives. Pour chaque film m, nous identifions les utilisateurs ayant aimé ce film, c'est-à-dire ayant attribué une note de quatre ou cinq étoiles. Puis pour chaque attribut démographique a (e.g. âge, profession, origine, etc.), nous construisons un multi-ensemble des valeurs prises par les utilisateurs sélectionnés pour ce film : $E_c(m) = \{(k/x)\}$, x une valeur de l'attribut c et k le nombre d'utilisateurs sélectionnés possédant cette valeur. Parmi ces différentes valeurs associées à un film, nous identifions les plus représentatives du film, c'est-à-dire celles possédées par le plus d'utilisateurs ayant aimé le film concerné. Nous construisons ainsi un ensemble flou $T_c(m) = \{\mu_T(x)/x\}$ afin de quantifier la représentativité de la valeur x pour le film m, où $\mu_T(x) = \frac{k}{|E|}$, k et le nombre de copies de x dans E. La quantification $\mu_T(x)$ peut être considérée comme un degré de typicité selon la définition statistique de cette notion [146].

Une fois l'ensemble des caractéristiques typiques calculé pour chaque film, nous construisons une matrice de similarité entre films. La construction de cette matrice repose sur l'utilisation d'une mesure permettant de quantifier la similarité entre deux ensembles flous de caractéristiques typiques $T_c(m_i)$ et $T_c(m_j)$, soit à l'aide d'un indice de Jaccard :

$$\mu_{matches}(T_c(m_i), T_c(m_j)) = \frac{\sum_{x \in U} \min(\mu_{T_c(m_i)}(x), \mu_{T_c(m_j)}(x))}{\sum_{x \in U} \max(\mu_{T_c(m_i)}(x), \mu_{T_c(m_j)}(x))},$$

soit à l'aide d'une mesure classique d'égalité entre ensemble flous [22] telle que :

$$\mu_{matches}(T_c(m_i), T_c(m_j)) = \inf_{x \in U} 1 - |\mu_{T_c(m_i)}(x) - \mu_{T_c(m_j)}(x)|.$$

La similarité entre deux films est alors calculée à partir de l'agrégation des degrés de similarité sur les différents attributs pris en compte. Cette agrégation peut être réalisée soit à l'aide de la norme min soit à l'aide d'une mesure pondérée, comme le minimum pondéré [46], pour privilégier certains attributs. Nous avons utilisé la norme min dans nos expérimentations et l'indice de Jaccard pour quantifier la similarité de deux films sur un attribut. Ainsi la cellule d'indice (i, j) de la matrice renseignant sur la similarité entre les films m_i et m_j est calculée comme suit :

$$s_{i,j} = \min_c \mu_{matches}(T_c(m_i), T_c(m_j)).$$

Utilisation des évaluations émises par l'utilisateur

Dans le cas où un utilisateur a déjà émis des évaluations sur certains films, la matrice de similarité est utilisée pour identifier les films dont les caractéristiques typiques sont similaires à celles des films aimés par l'utilisateur. Ainsi le score prédit pour un film m_i dépend de sa similarité vis-à-vis de l'un des films aimés par l'utilisateur u, ces films étant dénotés $LIKE(u)$:

$$p_{u,m_i} = \max_j(s_{i,j}|m_j \in LIKE(u)).$$

Ainsi, partant par exemple du fait qu'un utilisateur a évalué positivement les films *Star Wars IV* et *Tron*, nous pouvons observer que ces films sont entre autres appréciés par les utilisateurs dans les tranches d'age $[18, 24]$ ans et $[25 - 34]$, pour ensuite recommender des films possédant ces traits démographiques typiques, comme e.g. *Star Wars V* ou *Avengers*.

Utilisation des données démographiques de l'utilisateur

Un problème bien connu des systèmes de recommandation est décrit sous le terme de *cold start problem* et correspond au cas d'un nouvel utilisateur du système n'ayant pas encore évalué de film. Dans ce cas, l'utilisation de la matrice de similarité décrite précédemment n'est pas applicable.

Nous avons alors suggéré une seconde stratégie de recommandation basée cette fois-ci sur l'utilisation directe des informations démographiques. Pour ce faire, nous identifions pour chaque caractéristique démographique de l'utilisateur (e.g. tranche d'âge, profession, etc.) l'ensemble flou des films typiquement appréciés par les autres utilisateurs possédant ce trait démographique. Les ensembles de films typiquement appréciés pour chaque trait démographique de l'utilisateur sont agrégés pour identifier ceux qui semblent les plus associés aux données démographiques de l'utilisateur.

Ainsi, si en partant e.g., des caractéristique de *Sophie*, nous observons qu'elle appartient à la même tranche d'age que *Nolan* et *Alice*, occupe la même profession que *Léa* et *Pierre*, et réside dans le même genre d'habitat que *Paul* et *Virginie*, les films suggérés à *Sophie* seront ceux généralement appréciés par Nolan, Alice, Léa, Pierre, Paul et Virginie.

Interprétabilité des recommandations

Une particularité des approches coopératives étudiées lors de la thèse d'A. Moreau réside dans la recherche d'interprétabilité des connaissances indirectes inférées, qu'il s'agisse de recommandation de films, d'explication d'un ensemble de réponses ou d'inférence de requête à partir d'exemples. Une façon d'améliorer les systèmes coopératifs et notamment les stratégies de recommandation est d'accompagner les réponses indirectes d'explications sur leur provenance.

Ainsi, dans le cas où un film est suggéré sur la base des évaluations déjà émises par l'utilisateur, nous générons automatiquement une explication sur la provenance de la recommandation comme celle-ci : « *X-Men was recommended to you because you liked movies such as Harry Potter and LOTR, and all of these movies have the same typical audience who liked them.* ». Pour le second cas utilisant les données démographiques de l'utilisateur, une explication de la forme suivante est générée : « *X-Men was recommended to you because this movie is typically liked by young men in college such as yourself.* ».

Expérimentation

Cette approche de recommandation de réponses indirectes a été implémentée puis testée sur le corpus MovieLens 1M [15] incluant plus d'un million d'évaluations d'environ 3000 films. En prenant 80% de ce corpus pour construire les matrices de similarité et 20% pour tester la prédiction de

15. http ://grouplens.org/datasets/movielens/

score, nous avons comparé la justesse des prédictions fournies par notre approche avec les résultats obtenus par des méthodes classiques de recommandation par filtrage collaboratif [80, 57].

Les résultats obtenus montrent que notre approche n'obtient pas forcément de meilleurs résultats que les approches classiques, seul le rappel est meilleur et par conséquent la F-mesure également. Ceci s'explique par le fait que l'utilisation de données démographiques permet de générer des prédictions même pour des utilisateurs n'ayant pas de profil d'évaluation ou un profil atypique. En effet, lorsque l'on observe les recommandations suggérées uniquement par notre approche, on constate qu'elles sont pertinentes pour des utilisateurs dont le profil d'évaluation des films est atypique. Les expérimentations menées nous ont permis de conclure que l'utilisation de données démographiques pouvait être complémentaire d'approches plus classiques et gérer également le cas problématique du *cold start*.

10 Extraction de connaissances à partir d'un résumé de données brutes

Les approches coopératives décrites dans les sections précédentes permettaient d'enrichir les résultats d'une requête soumise à une BD. Dans la section 3.6, nous avons vu que l'importance prise par les données dans le quotidien de nombreux professionnels avait fait évoluer les besoins en termes d'accès aux données. Un des enjeux majeurs est actuellement de fournir des approches efficaces et pertinentes d'exploration de données brutes et d'extraction de connaissance à partir de ces données [3]. De par sa capacité à personnaliser les processus de représentation et d'exploration de données ainsi que l'interprétabilité des connaissances générées, le *soft computing* a une carte importante à jouer face aux méthodes purement statistiques qui restent à ce jour pourtant les plus utilisées et étudiées.

Ainsi, en complément de l'approche de résumé de grandes quantités de données (Sec. 3.6) et des premiers travaux réalisés sur la représentation de réécritures linguistiques, j'ai récemment (depuis 2015) travaillé à l'enrichissement de ces fonctionnalités d'exploration de données brutes. Ces travaux m'ont permis d'intensifier et de pérenniser les liens initiés avec les entreprises Semsoft et DCBrain, et notamment d'obtenir un financement pour une thèse CIFRE (étudiant Toan Ngoc Duong) et de monter un nouveau projet RAPID-DGA sur l'utilisation de ces méthodes pour l'analyse de journaux d'évènements réseau et système (projet en cours d'évaluation par la DGA).

10.1 La réécriture linguistique comme point de départ à l'extraction de connaissances

À partir d'un jeu de données brutes $\mathcal{D}$ décrivant des objets sur un ensemble d'attributs $\mathcal{A}$ et d'un vocabulaire utilisateur $\mathcal{V}$, nous avons proposé un processus distribué de réécriture des données (Sec. 6.1) permettant de construire une représentation linguistique, personnalisée et synthétique des données. Le processus de réécriture génère un vecteur de réécriture indiquant la distribution des données vis-à-vis des termes du vocabulaire :

$$RV_d^{\mathcal{V}}, d \in \mathcal{D} : \langle \rho_{v_1^{A_1}}(\mathcal{D}), \ldots, \rho_{v_{q_1}^{A_1}}(\mathcal{D}), \ldots, \rho_{v_1^{A_n}}(\mathcal{D}), \ldots, \rho_{v_{q_n}^{A_n}}(\mathcal{D}) \rangle.$$

Une restitution graphique d'un tel vecteur de réécriture a été suggérée dans la section 3.6 sous forme de nuage de termes. Outre le fait que la réécriture linguistique permet une représentation unifiée et concise de données décrites sur plusieurs dimensions numériques et catégorielles, nous avons vu que des fonctionnalités d'exploration des données pouvaient être construites à partir de ce résumé des données (Sec. 3.6). Ainsi, en exploitant notamment des stratégies dédiées de stockage et d'indexation (Sec. 6.3), la sélection d'un terme dans ce nuage permet de récupérer les objets satisfaisant ce terme et de construire le vecteur de réécriture de ce sous-ensemble $s \subseteq \mathcal{D}$ d'objets $RV_s^{\mathcal{V}}$. Le vecteur de réécriture du sous-ensemble s est à son tour restitué sous forme d'un nuage de termes permettant de visualiser rapidement les propriétés possédées par les objets satisfaisant le terme sélectionné dans le nuage précédent. Les figures 49 et 50 illustrent respectivement le nuage

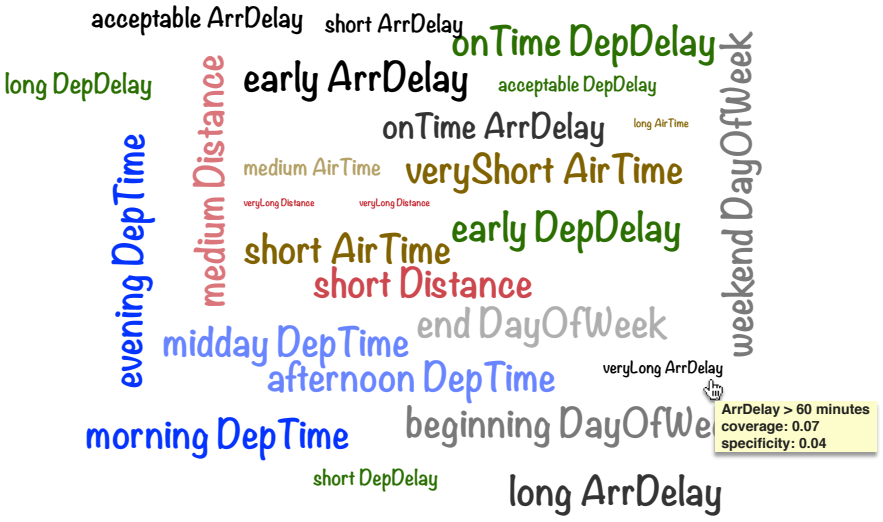

FIGURE 49 – Nuage de termes représentant l'ensemble des vols d'avions du jeu de données $\mathcal{D}$

de termes d'un jeu de données complet décrivant des vols d'avion et le nuage de termes concernant uniquement les vols atterrissant avec un retard très important (*veryLong ArrDelay*).

FIGURE 50 – Nuage de termes représentant uniquement les vols retardés

Afin d'enrichir la restitution graphique des objets satisfaisant un terme sélectionné, nous (R.R. Yager, O. Pivert et moi-même) avons travaillé au développement de stratégies d'extraction de connaissances additionnelles en utilisant uniquement les deux vecteurs de réécriture $RV_{\mathcal{D}}^{\mathcal{V}}$ et $RV_{s}^{\mathcal{V}}$. L'objectif de ces stratégies est d'aider l'utilisateur à mieux comprendre le terme sélectionné sans avoir recours à un nouvel accès, potentiellement coûteux, aux données. Plus globalement, ce travail nous a permis d'initier un nouvel axe de recherche concernant l'extraction de connaissances à partir d'un résumé linguistique personnalisé d'un jeu de données (Sec. 13).

10.2 Quantifier le caractère informatif d'un descripteur linguistique

La première représentation graphique du vecteur de réécriture d'un ensemble de données que nous avons proposée dispose, à la manière des nuages de mots apparaissant sur le web, les termes

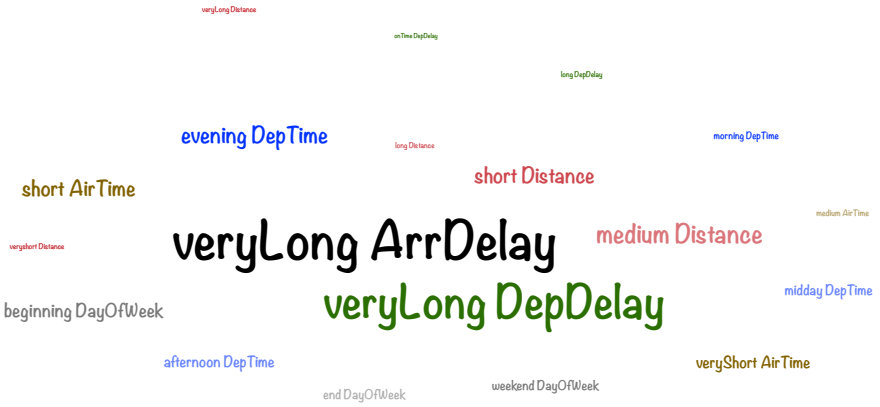

FIGURE 51 – Nuage de termes structuré par une mesure de spécificité

de façon aléatoire [66] (Fig. 50 et 49). Il n'y a donc aucune raison pour qu'un terme se retrouve au centre de la visualisation ou bien à côté d'un autre terme.

Afin d'aider l'utilisateur à mieux comprendre les propriétés qui caractérisent un ensemble de données $s \subseteq \mathcal{D}$, nous avons défini une mesure quantifiant la représentativité de chaque terme apparaissant dans $RV_s^{\mathcal{V}}$ par rapport à s.

Cette quantification de la représentativité s'inspire de la mesure de spécificité introduite par R.R. Yager dans [141] pour appréhender le degré d'imprécision impliqué dans une assertion linguistique incluant un SEF. Par exemple, l'imprécision liée à l'assertion « ce temps de vol est long » dépend de la définition de la fonction d'appartenance associée à l'adjectif *long*, i.e. plus le domaine de définition couvert par la fonction est grand plus grande est l'imprécision. Cette notion a donc été étendue à notre contexte pour quantifier la représentativité d'un terme v vis-à-vis de l'ensemble des données s.

Un descripteur v est considéré comme représentatif d'un ensemble s si, sur la dimension concernée, il constitue la seul description possible, reprenant ainsi la propriété de spécificité maximale pour un ensemble singleton [85].
Cette mesure est un peu plus sophistiquée que cela car elle permet également de discriminer plusieurs descripteurs d'une même partition qui couvrent s. L'idée de cette mesure est de favoriser les descripteurs adjacents dans la partition qui couvrent majoritairement l'ensemble s. La mesure notée $Sp(v, s)$ combine donc deux notions : la couverture de s par v ($cover(v, s)$) et la distance de v par rapport aux autres descriptions possibles sur la même dimension ($d(v, s)$).

Une visualisation plus sophistiquée du vecteur de réécriture $\mathcal{RV}_s^{\mathcal{V}}$ et exploitant cette mesure de représentativité a été suggérée dans [132] afin de mettre en exergue les termes les plus informatifs. Cette représentation graphique sous forme de spirale consiste à placer au centre de la représentation, la zone la plus visible [66], les termes les plus informatifs selon la mesure R. Les figures 50 et 51 synthétisent le même ensemble de données, la première utilise un nuage de termes dont la structure est générée aléatoirement et la seconde dispose les termes selon leur degré de spécificité. On constate que cette dernière représentation permet d'identifier plus facilement les termes représentant les vols retardés.

La quantification de l'informativité de chaque terme apparaissant dans un vecteur de réécriture, ainsi que la génération de la visualisation graphique se font en temps très court car ces processus reposent sur un parcours séquentiel d'un vecteur très petit (quelques dizaines d'éléments occupant quelques kilo-octets en mémoire).

10.3 Enrichir le processus d'exploration des données

En complément de cette structuration du nuage de termes pour mettre en exergue les termes les plus représentatifs d'un vecteur, nous avons défini des méthodes permettant de mieux comprendre le contexte d'un terme sélectionné, disons v. Ces méthodes peuvent être appliquées une fois un terme sélectionné et le vecteur de réécriture des données satisfaisant ce terme généré. Trois types de connaissances sont extraits à partir de $\mathcal{RV}_{\mathcal{D}}^{\mathcal{V}}$ et $\mathcal{RV}_{s}^{\mathcal{V}}$ pour expliquer à l'utilisateur :

— les termes de $\mathcal{RV}_{s}^{\mathcal{V}}$ corrélés au terme sélectionné v,
— les termes « surprenants » de $\mathcal{RV}_{s}^{\mathcal{V}}$,
— et les dimensions sur lesquelles les réécritures des objets de $\mathcal{RV}_{s}^{\mathcal{V}}$ sont les plus diversifiées.

Par l'identification de propriétés corrélées

Pour un terme v sélectionné dans un nuage, v peut correspondre à une conjonction de termes suite à des sélections successives de termes, nous identifions dans $RV_{s}^{\mathcal{V}}$ ($s = \{d \in \mathcal{D}, \mu_v(d) \geq \alpha$, où α est un seuil de satisfaction défini par l'utilisateur) les termes $v' \neq v$ corrélés à v [39, 38]. Pour quantifier la dépendance d'un terme v' par rapport à v, nous comparons la proportion d'objets de $\mathcal{D}$ et de s satisfaisant v'. La mesure suivante nous permet donc d'obtenir, en utilisant uniquement les informations présentes dans $\mathcal{RV}_{\mathcal{D}}^{\mathcal{V}}$ et $\mathcal{RV}_{s}^{\mathcal{V}}$ le degré de lift des règles d'association $v \rightarrow v'$:

$$dep(v, v') = \frac{cover(v', \mathcal{D}_v)}{cover(v', \mathcal{D})}.$$

Ce degré étant défini dans l'intervalle $[-1, 1]$, nous procédons ensuite à sa normalisation dans l'intervalle unité :

$$assoc(v, v') = \begin{cases} 0 & \text{if } dep(v, v') \leq 1, \\ 1 - \frac{1}{dep(v, v')} & \text{otherwise.} \end{cases}$$

Pour restituer à l'utilisateur ces connaissances permettant de mieux comprendre les propriétés du terme v sélectionné, nous utilisons la représentation graphique illustrée par la figure 52 où la longueur de la flèche reliant v à v' dépend du degré d'association $assoc(v, v')$. Cette représentation permet notamment de comprendre les raisons d'un phénomène, comme le retard des avions (e.g. départ le soir), ou bien d'identifier les traits caractéristiques d'un objet, par exemple des voitures polluantes (e.g. moteur puissant). De même, en exploitant uniquement les deux vecteurs de réécriture, ces connaissances sont générées en temps très court, quelques milli-secondes.

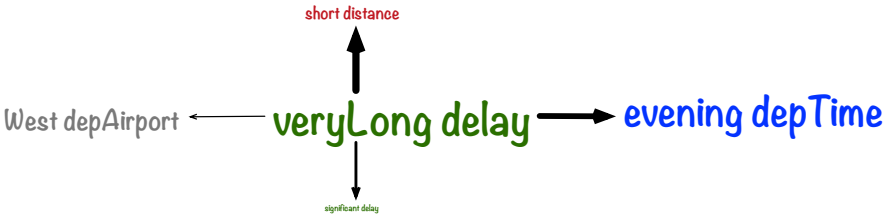

FIGURE 52 – Affichage des termes corrélés à un terme sélectionné, ici *'veryLong ArrDelay'*

Par l'identification de propriétés atypiques

Alors que les règles d'association expliquent des généralités entre termes, il peut parfois être également très informatif pour un utilisateur d'identifier des propriétés assez surprenantes dans un vecteur de réécriture $\mathcal{RV}_{s}^{\mathcal{V}}$ contenant les objets satisfaisant un terme sélectionné v. Un terme v' peut constituer une propriété surprenante, i.e. atypique, s'il couvre un sous-ensemble de s de faible cardinalité et s'il s'éloigne des termes couvrant majoritairement s sur la même dimension. Ainsi pour quantifier le degré d'atypicité d'un terme dans un vecteur, deux mesures sont combinées : 1) la couverture de s par v_{ij} et 2) la distance de v_{ij} vis-à-vis des autres termes couvrant s sur la

même dimension A_i.

La distance entre v_{ij} et les termes principaux couvrant s sur A_i est quantifiée de la façon suivante :

$$D(v_{ij}, s) = \max_{v_{ik} \in RV_s} \min(cover(v_{ik}, s), d(v_{ij}, v_{ik})),$$

où $d(v_{ij}, v_{ik}) = \frac{|j-k|}{q_i - 1}$ si A_i est un attribut numérique, $d(v_{ij}, v_{ik}) = 1$ if $v_{ij} \neq v_{ik}$, 0 si A_i est symbolique.

Le terme v_{ij} apparaît de manière surprenante dans le vecteur $RV_s^{\mathcal{V}}$ s'il couvre uniquement quelques objets de s alors qu'il n'était pas particulièrement rare dans $\mathcal{D}$. Cette couverture surprenante est quantifiée de la façon suivante :

$$NC(v_{ij}, \mathcal{D}_v) = 1 - min(1, \frac{cover(v_{ij}, \mathcal{D}_v)}{cover(v_{ij}, \mathcal{D})}).$$

Finalement, le degré d'atypicité combine les mesures D et NC à l'aide de la norme min :

$$atyp(v_{ij}, \mathcal{D}_v) = \min(D(v_{ij}, \mathcal{D}_v), NC(v_{ij}, \mathcal{D}_v)).$$

La figure 53 suggère une stratégie de restitution graphique des termes apparaissant de manière surprenante dans un vecteur de réécriture.

FIGURE 53 – Affichage de propriétés surprenantes pour les vols arrivant avec un long retard

Par l'identification de dimensions offrant un maximum de diversité

Alors que les deux premières stratégies d'extraction de connaissances à partir des vecteurs de réécriture mettent en relation des termes du vocabulaire, la troisième stratégie développée consiste à identifier les dimensions offrant la plus grande diversité de termes. Cette fonctionnalité peut être notamment très utile pour explorer des données décrivant des objets dans un contexte commercial. Si on reprend le cas des voitures d'occasion et que l'on sélectionne comme terme de départ la propriété *prix 'bas'*, il peut apparaître intéressant de savoir que dans cette gamme de prix, on dispose d'un choix large de couleurs de voiture ou de marques mais pas sur le kilométrage, ni l'année ou le niveau d'options.

Nous cherchons donc à quantifier la diversité d'un ensemble flou, désignons le par X, contenant l'ensemble des termes sur une dimension donnée où chaque terme est associé à son degré de couverture des objets de l'ensemble s. Pour quantifier cette diversité, qui doit être maximale si tous les termes d'une même dimension couvrent de manière égale un ensemble s, nous utilisons simplement une mesure d'écart type :

$$div(X, s) = 1 - \sqrt{\frac{\sum_{v \in X}(cover(v, s) - \overline{cover(X, s)})^2}{|X|}},$$

où $\overline{cover(X, s)}$ est la couverture moyenne des termes dans X.

La figure 54 suggère une représentation possible des dimensions identifiées comme offrant la plus grande diversité.

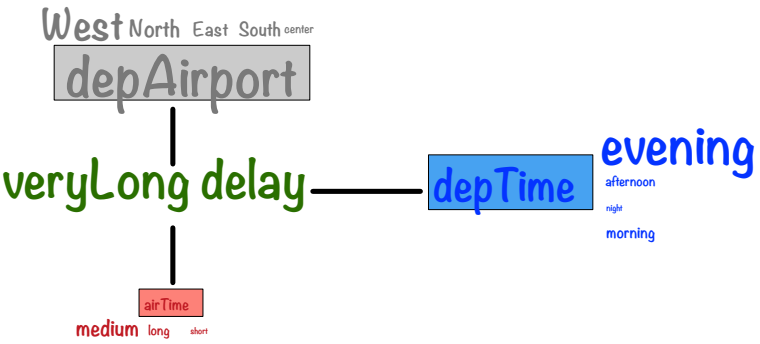

FIGURE 54 – Affichage de dimensions offrant une grande diversité de termes par rapport au terme sélectionné *veryLong ArrDelay*

11 Bilan sur l'enrichissement des résultats retournés par les méthodes d'accès aux données

Le début de ce chapitre était consacré aux approches coopératives permettant d'enrichir les résultats retournés par l'exécution d'une requête. Ces travaux s'insèrent dans la continuité des approches coopératives définies par P. Bosc, O. Pivert et A. Hadjali avant mon arrivée dans l'équipe PILGRIM → SHAMAN. Cependant, venant d'un domaine de recherche différent, le TAL, je pense avoir apporté un regard nouveau sur ces problématiques et proposé des solutions pragmatiques et novatrices, comme e.g. l'intensification de requête par prédicats corrélés et la recherche de transparence dans les stratégies de recommandation de réponses indirectes. Le travail réalisé avec A. Moreau dans le cadre de sa thèse, dont la soutenance aura lieu en décembre 2017, autour de l'explication de réponses et la recommandation, seront poursuivis lors d'un postdoc qui débutera en octobre 2017. L'objectif de ce postdoc est de transposer ces stratégies coopératives définies pour des données relationnelles vers des données sémantiques en cherchant à tirer profit de la richesse du modèle sémantique sous-jacent.

La dernière partie du document est très importante dans ce bilan car elle ouvre sur des perspectives de recherche à moyen terme qui seront détaillées dans la section 13. Ces perspectives concernent l'extraction de connaissances à partir de résumés linguistiques et personnalisés de données. Une thèse CIFRE a débuté en novembre 2016 sur ce sujet visant à établir des stratégies de comparaison de résumés de données décrivant différentes plages de temps ou différentes catégories (e.g. comparaison des résumés des restaurants à Paris et à Londres).

Ces travaux m'ont également permis de nouer des contacts très enrichissants avec R.R. Yager, directeur du *Machine Intelligence Institute* à l'IONA college (NY-USA) et pionnier de la logique floue. Nous avons entamé une collaboration intéressante et fructueuse sur l'axe de recherche qui m'occupe actuellement le plus : l'enrichissement des accès aux données brutes dans un contexte applicatif de *business intelligence*. Un projet RAPID-DGA vient d'être déposé pour me permettre de compléter ces approches d'exploration et d'extraction de connaissances à partir de données brutes dans un contexte d'analyse de logs.

12 Bilan des activités de recherche [2009-2017] et conclusion

L'écriture de ce document a été pour moi l'occasion d'effectuer un travail de synthèse des travaux que j'ai réalisés depuis la transition thématique opérée en 2009 du TAL aux BD. Le dénominateur commun des contributions mentionnées dans ce document est celui de l'enrichissement des méthodes d'accès aux données par la prise en compte, en tous points du processus, des particularités de l'utilisateur final. L'accès aux données y est décrit comme un processus cyclique partant de l'expression d'un besoin d'information exprimé par un utilisateur, puis reposant sur une étape centrale de confrontation de ce besoin vis-à-vis des données interrogées, et bouclant sur une restitution à l'utilisateur des résultats trouvés.

La première partie du document positionne mes contributions sur l'enrichissement des méthodes et langages d'interrogation de données dans la continuité des travaux réalisés précédemment par mes collègues de l'équipe SHAMAN. Après des collaborations très enrichissantes avec P. Bosc et O. Pivert sur l'expressivité du langage SQLf, j'ai initié une démarche importante visant à prouver, par la pratique, que des systèmes d'interrogation floue pouvaient être implémentés et répondaient à un besoin actuel majeur autour de la personnalisation des interfaces d'accès aux données. Ces travaux m'ont à la fois permis de me faire connaître auprès de la communauté des chercheurs travaillant sur l'interrogation floue de BDs, mais également d'accroître la visibilité de l'équipe SHAMAN à travers une démonstration logicielle lors de la conférence VLDB. De manière plus indépendante, je me suis ensuite intéressé aux méthodes d'interrogation par mots-clés de données structurées en cherchant à réutiliser les compétences acquises lors de mon doctorat sur l'exploitation de ressources linguistiques. Sollicité par des acteurs industriels, en l'occurence Semsoft et la Direction Générale des Armements, j'ai ensuite abordé la problématique de l'enrichissement et la personnalisation des méthodes d'exploration de données non structurées. Au cours de ces différents travaux sur la personnalisation des interfaces d'accès aux données, j'ai également abordé, d'un point de vue plus global, la question de la formalisation d'un vocabulaire utilisateur dédié à un jeu de données. Initialement réservée aux professionnels de l'informatique, l'interrogation de données structurées et non structurées est devenue une activité cruciale dans le quotidien de nombreux professionnels et particuliers. La visibilité de mes contributions pour personnaliser et rentre plus accessibles et intuitives les interfaces d'interrogation de données devraient perdurer et me permettre d'établir des liens avec des acteurs professionnels en vue d'effectuer des transferts technologiques.

L'enrichissement des interfaces d'interrogation ne peut pas se faire au détriment de l'efficacité des accès aux données. Les travaux présentés dans la seconde partie du document montrent clairement qu'un compromis intéressant peut être trouvé entre enrichissement et efficacité. En effet, le calcul *a priori* de méta-données et des techniques dédiées de stockage et d'indexation peuvent garantir un accès personnalisé et efficace à de grandes quantités de données.

Alors que les communautés des BD et de la gestion des connaissances se sont focalisées lors des dix dernières années sur le stockage et l'accès à de grandes quantités de données, l'enjeu actuel concerne désormais la valeur ajoutée que l'on peut extraire des données et des résultats produits par les interfaces d'interrogation. Si l'on regarde les publications des grandes conférences de BD (e.g. VLDB, PODS, etc.), on constate un net regain d'intérêt pour le développement de stratégies visant à rendre plus interprétables et transparents les processus de traitement des données. Ainsi, l'équipe SHAMAN et derrière elle la communauté du *soft computing* ont été des précurseurs en cherchant à enrichir à la fois les méthodes d'accès et les résultats des interactions entre les utilisateurs et les données. En ce sens, les stratégies de réponse coopérative décrites dans la troisième partie de ce document répondent à un enjeu actuel majeur, celui de rendre le plus interprétable possible les résultats générés par les systèmes d'accès aux données.

En constatant que les attentes des utilisateurs de systèmes de gestion de données et, par conséquent, les axes scientifiques identifiés comme cruciaux peuvent grandement bénéficier des aouts du *soft computing*, j'aborde les prochaines années de recherche avec beaucoup d'enthousiasme. Je suis en effet convaincu que, moyennant un effort concentré sur les expérimentations, nous pouvons convaincre la communauté des BD que le *soft computing* constitue une réponse originale et crédible

aux enjeux actuels en gestion de grandes quantités de données.

13 Perspectives de recherche

La synthèse de mes contributions présentée dans ce document a comme premier objectif de permettre un regard critique sur les travaux de recherche réalisés depuis 2009. Elle m'a également permis de prioriser les différentes perspectives de recherche que j'ai pu identifier lors de collaborations fructueuses avec mes collègues de l'IRISA ou d'autres laboratoires comme le LIP6.

Améliorer la visibilité de nos approches d'interrogation floue de BD

À travers la définition d'une algèbre d'interrogation floue, d'un langage d'interrogation (SQLf) et l'implémentation de modules d'interrogation floue intégrés dans les principaux SGBD commerciaux (PostgreSQL et Neo4J), l'équipe SHAMAN peut être considérée comme un des acteurs scientifiques majeurs de l'interrogation personnalisée de données structurées. Nous recevons un nombre important de sollicitations, de doctorants principalement mais également de chercheurs et d'industriels, pour documenter et enrichir nos solutions d'interrogation floue. J'ai récemment pris contact avec la Société Accélératrice de Transfert Technologique pour obtenir un financement afin d'effectuer une maturation logicielle du module PostgreSQLf. L'idée est de capitaliser sur les expérimentations menées à l'aide de PostgreSQLf et d'améliorer la visibilité du langage SQLf et de l'interface ReqFlex en développant une interface d'interrogation floue de sources de données indépendante du système sous-jacent. Ceci repose sur une implémentation efficace et l'intégration d'une stratégie de dérivation au sein de l'interface graphique d'interrogation afin d'exploiter les performances intrinsèques des systèmes de gestion de données.

Dans un second temps, nous enrichirons cette interface d'interrogation par les différentes stratégies de réponse coopérative que nous avons conçues, en fournissant :

- des explications sur la structure intrinsèque des tuples retournés en résultat d'une requête floue,
- les raisons de la vacuité d'un résultat et des suggestions de correction,
- des stratégies d'aide à la construction de requêtes à base d'exemple (approche en cours de publication),
- des suggestions de réponses indirectes.

Construire des chaînes de traitement de données brutes orientées utilisateur

Fournir des fonctionnalités pragmatiques et efficaces d'analyse de données brutes est un enjeu actuel majeur auquel les communautés des BD et de la gestion de données doivent répondre. En poursuivant des expérimentations sur la réécriture linguistique et personnalisée de grandes quantités de données, nous pouvons fournir des chaînes d'analyse de données brutes efficaces, et dont les connaissances qu'un utilisateur peut en extraire sont nettement plus interprétables que les résultats numériques fournis classiquement.

À travers une thèse CIFRE financée par l'entreprise Semsoft, nous construisons actuellement des outils d'exploration flexible et interactive de données brutes dont la caractéristique est d'utiliser uniquement une représentation linguistique personnalisée des données et des connaissances extraites. Ceci s'inscrit donc parfaitement dans le paradigme suggéré par LA. Zadeh en 1996, celui de calculer avec des mots [147], paradigme qui guide de manière délibérée mes propres travaux. Afin d'ajuster ce paradigme aux besoins d'utilisateurs réels, nous allons, par le biais de Semsoft et de ses clients, disposer de précieux retours d'expérience sur l'utilisation, en milieu réel, de notre stratégie d'extraction de connaissances à partir de résumés de données.

Enrichir la restitution des connaissances produites

La restitution graphique de connaissances à un utilisateur final est une problématique à part entière dont la résolution repose sur des compétences en ergonomie et des considérations artistiques. Dans le cadre d'un projet DGA (dispositif RAPID) en cours de montage, nous allons collaborer avec l'équipe Ingénierie de l'Interaction Homme-Machine (IIHM) du Laboratoire d'Informatique de Grenoble (LIG) pour fournir à des experts en sécurité informatique de la DGA, des fonctionnalités graphiques d'exploration et d'analyse de logs système et réseau. À travers cette collaboration, nous espérons acquérir quelques compétences sur la construction d'interfaces intuitives d'interrogation floue.

Ce projet nous offrira également l'opportunité de montrer que le *soft computing* peut apporter des réponses pragmatiques et efficaces aux problématiques très techniques et spécialisées liées à la cyber-sécurité et plus particulièrement l'analyse de journaux système.

Exploiter la complémentarité des techniques de fouille de données, d'interrogation de BD, de *soft computing* et de visualisation de connaissances

De manière plus globale, mes perspectives de recherche à plus long terme concernent la volonté de créer une intersection entre des domaines de recherche trop cloisonnés malgré leur complémentarité certaine. En effet, dans le but d'aboutir à des systèmes personnalisés et coopératifs d'exploration et d'analyse de données, il apparaît indispensable de combiner des contributions existantes issues des domaines de la fouille de données massives, de l'interrogation de BD, du *soft computing* et de la visualisation de connaissances.

Les premiers travaux réalisés sur le résumé linguistique de données brutes et la définition de fonctionnalités d'exploration et d'extraction de connaissances à partir de ce résumé (Sec. 3.6 et 10) ont clairement mis en exergue l'intérêt d'approches multi-disciplinaires. Nous avons en effet vu que des fonctionnalités d'exploration de données, fonctionnalités inspirées par des travaux antérieurs sur l'interrogation floue de BD d'une part et les approches de réponse coopérative d'autre part, permettaient d'extraire des connaissances, telles que des règles d'association, de manière interactive. L'interactivité permet d'intégrer l'utilisateur final dans le processus de découverte de connaissances et ainsi d'aboutir à des résultats plus interprétables, notamment car ils résultent d'une démarche d'exploration conduite par l'utilisateur lui-même.

Dans l'optique de concevoir des stratégies pragmatiques et efficaces d'analyse de données massives, je vais poursuivre mes travaux sur le développement de méthodes flexibles d'interrogation pour exploiter les résultats de méthodes automatiques d'analyse de données, tout en veillant à ce que les résultats et connaissances extraites puissent être restitués à l'utilisateur de manière la plus interprétable possible. Je travaille notamment actuellement sur des méthodes d'exploration interactive et personnalisée de la structure intrinsèque d'un jeu de données, structure étant construire automatiquement par un algorithme de clustering. En parallèle, des réflexions sont également menées sur la façon dont cette structure doit être restituée graphiquement à l'utilisateur afin de s'assurer qu'il dispose d'une visualisation complète et exploitable des données et de leur structure.

En conclusion, l'objectif à plus long terme de mes travaux est de tendre vers un compromis intelligent entre la capacité des méthodes d'analyse à traiter de grandes quantités de données et la volonté de générer des connaissances pertinentes et interprétables qui permettent à l'utilisateur d'accélérer la transformation de données en connaissances.

Quatrième partie
Annexes

A Projets de recherche

Le montage de dossiers en vue d'obtenir des financements occupe, pas forcément par choix, une part important de mes activités professionnelles. En établissant des partenariats avec des entreprises, j'ai pu obtenir des financements me permettant d'encadrer dans de bonnes conditions financières les doctorants que j'ai supervisés. En effet, les bourses régionales et ministérielles prennent en charge les salaires des doctorants mais pas le financement de leur matériel, de formations complémentaires (e.g. les écoles d'été), les déplacements aux conférences, l'organisation de la soutenance, etc. À part la thèse d'Amira Essaid qui s'inscrivait dans le cadre d'une co-tutelle avec l'université de Tunis, Aurélien Moreau et Toan Duong Ngoc disposent d'un budget confortable pour effectuer leur thèse dans de bonnes conditions. Ces deux thèses sont réalisées respectivement dans le cadre d'un projet DGA et d'un montage CIFRE avec SEMSOFT.

Projet PostgreSQLf

Afin de traduire le langage SQLf et son algèbre associée en outil opérationnel, j'ai répondu à un appel à projets lancé en 2012 par l'association francophone PostgreSQL. Dans le cadre de la sortie de la version 9.0 du SGBD PostgreSQL, cet appel à projets avait pour but de motiver des développements sur le nouveau gestionnaire de modules nommé PGXN. Le projet proposé pour implémenter un module d'interrogation floue au-dessus de PostgreSQL a été sélectionné et m'a permis d'obtenir un financement pour embaucher un ingénieur de recherche, Thomas Girault (3 mois) et un stagiaire de DUT, Kevin Pensec (10 semaines). Outre le fait d'avoir abouti à un module opérationnel, ce travail d'implémentation a également donné lieu à d'intéressantes publications sur les stratégies d'implémentation de fonctionnalités d'interrogation floue au-dessus d'un SGBD relationnel.

Contrat de recherche sur l'interrogation par mots-clés

En 2013, j'ai obtenu un financement sous forme de contrat de recherche externe de la part de l'entreprise Semsoft pour travailler sur la problématique de l'interrogation par mots-clés enrichis de données sémantiques. J'ai fourni à l'entreprise un document sur l'état de l'art des stratégies d'interrogation par mots-clés de données structurées qui a servi de base de travail à l'élaboration d'une stratégie novatrice d'interrogation par mots-clés (voir projets IntelSearch et 360Predict).

Le projet IntelSearch

Suite à des premiers contacts initiés par ma collègue Hélène Jaudoin avec l'entreprise Semsoft, j'ai pu participer au montage et la réalisation d'un projet PME financé par le pôle de compétitivité Images et Réseaux (2014-2015). Dans le cadre de ce projet, j'ai initié des travaux sur l'enrichissement des méthodes d'interrogation par mots-clés de données structurées. J'ai pu utiliser le budget de ce projet pour financer la valorisation scientifique de mes premières contributions autour de l'interrogation par mots-clés de données structurées [130, 131] et initier un transfert technologique vers Semsoft à travers l'outil Aggrego Search.

Le projet 360Predict

Suite au travail de veille scientifique autour des stratégies d'interrogation par mots-clés de données structurées (CRE) et au développement de l'approche Aggrego Search, j'ai été sollicité par l'entreprise Semsoft pour prendre part à un projet PME financé par le pôle de compétitivité Images et Réseaux (2015-2016). Un consortium composé de deux PMEs, Semsoft et Predicsis, et du laboratoire de recherche auquel j'appartiens, l'IRISA, a ainsi été monté pour interfacer les fonctionnalités de médiation développées par Semsoft et les fonctionnalités d'analyse prédictive

fournies par Predicsis. Mon rôle dans ce projet était de définir des méthodes intuitives et expressives d'accès aux données sémantiques. J'ai obtenu un financement pour un postdoc d'un an ainsi que pour assurer le déplacement à 3 conférences internationales. C'est au cours de ce projet que j'ai supervisé Khadim Dramé pour la construction d'une approche d'interrogation par mots-clés qui soit à la fois expressive, interactive et efficace, ainsi que l'implémentation du prototype iKeys (Sec. B).

Le projet Open Data INtelligence (ODIN)

Le projet ODIN est financé par la Direction Générale de l'Armement à travers le dispositif RAPID. Débuté en 2014 et pour une durée de quatre années, ce projet vise à construire une chaîne de traitement partant de données stockées dans des sources distribuées et visant à enrichir un entrepôt sémantique sur lequel des analyses dimensionnelles sont effectuées. Pour construire cette stratégie novatrice de traitement et de valorisation de données hétérogènes, un partenariat a été créé entre l'entreprise SEMSOFT, le LRI-INRIA Saclay et l'IRISA-Université de Rennes 1.

Dans le cadre de ce projet, je supervise la thèse d'Aurélien Moreau (2014-2017) ainsi que le postdoc d'Ahmed Abi (octobre 2017-octobre 2018).

Projets en cours d'évaluation

Le projet ODIN arrivant à sa fin, j'ai passé un temps non négligeable en 2017 pour assurer la continuité des financement de mes travaux. Au cours des échanges réalisés avec la DGA pour le suivi du projet ODIN, nous avons été sollicités pour travailler sur la problématique de l'exploration de grandes masses de données non structurées dans un contexte de cyber-sécurité. Ce projet, si il est financé, sera réalisé en collaboration avec Semsoft et l'équipe IIHM du Laboratoire d'Informatique de Grenoble afin de disposer de compétences sur la restitution graphique de représentations linguistiques personnalisées. Un financement de thèse dont je serai peut-être pour la première fois directeur est demandé.

Depuis deux ans, j'ai la chance de travailler avec R.R. Yager, un pilier du *soft computing* mondial et nous essayons d'avoir des contacts réguliers pour avancer sur des problématiques liées à l'analyse de données à l'aide de descripteurs linguistiques subjectifs. Afin d'apprendre encore davantage de choses à ses côtés, je viens de déposer une demande de bourse européenne Marie Sklodowska-Curie qui me permettrait de faire de la recherche à 100% à ses côtés pendant un an.

Finalement, afin de diversifier nos actions et nos collaborations, j'ai initié des contacts avec l'entreprise DCBrain, entreprise spécialisée dans la modélisation de graphes de flux à partir de données brutes. L'objectif est de monter un partenariat, soit sous forme de thèse CIFRE ou de projet PME, pour travailler sur la prise en compte de données de qualité lors de la modélisation des graphes et de leur exploration.

B Prototypes de recherche

Bien que leur implémentation ne soit pas une tâche aisée, les prototypes de recherche constituent des éléments importants pour valoriser et améliorer la visibilité de nos travaux de recherche. Convaincu que nos approches d'interrogation floue et de réponse coopérative répondent à des besoins concrets en manipulation de données, j'ai consacré un temps non négligeable à l'implémentation de prototypes de recherche. Ces prototypes ont principalement été utilisés pour mener des expérimentations mais également pour communiquer lors de session démonstration dans des conférences. Pour le moment, les prototypes décrits dans cette section sont relativement indépendants. Nous allons cependant bénéficier d'un support ingénieur à raison d'une journée par semaine à partir d'octobre 2017 afin de développer une plateforme unifiée de démonstration.

PostgreSQLf + Reqflex

Ce travail d'implémentation est sans doute le plus important, en termes de temps consacré et de retombés scientifiques. Ce travail a pu débuter grâce à l'obtention d'un financement auprès de l'association francophone PostgreSQL. J'ai ainsi pu recruter un ingénieur de recherche, Thomas Girault pour m'aider dans le développement de PostgreSQLf (Fig. 55) et Kevin Pensec, pour son stage de DUT informatique, afin de compléter l'aspect graphique de ReqFlex (Fig. 56). Ces deux prototypes sont accessibles via la plateforme de partage de code OpenSource GitHub mais des travaux sont en cours pour publier une version améliorée de ce prototype (documentation en anglais et intégration d'une étape de dérivation lors des opérations de sélection).

```
reqflexdb=# select create_predicate('elevee', 9, 13, 22, 25);
 create_predicate
------------------

(1 row)

reqflexdb=# select *, get_mu() as mu
from employees
where anciennete ~= 'elevee';
 ide |  nom  | anciennete | dep | posteoccupe |  mu
-----+-------+------------+-----+-------------+------
 e2  | Lucas |         10 | d3  | chercheur   | 0.25
 e8  | Marie |         19 | d4  | PDG         |    1
(2 rows)
```

FIGURE 55 – PostgreSQLf : module d'extension de PostgreSQL pour l'interrogation floue

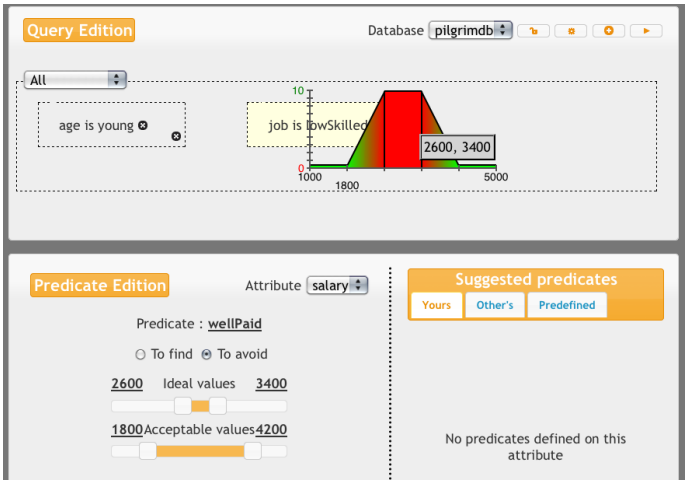

FIGURE 56 – ReqFlex : interface graphique d'interrogation floue

D'un point de vue technique, l'implémentation de PostgreSQLf m'a permis de découvrir le gestionnaire de modules PGXN intégré au SGBD PostgreSQL à partir de la version 9.0. J'ai également pu identifier les limites techniques du gestionnaire PGXN pour l'utilisation des index à partir de fonctions utilisateur. C'est pourquoi nous nous orientons désormais vers un développement de ReqFlex indépendant de tout SGBD.

Cortex, Lucifer et Falstaff

J'ai implémenté entièrement les trois prototypes Cortex, Lucifer et Falstaff sous forme d'interface web connectée à une base de données décrivant des voitures d'occasions. La relation décrivant les voitures d'occasion (95 000 annonces issues du site `Leboncoin.fr`) ont été reliées aux données techniques de chaque véhicule (BD *eurotax glass* décrivant 65 000 véhicules différents) afin d'augmenter le nombre de dimensions sur lesquelles chaque voiture est décrite.

Le prototype Cortex (Fig. 57) est l'implémentation de l'approche décrite dans le papier [14] et a été utilisé pour mener des expérimentations et participer à une session démonstration [12].

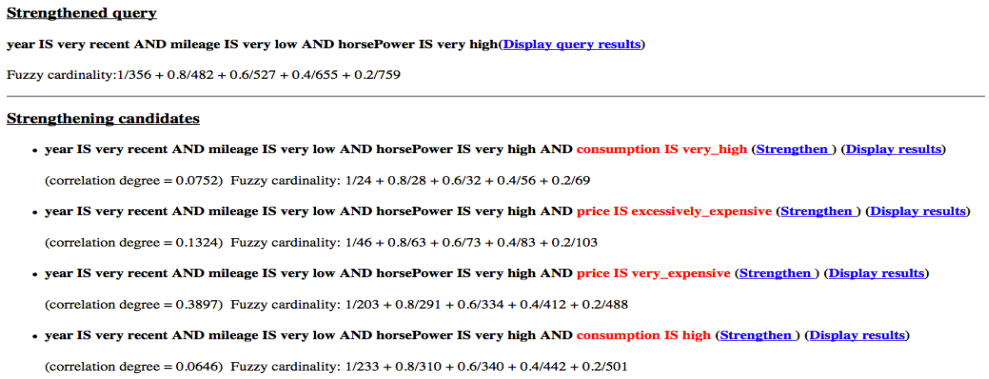

FIGURE 57 – Cortex : Suggestion de requêtes intensifiées

Lucifer (Fig. 58) exploite la même base de données mais implémente l'approche coopérative de gestion des requêtes retournant un ensemble vide de réponses [129]. Ce prototype nous a permis de mener les expérimentations décrites dans la section 8.3.

Falstaff (Fig. 59) est l'implémentaion de l'approche d'interrogation par facette que nous avons proposée dans l'article [126]. Ce travail mériterait d'être davantage visible dans la mesure où il s'agit de l'unique, jusqu'à preuve du contraire, implémentation d'un système de navigation par facette basé sur des descripteurs linguistiques (i.e. SEFs). Le principal axe d'amélioration concerne son ergonomie et son esthétisme afin d'alléger l'interface d'interrogation qui est, à l'heure actuelle, un peu chargée.

Queries, COKE, iKeys

L'interrogation par mots-clés de données structurées constitue une thématique un peu plus exploratoire au sein de l'équipe SHAMAN, puisqu'avant mon arrivée, aucun collègue n'avait encore abordé cette problématique. L'implémentation de prototypes de recherche était alors extrêmement important puisqu'elle permettait de vérifier par la pratique la cohérence et la pertinence des stratégies d'interprétation proposées. De plus, ce travail sur l'interrogation par mots-clés s'inscrit dans le cadre d'un partenariat avec l'entreprise Semsoft (sous forme de contrat de recherche externe et d'un projet PME), l'implémentation de prototypes a ainsi permis d'accélérer le transfert de ces outils dans l'infrastructure logicielle de l'entreprise. J'ai en effet pu conseiller les ingénieurs de Semsoft sur les détails algorithmiques et les stratégies d'implémentation à utiliser. Finalement, le développement des prototypes COKE et iKeys m'a également permis de monter en compétence sur les systèmes de stockage et d'indexation (comme e.g. Virtuoso) de grandes quantités de données sémantiques. Ces prototypes ont en effet été implémentés au dessus de la base IMDB contenant plusieurs dizaines de millions de triplets RDF.

Chronologiquement, Queries (Fig. 60), premier prototype implémenté, est une solution pragmatique, fonctionnelle et très efficace de construction intuitive de requêtes SPARQL. Malgré son efficacité et son utilisation intuitive, Queries souffre d'un manque de flexibilité. COKE (Fig. 61) offre davantage de flexibilité en se positionnant à mi-chemin des stratégies d'interrogation par mots-clés et des systèmes en langue naturelle. Le prototype iKeys (Fig. 62) utilise ensuite des phases d'interaction avec l'utilisateur pour améliorer à la fois la flexibilité et l'expressivité de l'interface d'interrogation. J'ai implémenté entièrement les deux premiers prototypes et ai bénéficié de l'aide de Khadim Dramé pour l'implémentation d'iKeys.

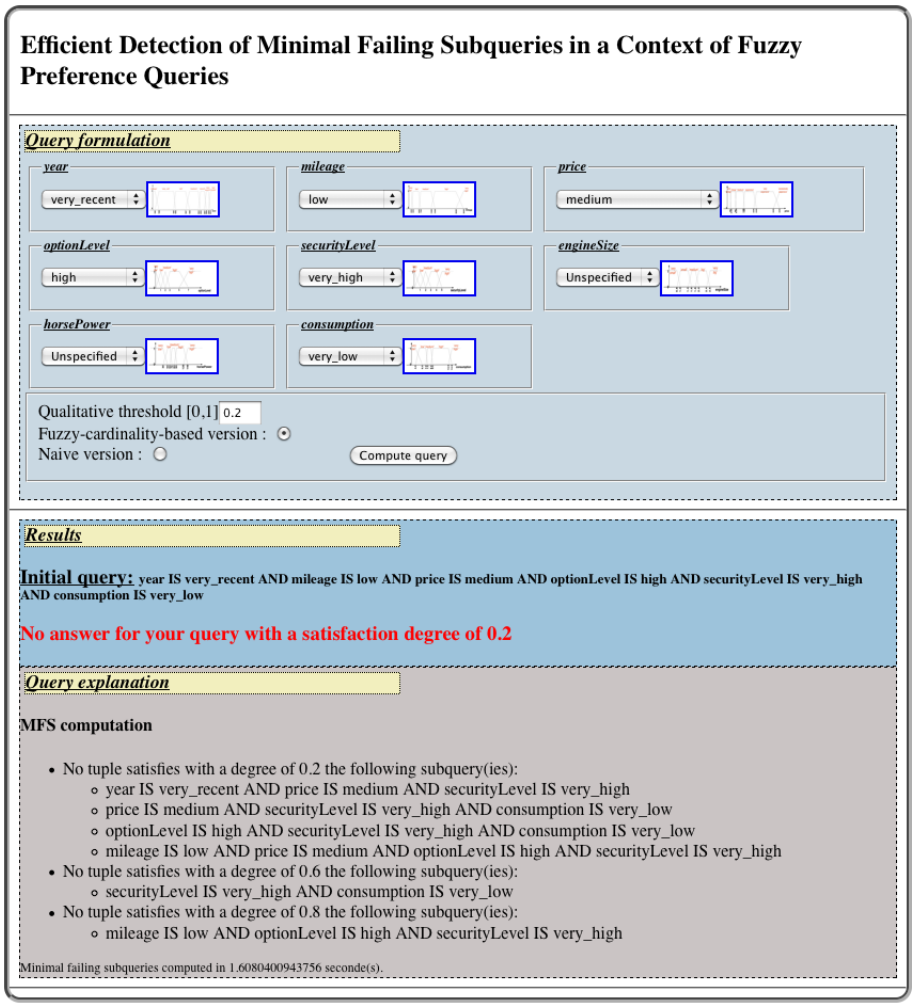

FIGURE 58 – Lucifer : Explication des raisons de l'échec et suggestion de requêtes fonctionnelles

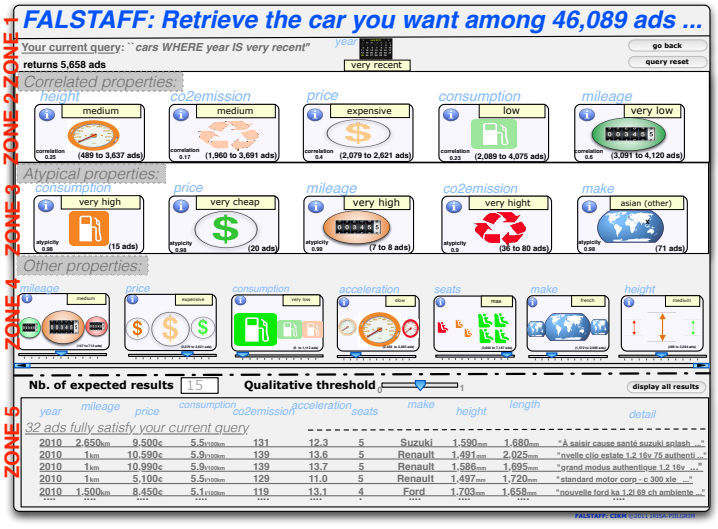

FIGURE 59 – Interface d'interrogation par facettes exploitant un vocabulaire utilisateur

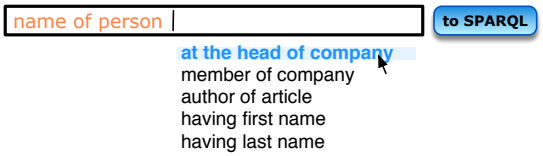

FIGURE 60 – Queries : Parcours interactif et par autocomplétion d'une ontologie

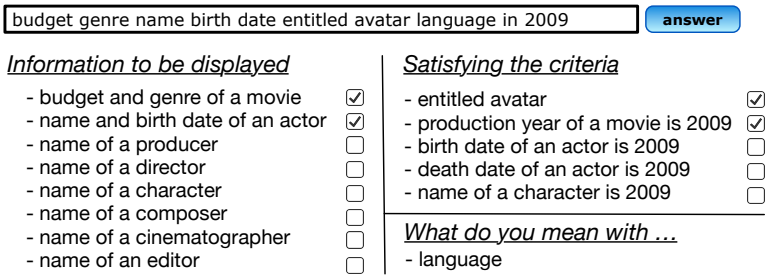

FIGURE 61 – COKE : Construction semi-contrôlée de requêtes par mots-clés

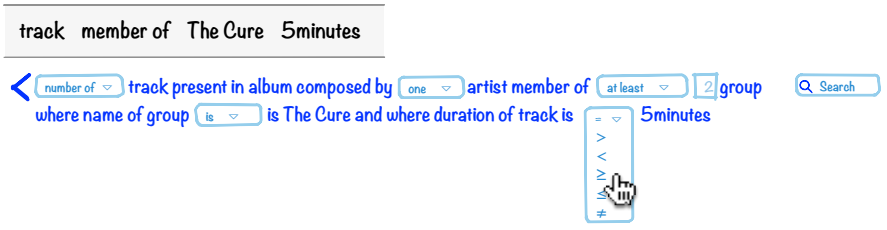

FIGURE 62 – iKeys : Construction interactive de requêtes par mots-clés

C Encadrements

Amira Essaid

Peu après mon arrivée en tant que maître de conférence à l'IUT de Lannion, Arnaud Martin (professeur des universités co-responsable de l'équipe IRISA/DRUID) m'a proposé de co-encadrer la thèse d'Amira Essaid. La répartition de l'encadrement était la suivante : Arnaud Martin directeur à 33%, Boutheina Ben Yaghlane co-encadrement à 33% et moi-même co-encadrement 33%). Le sujet de cette thèse portait sur l'utilisation de la théorie des fonctions de croyance pour mettre en place une stratégie capable de fournir des alignements incertains d'ontologies. A. Martin et B. Ben Yaghlane étaient en charge d'encadrer Amira Essaid sur la formalisation du problème dans le cadre des fonctions de croyance et j'étais en charge du cadre applicatif. L'alignement d'ontologies repose sur l'usage de mesures d'appariement et éventuellement sur l'utilisation de sources de connaissances terminologiques comme WordNet, source que j'ai pu étudier lors de ma thèse. Le contexte de co-tutelle et l'absence d'un financement complet sur cette thèse a rendu sa réalisation délicate, mais la thèse a été soutenue e 2015 et a donné lieu à trois publications, dont deux dans des conférences internationales [50, 51] et une dans une conférence nationale [49].

Amira Essaid occupe actuellement un poste d'ingénieur de recherche à l'université de Strasbourg dans le laboratoire ICube.

Aurélien Moreau

J'ai tout d'abord co-encadré avec O. Pivert le stage de master 2 recherche effectué par Aurélien Moreau dans l'équipe SHAMAN. Ce stage portait sur l'utilisation de la théorie des sous-ensembles flous pour construire une stratégie coopérative visant à expliquer linguistiquement le contenu d'un ensemble de réponses retourné suite à l'exécution d'une requête. Ce stage a ensuite été poursuivi par une thèse financée par le projet ODIN (Sec. A). La sujet de la thèse, qui sera soutenue en décembre 2017, porte sur les approches coopératives permettant à un utilisateur de mieux comprendre et exploiter les résultats de ses requêtes. Trois principales contributions ont été proposées lors de ce projet. La première consiste à générer des explications linguistiques de la structure intrinsèque d'un ensemble de réponses. La seconde vise à enrichir, sous forme de recommantations, les résultats d'une requête par des réponses indirectes. La troisième contribution consiste à aider l'utilisateur à construire une requête. Pour ce faire, au lieu de formaliser son besoin d'information par un ensemble de conditions, l'utilisateur évalue positivement ou négativement des objets de la base, parmi un ensemble pré-identifié, et une stratégie de caractérisation infère la requête respectant ces évaluations binaires. Le travail d'Aurélien Moreau a donné lieu à trois publications dans des conférences internationales [89, 91, 92] (une quatrième vient d'être soumise pour évaluation à ACM/SAC 2017) et deux conférences nationales [109, 90].

Aurélien Moreau occupera un poste d'ingénieur de recherche pédagogique à partir du 1 novembre 2017 à l'ENSSAT de Lannion.

Khadim Dramé

Dans le cadre du projet PME 360Predict (Sec. A), j'ai obtenu un financement pour encadrer un postdoc sur une durée d'un an. Khadim Dramé, après avoir obtenu un doctorat à l'université de Bordeaux, a travaillé sous ma supervision pour le développement d'une approche d'interrogation par mots-clés. Cette approche faisait suite aux travaux que j'ai réalisés pour Semsoft et qui avaient abouti aux systèmes Aggrego Search et COKE. L'objectif de la nouvelle approche, nommée iKeys, était de fournir davantage de liberté à l'utilisateur lors de l'expression de sa requête tout en aboutissant à un langage à base de mots-clés capable d'intégrer des conditions de sélection incluant l'appel à des fonctions d'agrégation (e.g. pour rechercher « les groupes de musiques ayant sorti plus de 5 albums après 2016 »). Ce travail a donné lieu à deux publications dans des conférences internationales [42, 124] et son transfert dans l'écosystème logiciel de l'entreprise Semsoft est en cours.

Khadim Dramé occupe actuellement un poste permanent d'enseignant-chercheur à l'université Assane Seck de Ziguinchor (Sénégal) et effectuera un nouveau séjour de recherche dans l'équipe SHAMAN en octobre 2017.

Toan Ngoc Duong

Toan Ngoc Duong a commencé une thèse CIFRE en partenariat avec l'enterprise Semsoft en octobre 2016. L'objectif de sa thèse est de fournir des fonctionnalités permettant à un expert métier (décisionnaire, communiquant, assureur, etc.) d'explorer et de comprendre un jeu de données brutes. L'enterprise Semsoft est spécialisée dans la médiation de sources de données hétérogènes potentiellement distribuées. Avant d'intégrer une source de données dans un tel système d'information, ce qui est une tâche coûteuse en temps et nécessitant les compétences d'un informaticien, ces fonctionnalités permettront à l'expert métier d'analyser de manière autonome un jeu de données brutes et ainsi juger de son potentiel pour enrichir son entrepôt de données. Pour mener à bien la thèse et le transfert des méthodes développées dans la suite logicielle de Semsoft, je travaille également en étroite collaboration avec les ingénieurs de l'entreprise.

Ahmed Abid

Le projet ODIN (Sec. A) a tout d'abord financé la thèse d'Aurélien Moreau mais également un postdoc d'un an. Le 1 octobre 2017 Ahmed Abid, ayant obtenu en 2017 un doctorat de l'université de Tours, effectuera un postdoc d'un an sous la direction d'Olivier Pivert et moi-même. Le sujet de ce postdoc est de transposer les stratégies coopératives développées lors de la thèse d'Aurélien Moreau pour des données sémantiques et non plus uniquement relationnelles. L'idée est d'exploiter la richesse du schéma sémantique sous-jacent pour identifier des liens entre objets.

Co-encadrements de stagiaires et de prestataires

Au cours de ces neuf dernières années, j'ai eu également l'occasion d'encadrer plusieurs stages dans l'optique de partager avec des étudiants nos problématiques sur la gestion de données. Ces stages avaient des sujets divers allant de problématiques :

- d'administration système pour mettre en place un serveur SVN et des stratégies de sauvegarde sur les postes des members de l'équipe SHAMAN (2 étudiants de LP aGSRI, IUT de Lannion),
- de développement d'interface graphique pour la construction d'un vocabulaire d'interrogation floue (2 étudiants de DUT R&T et informatique, 1 étudiant de deuxième année d'école d'ingénieur).

D Autres activités de recherche

Evaluation d'article

J'ai eu l'occasion de faire partie des comités de programme pour les conférences suivantes :

- IFSA-SCIS (International Fuzzy Systems Association - Soft Computing and Intelligent Systems) 2017,
- FQAS (Flexible Query Answering Systems) 2017,
- DEXA (Database and Expert Systems Applications) 2016,
- IPMU (Information Processing and Management of Uncertainty) 2016,
- IFSA-EUSFLAT (International Fuzzy Systems Association-European Society for Fuzzy Logic and Technology) 2015,
- ISMIS (International Symposium on Methodologies for Intelligent Systems) 2015 - 2017,
- RECITAL (Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues) 2015,

mais également de contribuer en tant que relecteur externe pour les journaux et conférences suivants :

- IJAEIS (International Journal of Agricultural and Environmental Information Systems) 2017,
- ACM/SAC (Symposium On Applied Computing) 2016,
- ADBIS (Advances in Databases and Information Systems) 2015,
- Fuzz-IEEE 2015,
- ECOINF (International Journal on Ecoinformatics and Computational Ecology) 2014,
- ANR-CONTINT 2013,
- COSI (Colloque sur l'Optimisation et les systèmes d'Information) 2011,
- DEXA (Database and Expert Systems Applications) 2011,
- FQAS (Flexible Query Answering Systems) 2011,
- IPMU (Information Processing and Management of Uncertainty) 2010.

Exposés invités

Lors de participations à des conférences, j'ai eu la chance de rencontrer plusieurs membres du Laboratoire d'Informatique Paris 6, notamment des chercheurs de l'équipe LFI dirigée actuellement par Christophe Marsala. Des liens très enrichissants et constructifs ont été créés avec ces personnes et ont donné lieu à des travaux collaboratifs. J'ai eu l'opportunité d'effectuer deux séminaires au LIP6 :

- un premier en 2013 sur les approches coopératives,
- et un second en 2016 sur une approche de *business intelligence* basée sur des techniques de *soft computing*.

Suite au projet PostgreSQLf, j'ai été invité par l'association PostgreSQLf pour effectuer une présentation de ce travail lors de la conférence annuelle francophone sur PostgreSQL (PgDay 2012).

Transfert recherche → enseignement

Bien que la fonction d'enseignant-chercheur puisse conduire à des frustrations quant au temps disponible pour mener des activités de recherche, cette fonction offre des occasions intéressantes de partage de connaissances avec des étudiants. Depuis 2014, je dispense aux étudiants de master 2 de l'USTH (University of Science and Technology of Hanoi) un module d'initiation à la recherche au cours duquel les étudiants réalisent un projet d'analyse de données en utilisant des outils du *soft computing*.

À partir de cette année, je vais également proposer aux étudiants de troisième année de l'ENSSAT un module sur l'interrogation floue de base de données.

E Liste de mes publications

Chapitres de livre

— Olivier Pivert and Grégory Smits, How to Efficiently Diagnose and Repair Fuzzy Database Queries that Fail. Studies in Fuzziness and Soft Computing, Vol. 326, E. Tamir et al. (Eds) : Fifty years of fuzzy logic and its applications, pp. 499-517 (2015).
— Grégory Smits, Olivier Pivert, Allel HadjAli : Fuzzy Cardinalities as a Basis to Cooperative Answering. Flexible Approaches in Data, Information and Knowledge Management, pp. 261-289 (2013).

Articles de journaux

— G. Smits, and O.Pivert. Une approche coopérative d'aide à la réparation de requêtes floues. Revue Ingénierie des Systèmes d'Information, 21(3), 11-30 (2016).
— G. Smits, O. Pivert, and T. Girault. Interrogation floue de bases de données relationnelles : de la théorie à la pratique. Revue d ?Intelligence Artificielle, 29(5), 569-593 (2015).
— P. Bosc, O. Pivert, and G. Smits : An Approach to Database Preference Queries Based on an Outranking Relation. International Journal of Computational Intelligence Systems, Atlantis Press, 5 (4), pp.789-804 (2012).
— O. Pivert, and G. Smits (2012, January). SQLSP : A Preference Query Language with Symbolic Score, Int. J. Computational Intelligence Systems 5(4) : 789-804.
— P. Bosc, A. Hadjali, O. Pivert, and G. Smits. An approach based on predicate correlation to the reduction of plethoric answer sets. In F. Guillet, G. Ritschard, and D. Zighed, editors, Advances in Knowledge Discovery and Management Vol. 2, Studies in Computational Intelligence, vol. 398, pages 213-234. Springer, 2012.
— J. Heinecke, G. Smits, C. Chardenon, E. Guimier De Neef, E. Maillebuau et M. Boualem : TiLT : a natural language processing platform. TAL 49(2) : 17-41 (2008). Conference papers

Actes dans des conférences avec comité d'évaluation

2017

— G Smits, O. Pivert, and P. Nerzic. Calcul, stockage et utilisation efficaces de résumés linguistiques de données massives. Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'17), 19-20 octobre 2017, Amiens, France.
— G. Smits, M.J. Lesot, and O. Pivert. Vocabulary Elicitation for Informative Descriptions of Classes. In Proc. of the joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCSI'17), Otsu, Japan, 2017.
— G. Smits, R.R. Yager, and O. Piver. Interactive Data Exploration on Top of Linguistic Summaries. In Proc. of the 26th IEEE International Conference on Fuzzy Systems (Fuzz-IEEE'17), Naples, Italia, 2017.
— A. Moreau, O. Pivert, and G. Smits. A Typicality-Based Recommendation Approach Leveraging Demographic Data. In Proc. of the 12th International Conference on Flexible Query Answering Systems (FQAS'17), London, England, 2017.

2016

— K. Dramé, G. Smits and O. Pivert. IKEYS : Interactive Keyword Search Dedicated to Corporate Data. In Proc. of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW ?16), 19-23 November 2016, Bologna, Italy.
— O. Pivert, O. Slama, G. Smits, and V. Thion. A Fuzzy Extension of SPARQL for Querying Gradual RDF Data.In Proc. of the IEEE International Conference on Research Challenges in Information Science, 1-3 June 2016, Grenoble, France.
— A. Moreau, O. Pivert, and G. Smits. Caractérisation floue de clusters de réponses. Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'16), 3-4 novembre

2016, La Rochelle, France.

— O. Pivert, O. Slama, G. Smits, and V. Thion. SUGAR : A Graph Database Fuzzy Querying System.In Proc. of the IEEE International Conference on Research Challenges in Information Science, 1-3 June 2016, Grenoble, France.

— G. Smits, O. Pivert, and R.R. Yager. A Soft Computing Approach to Agile Business Intelligence. In Proc. of the 25th IEEE International Conference on Fuzzy Systems (Fuzz-IEEE'16), Vancouver, Canada, 2016.

— A. Moreau, O. Pivert, and G. Smits. A Fuzzy Approach to the Characterization of Database Query Answers. In Proc. of the 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU ?16), Eindhoven, The Netherlands, 2016.

2015

— K. Dramé, G. Smits and O. Pivert. Coarse to Fine Keyword Queries with User Interactions. In Proc. of the 17th International Conference on Information Integration and Web-based Applications & Services (iiWas'15), Bruxelles, Belgium, 2015.

— A. Moreau, O. Pivert, and G. Smits. A Clustering-Based Approach to the Explanation of Database Query Answers. In Proc. of the 11th International Conference on Flexible Query Answering Systems (FQAS'13), Cracow, Poland, 2015.

— O. Pivert, and G. Smits. Réparation efficace de requêtes floues, In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'15), Poitiers, France, 2015.

— G. Smits, and O. Pivert. Explications linguistiques et graphiques de groupes de données, In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'15), Poitiers, France, 2015.

— O. Pivert, G. Smits, and V. Thion. Expression and Efficient Processing of Fuzzy Queries in a Graph Database Context, Proc. of the 24th IEEE International Conference on Fuzzy Systems (Fuzz-IEEE'15), Istanbul, Turkey, 2015.

— G. Smits, O. Pivert, and V. Thion. Connected Keywords, In Proc. of the 9th IEEE International Conference on Research Challenges in Information Science (RCIS ?15), Athens, Greece, 2015.

— G. Smits and O. Pivert. Linguistic and Graphical Explanation of a Cluster-Based Data Structure, In Proc. of the 9th International Conference on Scalable Uncertainty Management (SUM ?15), Quebec City, Canada, 2015.

— G. Smits and O. Pivert. Une approche coopérative d'aide à la réparation de requêtes floues. In Actes des 31e Journées Bases de Données Avancées (BDA'15), Île de Porquerolles, France, 2015.

2014

— G. Smits, O. Pivert, H. Jaudoin, and F. Paulus. AGGREGO SEARCH : Interactive Keyword Query Construction. In Proc. of the 17th International Conference on Extending Database Technology (EDBT'14), demo session, Athens, Greece, pages 636-639, 2014.

— G. Smits, O. Pivert, and M.-J. Lesot. A Vocabulary Revision Method Based on Modality Splitting. In Proc. of the 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'14), Montpellier, France, CCIS vol. 444, Springer, pages 140-149, 2014.

— O. Pivert and G. Smits. Plethoric Answers to Fuzzy Queries : A Reduction Method Based on Query Mining. In Proc. of the 21st International Symposium on Methodologies for Intelligent Systems (ISMIS'14), Roskilde, Denmark, LNAI vol. 8502, Springer, pages 295-304, 2014.

— H. Jaudoin, O. Pivert, G. Smits, and V. Thion. Data-Quality-Aware Skyline Queries. In Proc. of the 21st International Symposium on Methodologies for Intelligent Systems (ISMIS'14), Roskilde, Denmark, LNAI vol. 8502, Springer, pages 530-535, 2014.

— O. Pivert, V. Thion, H. Jaudoin, and G. Smits. On a Fuzzy Algebra for Querying Graph Databases. In Proc. of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'14), Limassol, Cyprus, pages 748-755, 2014.

— O. Pivert, V. Thion, H. Jaudoin, and G. Smits. Une algèbre floue pour l'interrogation flexible de bases de données graphes. In Actes des 30e journées Bases de Données Avancées (BDA'14), Grenoble-Autrans, France, 2014.
— G. Smits, O. Pivert, and M.-J. Lesot. Ajustement automatique de vocabulaire expert par scission de modalité. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'14), Cargèse, France, 2014.
— P. Nerzic, G. Smits, and O. Pivert. Un nouvel usage des prédicats flous pour l ?interrogation flexible de base de données. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'14), Cargèse, France, 2014.
— O. Pivert, G. Smits, A. Moreau, and H. Jaudoin. Réponses connexes fondées sur des associations typiques. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'14), Cargèse, France, 2014.
— A. Essaid, A. Martin, G. Smits, and B. Ben Yaghlane, "A distance-based decision in the credal level". International Conference on Artificial Intelligence and Symbolic Computation AISC 2014, Sevilla, Spain, December 2014.
— A. Essaid, A. Martin, G. Smits, and B. Ben Yaghlane. "Uncertainty in ontology matching : a decision rule-based approach". In IPMU, Montpellier, France, July 2014.

2013

— G. Smits, O. Pivert, T. Girault : ReqFlex : Fuzzy Queries for Everyone. PVLDB 6(12) : 1206-1209, Riva del Garda, Italia, 2013.
— G. Smits, O. Pivert, and T. Girault. Towards reconciling expressivity, efficiency and user-friendliness in database flexible querying. In Proc. of the 22th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'13), Hyderabad, India, 2013.
— M.-J. Lesot, G. Smits, and O. Pivert. Adequacy of a user-defined vocabulary to the data structure. In Proc. of the 22th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'13), Hyderabad, India, 2013.
— O. Pivert, G. Smits, and H. Jaudoin. Finding similar objects in relational databases – An association-based fuzzy approach. In Proc. of the 10th International Conference on Flexible Query Answering Systems (FQAS'13), Granada, Spain, LNAI vol. 8132, pages 425-436, 2013.
— G. Smits, O. Pivert, H. Jaudoin, and F. Paulus. An autocompletion mechanism for enriched keyword queries to RDF data sources. In Proc. of the 10th International Conference on Flexible Query Answering Systems (FQAS'13), Granada, Spain, LNAI vol. 8132, pages 601-612, 2013.
— G. Smits, O. Pivert, and T. Girault. Requêtes floues et SGBD relationnels : vers un couplage renforcé. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'13), Reims, pages 77-84, 2013.
— M.-J. Lesot, G. Smits, and O. Pivert. Mesures d'adéquation entre vocabulaire expert et structure de données. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'13), Reims, pages 243-250, 2013.
— A. Essaid, A. Martin, G. Smits, and B. Ben Yaglhane, "Processus de Décision Crédibiliste pour l'Alignement des Ontologies", Rencontres Francophones sur la Logique Floue et ses Applications (LFA), octobre 2013, Reims, France.

2012

— O. Pivert and G. Smits. Towards an efficient processing of outranking-based preference queries. In Proc. of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'12), Catania, Italy, CCIS vol. 297, Springer, pages 471-480, 2012.
— O. Pivert and G. Smits. On fuzzy preference queries explicitly handling satisfaction levels. In Proc. of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'12), Catania, Italy, CCIS vol. 297, Springer, pages 341-350, 2012.

— O. Pivert and G. Smits. On a preference query language that handles symbolic scores. In Proc. of the 16th East-European Conference on Advances in Databases and Information Systems (ADBIS'12), Poznan, Poland, LNCS vol. 7503, Springer, pages 296-309, 2012.
— G. Smits and O. Pivert. A fuzzy-summary-based approach to faceted search in relational databases. In Proc. of the 16th East-European Conference on Advances in Databases and Information Systems (ADBIS'12), Poznan, Poland, LNCS vol. 7503, Springer, pages 357-370, 2012.
— O. Pivert, G. Smits, A Hadjali, and H. Jaudoin. LUCIFER : Un système de détection de conflits dans les requêtes flexibles. In Actes de la 12e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC'12), Bordeaux, France, pages 617-620, 2012.
— O. Pivert, G. Smits, H. Jaudoin, and A. Hadjali. Détection efficace de conflits dans les requêtes floues. In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA'12), Compiègne, pages 97-104, 2012.
— G. Smits, O. Pivert, and T. Girault. PostgreSQLf : un système d'interrogation floue. In Actes des 28e journées Bases de Données Avancées (BDA'12), Session démonstrations, Clermont-Ferrand, France, 2012.

2011

— P. Bosc, O. Pivert, and G. Smits. A preference query model based on a fusion of local orders. In Proc. of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'11), Belfast, Northern Ireland, UK, LNAI vol. 6717, Springer, pages 725-736, 2011.
— O. Pivert, A. Hadjali, and G. Smits. On database queries involving inferred fuzzy predicates. In Proc. of the 19th International Symposium on Methodologies for Intelligent Systems (ISMIS'11), Warsaw, Poland, LNAI vol. 6804, Springer, pages 592-601, 2011.
— O. Pivert, A. Hadjali, and G. Smits. Searching for a compromise between satisfaction and diversity in database fuzzy querying. In Proc. of the joint 7th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT'11) and Rencontres Francophones sur la Logique Floue et ses Applications (LFA'11), Aix-Les-Bains, France, pages 402-408, 2011.
— O. Pivert, G. Smits, and A. Hadjali. Processing fuzzy queries in a peer data management system using distributed fuzzy summaries. In Proc. of the 5th International Conference on Scalable Uncertainty Management (SUM'11), Dayton, OH, USA, LNAI vol. 6929, Springer, pages 359-372, 2011.
— O. Pivert, G. Smits, A. Hadjali, and H. Jaudoin. Efficient detection of minimal failing subqueries in a fuzzy querying context. In Proc. of the 15th East-European Conference on Advances in Databases and Information Systems (ADBIS'11), Vienna, Austria, LNCS vol. 6909, Springer, pages 243-256, 2011.

2010

— P. Bosc, A. Hadjali, O. Pivert, and G. Smits. Trimming plethoric answers to fuzzy queries : An approach based on predicate correlation. In Proc. of the 13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'10), Dortmund, Germany, LNCS vol. 6178, Springer, pages 595-604, 2010.
— P. Bosc, A. Hadjali, O. Pivert, and G. Smits. On the use of fuzzy cardinalities for reducing plethoric answers to fuzzy queries. In Proc. of the 4th International Conference on Scalable Uncertainty Management (SUM'10), Toulouse, France, LNAI vol. 6379, Springer, pages 98-111, 2010.
— P. Bosc, O. Pivert, and G. Smits. A database preference query model based on a fuzzy outranking relation. In Proc. of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'10), Barcelona, Spain, pages 38-43, 2010.
— P. Bosc, O. Pivert, and G. Smits. A model based on outranking for database preference queries. In Proc. of the 13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'10), Dortmund, Germany,

CCIS vol. 81, Springer, pages 95-104, 2010.

— O. Pivert, A. Hadjali, and G. Smits, Estimating the relevance of a data source using a fuzzy-cardinality-based summary, In Proc. of the 5th IEEE International Conference on Intelligent Systems (IEEE IS'10), London, Great-Britain, pages 96-101, 2010.

— P. Bosc, A. Hadjali, O. Pivert, and G. Smits. Une approche fondée sur la corrélation entre prédicats pour le traitement des réponses pléthoriques. In Actes de la 10e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC'10), Hammamet, Tunisie, pages 273-284, 2010.

— P. Bosc, A. Hadjali, O. Pivert, and G. Smits. CORTEX – CORrelaTion-based Query EXpansion. In Actes des 25e journées Bases de Données Avancées (BDA'10), Session démonstrations, Toulouse, France, 2010.

2009

— P. Bosc, O. Pivert, and G. Smits. A flexible querying approach based on outranking and classification. In Proc. of the 8th International Conference on Flexible Query Answering Systems (FQAS'09), pages 1-12, Roskilde, Denmark, 2009.

— G. Smits, and C. Chardenon. Controlling an SMS transcription system using heuristic and empirical criteria, In Proc. of 10th International Conference on Computing, Mexico City, Mexico, 2009.

2007

— G. Smits, and C. Chardenon. Vers une méthodologie générique de contrôle basée sur la combinaison de sources de jugement. In Actes de la 14ème conférence sur le Traitement Automatique des Langues Naturelles (pp. 273-282), Toulouse, France, 2007.

— O. Tardif, and G. Smits. Resolving coreference using an outranking approach. In Proc. of Recent Advances in Natural Language Processing, Borovetz, Bulgaria, 2007.

2006

— G. Smits, M. Plu, and P. Bellec. Personal semantic indexation of images using textual annotations (pp. 71-85). Springer Berlin Heidelberg, Athens, Greece, 2006.

— G. Smits. Contrôle dynamique multicritère des résultats d ?une chaîne de tal. in proceedings of RECITAL, Louvain, Belgique, 2006.

— C. Chardenon, and G. Smits. Combining Preferences to Control a Natural Language Processing Chain. In Proceedings of Multidisciplinary ECAI'06 Workshop on Advances in Preferences Handling (pp. 128-134), Riva del Garda, Italia, 2006.

Rapports scientifiques

— G. Smits. Une approche par surclassement pour le contrôle d'un processu d'analyse linguistique. Doctorat de l'Université de Caen, spécialité informatique, 2008.

— M. Guyomard, P. Alain, A. Hadjali, H. Jaudoin, and G. Smits. First Balance Sheeet of a Formal Approach to the Teaching of Data Structures, The B Method : from Research to Teaching , Nantes, France,2008.

— G. Smits. Méthodologie d'Aide Multicritère à la Décision pour le Contrôle d'une Chaîne de Traitement Automatique des Langues Naturelles, Forum ROADEF : Société française de Recherche Opérationnelle et d'Aide à la Décision, Grenoble, France, 2007.

— G. Smits. Entre analyse et décision. colloque international des doctorants et jeunes chercheurs associés du laboratoire, Université de Caen 2006.

Références

[1] R. Agrawal and E. L. Wimmers. A framework for expressing and combining preferences. In *ACM SIGMOD Record*, volume 29, pages 297–306. ACM, 2000.

[2] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1) :243–256, 2013.

[3] P. Bailis, E. Gan, S. Madden, D. Narayanan, K. Rong, and S. Suri. Macrobase : Prioritizing attention in fast data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 541–556. ACM, 2017.

[4] S. Batra and C. Tyagi. Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(2) :509–512, 2012.

[5] J. C. Bezdek, R. Ehrlich, and W. Full. Fcm : The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3) :191–203, 1984.

[6] A. Bhattacharya. *Fundamentals of database indexing and searching*. CRC Press, 2014.

[7] F. E. Boran, D. Akay, and R. R. Yager. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, 2016.

[8] G. Bordogna and G. Pasi. *Recent issues on fuzzy databases*, volume 53. Physica, 2013.

[9] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430. IEEE, 2001.

[10] P. Bosc, D. Dubois, O. Pivert, H. Prade, and M. De Calmes. Fuzzy summarization of data using fuzzy cardinalities. In *Proceedings of IPMU*, volume 2002, pages 1553–1559, 2002.

[11] P. Bosc, A. Hadjali, and O. Pivert. Empty versus overabundant answers to flexible relational queries. *Fuzzy sets and systems*, 159(12) :1450–1467, 2008.

[12] P. Bosc, A. Hadjali, O. Pivert, and G. Smits. Cortex : Correlation-based query expansion. In *26èmes Journées Bases de Données Avancées (BDA'10)*, 2010.

[13] P. Bosc, A. Hadjali, O. Pivert, and G. Smits. On the use of fuzzy cardinalities for reducing plethoric answers to fuzzy queries. In *SUM*, pages 98–111. Springer, 2010.

[14] P. Bosc, A. Hadjali, O. Pivert, and G. Smits. Une approche fondée sur la corrélation entre prédicats pour le traitement des réponses pléthoriques. In *10ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2010)*, pages 273–284, 2010.

[15] P. Bosc, O. Piver, and G. Smits. Sqlsp : A preference query language with symbolic scores. *Int. J. Computational Intelligence Systems*, 5(4) :789–804, 2012.

[16] P. Bosc and O. Pivert. Some approaches for relational databases flexible querying. *Journal of Intelligent Information Systems*, 1(3) :323–354, 1992.

[17] P. Bosc and O. Pivert. SQLf : a relational database language for fuzzy querying. *IEEE Trans. on Fuzzy Systems*, 3(1) :1–17, 1995.

[18] P. Bosc, O. Pivert, D. Dubois, and H. Prade. On fuzzy association rules based on fuzzy cardinalities. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 1, pages 461–464. IEEE, 2001.

[19] P. Bosc, O. Pivert, and G. Smits. On a fuzzy group-by and its use for fuzzy association rule mining. In *ADBIS*, pages 88–102. Springer, 2010.

[20] P. Bosc, O. Pivert, and G. Smits. A preference query model based on a fusion of local orders. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 725–736. Springer, 2011.

[21] P. Bosc, O. Pivert, and G. Smits. An approach to database preference queries based on an outranking relation. *International Journal of Computational Intelligence Systems*, 5(4) :789–804, 2012.

[22] B. Bouchon-Meunier, G. Coletti, M.-J. Lesot, and M. Rifqi. Towards a conscious choice of a fuzzy similarity measure : A qualitative point of view. In *IPMU*, pages 1–10. Springer, 2010.

[23] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy sets and systems*, 84(2) :143–153, 1996.

[24] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Cp-nets : A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.(JAIR)*, 21 :135–191, 2004.

[25] D. Bouyssou. Outranking methods. In *Encyclopedia of optimization*, pages 2887–2893. Springer, 2008.

[26] R. J. Brachman, H. J. Levesque, and R. Reiter. *Knowledge representation*. MIT press, 1992.

[27] P. Buche, J. Dibie-Barthélemy, and H. Chebil. Flexible sparql querying of web data tables driven by an ontology. In *International Conference on Flexible Query Answering Systems*, pages 345–357. Springer, 2009.

[28] A. Castelltort and A. Laurent. Fuzzy queries over nosql graph databases : perspectives for extending the cypher language. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 384–395. Springer, 2014.

[29] H. Chen, R. H. Chiang, and V. C. Storey. Business intelligence and analytics : From big data to big impact. *MIS quarterly*, 36(4) :1165–1188, 2012.

[30] J. Chomicki. Querying with intrinsic preferences. In *Proc. of EDBT'02*, pages 34–51, 2002.

[31] K. Collier. *Agile analytics : A value-driven approach to business intelligence and data warehousing*. Addison-Wesley, 2011.

[32] F. Corella and K. Lewison. A brief overview of cooperative answering. In *Technical report http : //www.pomcor.com/whitepapers/cooperative_responses.pdf*, 2009.

[33] F. Cuppens and R. Demolombe. Cooperative ansering : A methodology to provide intelligent access to databases. In *Proc. of DEXA'88*, pages 333–353, 1988.

[34] M. De Calmès, D. Dubois, E. Hullermeier, H. Prade, and F. Sedes. Flexibility and fuzzy case-based evaluation in querying : An illustration in an experimental setting. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(01) :43–66, 2003.

[35] J. Dean and S. Ghemawat. Mapreduce : simplified data processing on large clusters. *Communications of the ACM*, 51(1) :107–113, 2008.

[36] M. Delgado, D. Sánchez, and M. A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23(1) :23–66, 2000.

[37] I. Dellal, S. Jean, A. Hadjali, B. Chardin, and M. Baron. On addressing the empty answer problem in uncertain knowledge bases. In *International Conference on Database and Expert Systems Applications*, pages 120–129. Springer, 2017.

[38] L. Di Jorio, A. Laurent, and M. Teisseire. Fast extraction of gradual association rules : A heuristic based method. In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pages 205–210. ACM, 2008.

[39] L. Di-Jorio, A. Laurent, and M. Teisseire. Mining frequent gradual itemsets from large databases. In *International Symposium on Intelligent Data Analysis*, pages 297–308. Springer, 2009.

[40] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. Zhai. Topcells : Keyword-based search of top-k aggregated documents in text cube. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 381–384. IEEE, 2010.

[41] M. Doumpos, Y. Marinakis, M. Marinaki, and C. Zopounidis. An evolutionary approach to construction of outranking models for multicriteria classification : The case of the electre tri method. *European Journal of Operational Research*, 199(2) :496–505, 2009.

[42] K. Dramé, G. Smits, and O. Pivert. Coarse to fine keyword queries with user interactions. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, page 66. ACM, 2015.

[43] D. Dubois and H. Prade. Fuzzy cardinality and the modeling of imprecise quantification. *Fuzzy sets and Systems*, 16(3) :199–230, 1985.

[44] D. Dubois and H. Prade. A note on measures of specificity for fuzzy sets. *International Journal of General System*, 10(4) :279–283, 1985.

[45] D. Dubois and H. Prade. Théorie des possibilités, 1985.

[46] D. Dubois and H. Prade. Weighted minimum and maximum operations in fuzzy set theory. *Information Sciences*, 39(2) :205–210, 1986.

[47] D. Dubois and H. Prade. Measuring properties of fuzzy sets : a general technique and its use in fuzzy query evaluation. *Fuzzy Sets and Systems*, 38(2) :137–152, 1990.

[48] D. Dubois and H. Prade. Using fuzzy sets in flexible querying : Why and how ? In *Flexible query answering systems*, pages 45–60. Springer, 1997.

[49] A. Essaid, A. Martin, G. Smits, and B. B. Yaghlane. *Processus de Décision Crédibiliste pour l'Alignement des Ontologies*. PhD thesis, Université de Rennes 1, 2012.

[50] A. Essaid, A. Martin, G. Smits, and B. B. Yaghlane. A distance-based decision in the credal level. In *International Conference on Artificial Intelligence and Symbolic Computation*, pages 147–156. Springer, 2014.

[51] A. Essaid, A. Martin, G. Smits, and B. B. Yaghlane. Uncertainty in ontology matching : A decision rule-based approach. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 46–55. Springer, 2014.

[52] U. M. Fayyad, A. Wierse, and G. G. Grinstein. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

[53] J. R. Figueira, V. Mousseau, and B. Roy. Electre methods. In *Multiple Criteria Decision Analysis*, pages 155–185. Springer, 2016.

[54] P. Fishburn. Preference structures and their numerical representations. *Theoretical Computer Science*, 217(2) :359–383, 1999.

[55] P. C. Fishburn. Utility theory for decision making. Technical report, DTIC Document, 1970.

[56] G. Fokou, S. Jean, A. Hadjali, and M. Baron. Cooperative techniques for SPARQL query relaxation in rdf databases. In *European Semantic Web Conference*, pages 237–252. Springer, 2015.

[57] S. Funk. Netflix update : Try this at home, 2006.

[58] J. Fürnkranz and E. Hüllermeier. *Preference learning : An introduction*. Springer, 2010.

[59] T. Gaasterland, P. Godfrey, and J. Minker. Relaxation as a platform for cooperative answering. *Journal of Intelligent Information Systems*, 1(3-4) :296–321, 1992.

[60] P. Godfrey. Minimization in cooperative response to failing database queries. *Int. J. Cooperative Inf. Syst.*, 6(2) :95–149, 1997.

[61] M. Golfarelli, S. Rizzi, and I. Cella. Beyond data warehousing : what's next in business intelligence ? In *Proceedings of the 7th ACM international workshop on Data warehousing and OLAP*, pages 1–6. ACM, 2004.

[62] S. Guillaume and B. Charnomordic. Generating an interpretable family of fuzzy partitions from data. *IEEE transactions on fuzzy systems*, 12(3) :324–335, 2004.

[63] S. Guillaume, B. Charnomordic, and P. Loisel. Fuzzy partitions : a way to integrate expert knowledge into distance calculations. *Information sciences*, 245 :76–95, 2013.

[64] R. Hammah and J. Curran. On distance measures for the fuzzy k-means algorithm for joint data. *Rock Mechanics and Rock Engineering*, 32(1) :1–27, 1999.

[65] M. Hearst. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR workshop on faceted search*, pages 1–5. Seattle, WA, 2006.

[66] M. Hearst, D. Rosner, et al. Tag clouds : Data analysis tool or social signaller ? In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 160–160. IEEE, 2008.

[67] J. Heinecke, G. Smits, C. Chardenon, E. G. De Neef, E. Maillebuau, and M. Boualem. Tilt : plate-forme pour le traitement automatique des langues naturelles. *Traitement automatique des langues*, 49(2) :17–41, 2008.

[68] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :5–53, 2004.

[69] V. Hristidis and Y. Papakonstantinou. Discover : Keyword search in relational databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 670–681. VLDB Endowment, 2002.

[70] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323, 1999.

[71] D. Jannach. Techniques for fast query relaxation in content-based recommender systems. In *Proc. of KI'06*, pages 49–63, 2006.

[72] D. Jannach. Finding preferred query relaxations in content-based recommenders. In *Intelligent Techniques and Tools for Novel System Architectures*, pages 81–97. Springer, 2008.

[73] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[74] S.-J. Kaplan. Cooperative responses from a portable natural language query system. *Artificial Intelligence*, 19 :165–187, 1982.

[75] W. Kießling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 311–322. VLDB Endowment, 2002.

[76] W. Kießling and G. Köstler. Preference SQL — design, implementation, experiences. In *Proc. of the 2002 VLDB Conference*, pages 990–1001, 2002.

[77] G. Koutrika, A. Simitsis, and Y. E. Ioannidis. Explaining structured queries in natural language. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 333–344. IEEE, 2010.

[78] N. Labroche. New incremental fuzzy c medoids clustering algorithms. In *Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American*, pages 1–6. IEEE, 2010.

[79] Y. Lei, V. Uren, and E. Motta. Semsearch : A search engine for the semantic web. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 238–245. Springer, 2006.

[80] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM, 2005.

[81] M.-J. Lesot and A. R. d'Allonnes. Credit-card fraud profiling using a hybrid incremental clustering methodology. In *Proc. of the Int. Conf. on Scalable Uncertainty Management (SUM'12)*, 2012.

[82] M.-J. Lesot, M. Rifqi, and H. Benhadda. Similarity measures for binary and numerical data : a survey. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 1(1) :63–84, 2008.

[83] M.-J. Lesot, M. Rifqi, and B. Bouchon-Meunier. Fuzzy prototypes : From a cognitive view to a machine learning principle. In *Fuzzy Sets and Their Extensions : Representation, Aggregation and Models*, pages 431–452. Springer, 2008.

[84] M.-J. Lesot, G. Smits, and O. Pivert. Adequacy of a user-defined vocabulary to the data structure. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[85] N. Marin, G. Rivas-Gervilla, D. Sanchez, and R. R. Yager. Specificity measures and referential success. *IEEE Transactions on Fuzzy Systems*, PP(99) :1–1, 2017.

[86] C. Marsala. Fuzzy partition inference over a set of numerical values. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems*, pages 1512–1517, 1995.

[87] C. Marsala and B. Bouchon-Meunier. Fuzzy partitioning using mathematical morphology in a learning scheme. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 2, pages 1512–1517. IEEE, 1996.

[88] J. N. Mordeson and P. S. Nair. *Fuzzy graphs and fuzzy hypergraphs*, volume 46. Physica, 2012.

[89] A. Moreau, O. Pivert, and G. Smits. A clustering-based approach to the explanation of database query answers. In *Proc. of the Int. Conf. on Flexible Query Answering Systems*, 2015.

[90] A. Moreau, O. Pivert, and G. Smits. Caractérisation floue de clusters de réponses. In *Rencontres Francophones sur la Logique Floue et ses Applications (LFA'16)*, pages 235–243. Cépaduès, 2016.

[91] A. Moreau, O. Pivert, and G. Smits. A fuzzy approach to the characterization of database query answers. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 329–340. Springer, 2016.

[92] A. Moreau, O. Pivert, and G. Smits. A typicality-based recommendation approach leveraging demographic data. In *International Conference on Flexible Query Answering Systems*, pages 71–83. Springer, 2017.

[93] A. Motro. Vague : A user interface to relational databases that permits vague queries. *ACM Transactions on Information Systems (TOIS)*, 6(3) :187–214, 1988.

[94] A. Motro. Cooperative database system. In *Proc. of FQAS'94*, pages 1–16, 1994.

[95] G. Pasi. Modeling users' preferences in systems for information access. *International Journal of Intelligent Systems*, 18(7) :793–808, 2003.

[96] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13(5-6) :393–408, 1999.

[97] P. Perny and M. Roubens. Fuzzy preference modeling. *Fuzzy sets in decision analysis, operations research and statistics*, pages 3–30, 1998.

[98] P. Perny and B. Roy. The use of fuzzy outranking relations in preference modelling. *Fuzzy sets and Systems*, 49(1) :33–53, 1992.

[99] O. Pivert. *Contribution à l'interrogation flexible de bases de données : expression et évaluation de requêtes floues.* PhD thesis, Université de Rennes 1, 1991.

[100] O. Pivert and P. Bosc. *Fuzzy preference queries to relational databases.* World Scientific, 2012.

[101] O. Pivert, A. Hadjali, and G. Smits. Estimating the relevance of a data source using a fuzzy-cardinality-based summary. In *Intelligent Systems (IS), 2010 5th IEEE International Conference*, pages 96–101. IEEE, 2010.

[102] O. Pivert, O. Slama, G. Smits, and V. Thion. A fuzzy extension of SPARQL for querying gradual RDF data. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–2. IEEE, 2016.

[103] O. Pivert, O. Slama, G. Smits, and V. Thion. Sugar : A graph database fuzzy querying system. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–2. IEEE, 2016.

[104] O. Pivert and G. Smits. On a preference query language that handles symbolic scores. In *East European Conference on Advances in Databases and Information Systems*, pages 296–309. Springer, 2012.

[105] O. Pivert and G. Smits. Towards an efficient processing of outranking-based preference queries. *Advances on Computational Intelligence*, pages 471–480, 2012.

[106] O. Pivert, G. Smits, and A. Hadjali. Processing fuzzy queries in a peer data management system using distributed fuzzy summaries. In *SUM*, pages 359–372. Springer, 2011.

[107] O. Pivert, G. Smits, A. Hadjali, and H. Jaudoin. Efficient detection of minimal failing subqueries in a fuzzy querying context. In *ADBIS*, pages 243–256, 2011.

[108] O. Pivert, G. Smits, and H. Jaudoin. Finding similar objects in relational databases—an association-based fuzzy approach. In *International Conference on Flexible Query Answering Systems*, pages 425–436. Springer, 2013.

[109] O. Pivert, G. Smits, A. Moreau, and H. Jaudoin. Réponses connexes fondées sur des associations typiques. In *Rencontres Francophones sur la Logique Floue et ses Applications (LFA'14)*. Cépaduès, 2014.

[110] O. Pivert, G. Smits, and V. Thion. Expression and efficient processing of fuzzy queries in a graph database context. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–8. IEEE, 2015.

[111] O. Pivert, V. Thion, H. Jaudoin, and G. Smits. On a fuzzy algebra for querying graph databases. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 748–755. IEEE, 2014.

[112] H. Prade and C. Testemale. Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information sciences*, 34(2) :115–143, 1984.

[113] K. Q. Pu and X. Yu. Keyword query cleaning. *Proceedings of the VLDB Endowment*, 1(1) :909–920, 2008.

[114] L. Qin, J. X. Yu, and L. Chang. Keyword search in databases : the power of rdbms. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 681–694. ACM, 2009.

[115] R.-W. Ras and A. Dardzinska. Intelligent query answering. In J. Wang, editor, *Encyclopedia of Data Warehousing and Mining - 2nd Edition*, volume II, pages 1073–1078. Idea Group, Inc, 2008.

[116] G. Raschia and N. Mouaddib. Saintetiq : a fuzzy set-based approach to database summarization. *Fuzzy Sets and Systems*, 129(2) :137 – 162, 2002.

[117] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender systems handbook*. Springer, 2015.

[118] M. Rifqi. Constructing prototypes from large databases. In *International conference on Information Processing and Management of Uncertainty in knowledge-based systems, IPMU'96*, 1996.

[119] A. Rosenfeld. Fuzzy graphs. *Fuzzy sets and their applications*, 513 :77–95, 1975.

[120] M. Roubens and P. Vincke. *Preference modelling*, volume 250. Springer Science & Business Media, 2012.

[121] B. Roy. The outranking approach and the foundations of electre methods. In *Readings in multiple criteria decision aid*, pages 155–183. Springer, 1990.

[122] A. Sen. Social choice theory. *Handbook of mathematical economics*, 3 :1073–1181, 1986.

[123] G. Smits. *Une approche par surclassement pour le contrôle d'un processus d'analyse linguistique*. PhD thesis, Caen, 2008.

[124] G. Smits, K. Drame, and O. Pivert. IKEYS : Interactive Keyword Search Dedicated to Corporate Data. In *20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16)*, Proc. of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16), Bologna, Italy, Nov. 2016.

[125] G. Smits, M.-J. Lesot, and O. Pivert. Vocabulary elicitation for informative descriptions of classes. In *joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCSI'17)*, 2017.

[126] G. Smits and O. Pivert. A fuzzy-summary-based approach to faceted search in relational databases. In *Advances in Databases and Information Systems*, volume 7503, pages 357–370, 2012.

[127] G. Smits and O. Pivert. Linguistic and graphical explanation of a cluster-based data structure. In *Scalable Uncertainty Management*, pages 186–200. Springer, 2015.

[128] G. Smits, O. Pivert, and T. Girault. Reqflex : fuzzy queries for everyone. *Proceedings of the VLDB Endowment*, 6(12) :1206–1209, 2013.

[129] G. Smits, O. Pivert, and A. Hadjali. Fuzzy cardinalities as a basis to cooperative answering. In *Flexible Approaches in Data, Information and Knowledge Management*, pages 261–289. Springer, 2014.

[130] G. Smits, O. Pivert, H. Jaudoin, and F. Paulus. Aggrego search : Interactive keyword query construction. In *EDBT*, pages 636–639, 2014.

[131] G. Smits, O. Pivert, and V. Thion. Connected keywords. In *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*, pages 112–120. IEEE, 2015.

[132] G. Smits, R. R. Yager, and O. Pivert. Interactive data exploration on top of linguistic summaries. In *In Proc. of the 26th IEEE International Conference on Fuzzy Systems (FuzzIEEE'17)*, 2017.

[133] Y. Takahashi. Fuzzy database query languages and their relational completeness theorem. *IEEE transactions on knowledge and data engineering*, 5(1) :122–125, 1993.

[134] R. Thomopoulos, P. Buche, and O. Haemmerlé. Hierarchical fuzzy sets to query possibilistic databases. In *Handbook of Research on Fuzzy Information Processing in Databases*, pages 299–324. IGI Global, 2008.

[135] L. Ughetto, W. A. Voglozin, and N. Mouaddib. Database querying with personalized vocabulary using data summaries. *Fuzzy Sets and Systems*, 159(15) :2030–2046, 2008.

[136] A. Urrutia, L. Tineo, and C. Gonzalez. FSQL and SQLf : Towards a standard in fuzzy databases. In J. Galindo, editor, *Fuzzy Databases." In Handbook of Research on Fuzzy Information Processing in Databases*, pages 270–298. Information Science Reference, Hershey, PA, USA, 2008.

[137] J. Vergne and E. Giguet. Regards théoriques sur le tagging. *Actes de Traitement Automatique des Langues Naturelles (TALN'98)*, pages 22–31, 1998.

[138] S. Walter, C. Unger, P. Cimiano, and D. Bär. Evaluation of a layered approach to question answering over linked data. *The Semantic Web–ISWC 2012*, pages 362–374, 2012.

[139] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. Blurme : Inferring and obfuscating user gender based on ratings. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 195–202. ACM, 2012.

[140] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 13(8) :841–847, 1991.

[141] R. R. Yager. Measures of specificity for possibility distributions. In *Proc. of IEEE Workshop on Languages for Automation : Cognitive Aspects in Information Processing, Palma de Mallorca, Spain*, pages 209–214, 1985.

[142] R. R. Yager. A characterization of the extension principle. *Fuzzy sets and systems*, 18(3) :205–217, 1986.

[143] R. R. Yager. On the specificity of a possibility distribution. *Fuzzy Sets and Systems*, 50(3) :279–292, 1992.

[144] L. A. Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.

[145] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3) :199–249, 1975.

[146] L. A. Zadeh. A computational theory of dispositions. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, pages 312–318. Association for Computational Linguistics, 1984.

[147] L. A. Zadeh. Fuzzy logic= computing with words. *IEEE transactions on fuzzy systems*, 4(2) :103–111, 1996.

[148] S. Zadrożny, G. De Tré, R. De Caluwe, and J. Kacprzyk. An overview of fuzzy approaches to flexible database querying. *Database Technologies : Concepts, Methodologies, Tools, and Applications*, 1, 2009.

[149] L. Zeng, L. Xu, Z. Shi, M. Wang, and W. Wu. Techniques, process, and enterprise solutions of business intelligence. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 6, pages 4722–4726. IEEE, 2006.