Interrogation de résumés de données et réparation de requêtes
Querying data summaries and repairing queries

W. A. Voglozin, G. Raschia, L. Ughetto, N. Mouaddib
Laboratoire d'Informatique de Nantes Atlantique, Université de Nantes
2 rue de la Houssinière, BP 92208, 44322 Nantes Cedex 3, France
{voglozin,raschia,ughetto,mouaddib}@lina.univ-nantes.fr

Résumé :

Les techniques du résumé de données sont aujourd'hui considérées comme un bon moyen de traiter les grandes masses de données, en particulier lorsque les valeurs précises de ces données ne sont pas nécessaires. Cependant, les résumés produits sont destinés à être exploités directement par un utilisateur humain, et il n'existe pas d'outils de traitement automatique. Dans cet article, un premier outil d'interrogation de résumés est proposé. Il fournit, sur certains attributs, une description des objets qui présentent certaines étiquettes sur d'autres attributs en exploitant une arborescence hiérarchique de résumés produite par le modèle SAINTETIQ. Il propose de plus une méthode coopérative pour modifier une requête sans réponse de manière à trouver des résultats sémantiquement proches de la requête.

Mots-clés :

résumé de données, interrogation flexible, requêtes coopératives

Abstract:

Data summarization techniques are now considered as accurate tools to handle huge databases, in particular when precise values of the data are not needed. However, the summaries are to be used directly by a human user, and no tool is provided to efficiently and automatically use them. In this paper, a first summary querying tool is proposed. By using the hierarchy of summaries produced by the SAINTETIQ process, it provides a description of objects depending on the linguistic labels specified as selection criteria. Moreover, automatic and cooperative methods are proposed to repair responseless queries.

Keywords:

data summarization, flexible querying, cooperative querying

1 Introduction

Dans le domaine des bases de données, les volumes atteints aujourd'hui rendent nécessaire une meilleure exploitation des données. Parmi les procédés dédiés à cet objectif, l'on retrouve les techniques de résumé de données ([3, 6, 7]). En particulier, le modèle SAINTETIQ [7] montre que les résumés linguistiques constituent un moyen d'appréhender le contenu d'une table de données. Grâce aux termes linguistiques fournis par des connaissances de domaine, ce modèle présente l'avantage d'adopter une représentation intelligible des résumés. Leur construction est incrémentale et réalise une classification hiérarchique des données vues à travers des connaissances de domaine, décrivant ainsi fidèlement les données.

Cependant, une fois les résumés obtenus, se pose le problème de leur utilisation. Bien que l'interprétation d'un résumé soit simple, elle devient difficilement envisageable pour un nombre élevé de résumés. Nous proposons ici d'exploiter l'aspect hiérarchique des résumés de SAINTETIQ dans le cadre de l'interrogation de la relation sous-jacente aux résumés. La théorie des sous-ensembles flous, utilisée dans le processus de résumé, permet une interrogation flexible vis-à-vis des données (voir [10] pour cet aspect). Ainsi, la méthode d'interrogation que nous présentons cherche à capitaliser le travail de résumé pour faire de l'interrogation flexible à moindre coût. Les recherches effectuées pour répondre aux requêtes emploient des opérateurs ensemblistes binaires et offrent une efficacité certaine.

Le mécanisme d'interrogation offre également les moyens de détecter les raisons de l'échec d'une requête. Partant de cette détection, nous introduisons un aspect coopératif dans l'interrogation à base de résumés, à savoir la réparation de requêtes. L'objectif visé est de fournir des réponses satisfaisantes même lorsque la requête n'admet pas de résultat dans le sens strict considéré auparavant. Il est ainsi possible, dans un mode interactif, de présenter à l'utilisateur les *raisons* de l'échec de sa requête et de lui soumettre des requêtes alternatives. En outre, la réparation peut être faite automatiquement par

un algorithme que nous présentons également.
La prochaine section donne un bref aperçu des résumés du modèle SAINTETIQ, l'accent étant mis sur les éléments du modèle utilisés pour l'interrogation. La section suivante décrit le fonctionnement du mécanisme de requête lorsque les résultats présentés doivent répondre de manière stricte à la requête. Ensuite, la réparation de requêtes est explicitée dans ses formes interactive et automatique.

2 Résumés du modèle SAINTETIQ

Le modèle SAINTETIQ [7] a pour objet la construction de résumés destinés à appréhender, de manière synthétique et à partir de regroupements, l'information présente au sein d'un grand ensemble de données constitué des tuples d'une relation de base de données.
Les résumés sont découverts au cours d'un processus d'apprentissage qui, à chaque instant, rend compte des groupes logiques qui semblent exister au vu des données déjà considérées. De ce fait, le processus est incrémental et les résumés construits forment une hiérarchie, représentée par un arbre, dont la relation d'ordre est basée sur le caractère spécifique (ou général) des résumés. Un résumé peut être exprimé par son extension, c'est-à-dire la liste des n-uplets qui le composent, ou par son intension. Dans ce dernier cas, les caractéristiques (étiquettes linguistiques) des n-uplets sont explicitées.

Exemple 1 :
Soit la relation R = (épaisseur, dureté, température) de plaques, manufacturées par une usine métallurgique, caractérisées par l'épaisseur en millimètres jusqu'à 50 mm, la dureté (sans unité) sur l'échelle B du test standard Rockwell et la température de fusion des métaux ou alliages pour le laminage à chaud.
Soient les enregistrements t_l, t_b et t_c (tableau 1) d'alliages de cuivre (laiton, bronze et cuivre renforcé à l'arsenic). Leur expression, après traduction en utilisant les variables linguistiques de la figure 1, les conduit à être couverts par le même résumé [1] $z_1 = \langle$moyen, doux, modéré$\rangle$. On note néanmoins que t_b est aussi couvert par $z_2 = \langle$mince, doux, modéré$\rangle$, de même que t_c se retrouve dans $z_3 = \langle$moyen, dur, modéré$\rangle$.
Plus haut dans la hiérarchie, z_1 et z_2 ont comme parent commun $z_4 = \langle$moyen + mince, doux, modéré$\rangle$. À un niveau supérieur, un résumé z_5 couvrant z_3 et z_4 (et donc également z_1 et z_2), aurait pour expression en intension $z_5 = \langle$moyen + mince, doux + dur, modéré$\rangle$.

[1] La notation est simplifiée dans cet exemple en faisant abstraction des degrés de satisfaction.

Une propriété intéressante, utilisée en section 3.2, découle de l'organisation hiérarchique : une étiquette apparaît dans un résumé seulement si elle apparaît dans l'un des résumés fils. Elle permet, en testant la présence d'une étiquette, de réduire rapidement l'espace de recherche par des coupures de branches. Notons que l'intension des résumés non-feuilles peut faire apparaître des attributs multivalués, par exemple $z = \langle 0.7/\text{fin} + 1.0/\text{mince} + 0.5/\text{moyen}, \ldots \rangle$. Pour plus d'informations sur la formation des concepts, voir [8].

3 Interrogation des résumés

Cette section traite de l'interrogation, de l'expression des requêtes à la formulation des résultats. À des fins d'illustration, nous reprenons la relation R de l'exemple 1 de même que les variables linguistiques de la figure 1.

3.1 Expression d'une requête

L'approche présentée dans ce papier vise à caractériser les données satisfaisant certains critères de sélection. Elle apporte une réponse à des questions telles que « comment sont les matériaux dont l'épaisseur est décrite par "fin" ? » ou « comment sont les matériaux à température de fusion élevée et de dureté moyenne ? ».
Dans le prototype développé, les questions sont formulées par le biais d'une interface intuitive qui compose la requête dans un langage ad'hoc dont l'opération élémentaire est une description des n-uplets. La réponse est donc un ensemble de descripteurs pour chaque attribut. A titre d'exemple, les requêtes associées aux questions ci-dessus sont respectivement :

Q_1 : DESCRIBE ON température, dureté
WITH épaisseur IN (fin)
Q_2 : DESCRIBE ON épaisseur
WITH température IN (élevé)
AND dureté IN (doux)

D'un point de vue formel, la formulation d'une requête consiste, pour chaque attribut servant de critère de sélection, en la donnée d'un ensemble

Tableau 1 – *Exemple de traduction d'enregistrements.*

Alliage	N-uplet	Traduction
UZ40	$t_l = \langle 10, 38, 900 \rangle$	$t_{l1} = \langle$ moyen, doux, modéré $\rangle$
CuSn12	$t_b = \langle 8, 40, 850 \rangle$	$t_{b1} = \langle$ moyen, doux, modéré $\rangle$, $t_{b2} = \langle$ mince, doux, modéré $\rangle$
CuAs05	$t_c = \langle 12, 44, 896 \rangle$	$t_{c1} = \langle$ moyen, doux, modéré $\rangle$, $t_{c2} = \langle$ moyen, dur, modéré $\rangle$

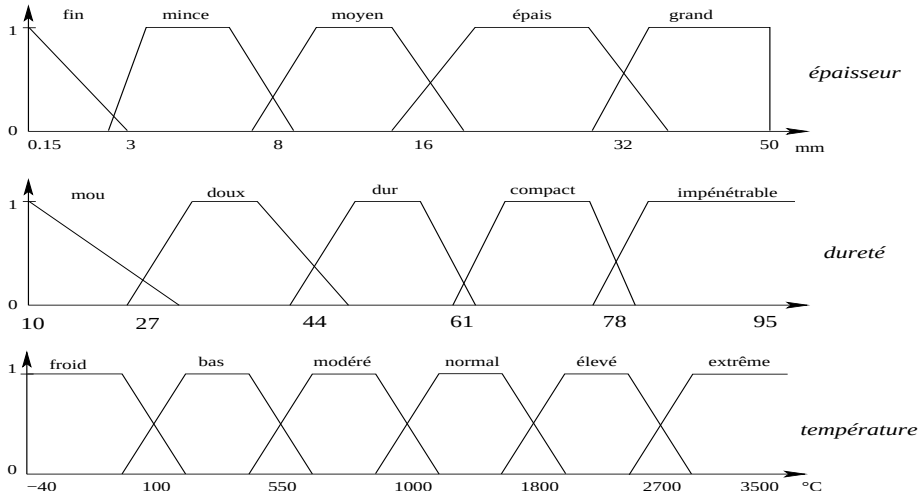

Figure 1 – *Variables linguistiques définies sur l'épaisseur, la dureté et la température.*

de descripteurs dits *descripteurs requis*. Cet ensemble, pour un attribut A_i, est noté C_i. Soient :
- S un ensemble d'attributs ;
- $\mathsf{R}(S)$ la relation dont les n-uplets sont résumés ;
- Q une requête (par exemple Q_1 ou Q_2) ;
- A un attribut requis c'est-à-dire faisant partie de la requête ($A \in S$) ;
- d une caractérisation floue.

On note X l'ensemble des attributs requis (attributs d'entrée) et Y l'ensemble des attributs sur lesquels porte la description (attributs de sortie). Par défaut, Y est le complément de X dans S ($Y = S \setminus X$). d est un descripteur requis si l'attribut A_i associé à sa variable linguistique est requis ($A_i \in X$) et si $d \in C_i$.

Exemple 2 :
Considérons les requêtes Q_1 et Q_2 ci-dessus.
Pour Q_1, $X = \{$épaisseur$\}$, $Y = \{$dureté, température$\}$ et l'on a $C_{\mathtt{ep}} = \{$fin$\}$.
Pour Q_2, $X = \{$dureté, température$\}$, $Y = \{$épaisseur$\}$, $C_{\mathtt{dur}} = \{$moyen$\}$, $C_{\mathtt{temp}} = \{$élevé$\}$.

Lorsque plusieurs descripteurs requis existent pour le même attribut, ils sont pris en compte de manière disjonctive. Ainsi, Q_3 = « DESCRIBE ON épaisseur, température WITH dureté IN (doux, mou) » traduit la condition de sélection « dureté = doux $\vee$ dureté = mou ». Dans le cas de plusieurs attributs requis, les conditions sont combinées de manière conjonctive. La requête Q_4 = « DESCRIBE ON température WITH épaisseur IN (moyen, épais) AND dureté IN (compact) » traduit la condition « (épaisseur = moyen $\wedge$ dureté = compact) $\vee$ (épaisseur = épais $\wedge$ dureté = compact) ». Ceci exclut des résultats des résumés qui ne présenteraient que la dureté (ou l'épaisseur) adéquate.

3.2 Évaluation d'une requête

Cette section traite de l'appariement entre un résumé particulier et une requête posée. L'issue en est une évaluation de la correspondance du résumé vis-à-vis de la requête, indiquant si le résumé participe au résultat.

La correspondance est une comparaison ensembliste opérée attribut par attribut. Une requête Q est ainsi transformée en une proposition lo-

gique P utilisée pour qualifier le lien résumé-requête. La proposition P est sous forme normale conjonctive, chaque descripteur apparaissant comme un littéral. Par suite, un ensemble de descripteurs requis pour un attribut forme une clause dans P.

Exemple 3 :
Pour la question q_5 « quelle est la dureté des matériaux fins ou d'épaisseur moyenne et de température normale ou élevée ? », la requête correspondante est Q_5 = « DESCRIBE ON dureté WITH épaisseur IN (fin, moyen) AND température IN (normal, élevé) ».
On a pour Q_5, X = {épaisseur, température}, $C_{\texttt{ep}}$ = {fin, moyen} et $C_{\texttt{temp}}$ = {normal, élevé}. Il s'ensuit que P_5 = (fin $\vee$ moyen) $\wedge$ (normal $\vee$ élevé).

Soit v une fonction de valuation définie comme suit : un littéral d d'une proposition P (dérivée d'une requête Q) sera positivement valué ($v(d) = \texttt{TRUE}$) si d apparaît dans l'intension du résumé z en cours d'appariement. On notera $v(P(z))$ la valuation de la proposition P dans le contexte de z.
Soit $\mathcal{L}_{A_i}(z)$ l'ensemble des descripteurs de z sur l'attribut A_i (par exemple, $\mathcal{L}_{A_i}(z)$ ={mince, moyen, épais} pour l'attribut *épaisseur*). Une première interprétation de P par rapport à Q permet d'écarter les résumés qui ne satisfont pas la proposition P. Ces résumés se distinguent par l'absence des descripteurs requis dans $\mathcal{L}_{A_i}(z)$. Cependant, l'exemple ci-dessous montre qu'une valuation positive de P dans le contexte d'un résumé z ne garantit pas que z est un résultat correct de la requête.

Exemple 4 :
Le tableau 2 montre les caractéristiques des matériaux d'un résumé z_0 et le résumé z_0 lui-même. Dans le cas où la correspondance de z_0 avec Q_5 (voir l'exemple 3) est évaluée, la valuation obtenue est $v(P_5(z_0)) = \texttt{TRUE}$, mais aucun matériau ne répond effectivement à la question q_5.

Un test de correspondance entre un résumé z et une requête Q conduit à l'un des trois cas suivants :

- **Cas 1** : aucune correspondance. $v(P(z)) = \texttt{FALSE}$. Pour un ou plusieurs attributs, z ne présente aucun des descripteurs requis par Q.
- **Cas 2** : correspondance exacte. z valide la proposition P et correspond à la sémantique de Q. L'expression suivante est vérifiée : $v(P(z)) = \texttt{TRUE}$ et $\forall i,\ \mathcal{L}_{A_i}(z) \subseteq C_i$.

Tableau 2 – *Exemple de résumé*

	épaisseur	*température*
ct_1	mince	extrême
ct_2	moyen	extrême
ct_3	épais	élevé
z_0	mince, moyen, épais	extrême, élevé

- **Cas 3** : cas d'indécision. Il existe au moins un attribut A_i pour lequel z dispose d'un ou plusieurs descripteurs autres que ceux requis par Q : $\exists i, \mathcal{L}_{A_i}(z) - C_i \neq \emptyset$. L'existence de descripteurs requis pour chaque attribut indique seulement, sans le garantir, *qu'il est possible de trouver des résultats dans le sous-arbre de racine* z. L'exploration du sous-arbre s'impose donc pour trouver ces résultats. On notera que ce troisième cas ne se produit plus au niveau des feuilles : l'indécision est toujours levée car les attributs y sont monovalués.

Ces cas de figures reflètent des comparaisons ensemblistes par attribut mettant en jeu l'ensemble $\mathcal{L}_{A_i}(z)$ des descripteurs de z sur A_i et l'ensemble C_i des descripteurs requis pour A_i.

3.3 Algorithme d'exploration

Cette section présente l'algorithme qui applique à l'ensemble des résumés, organisé en hiérarchie, la procédure d'appariement d'un résumé à une requête décrite dans la section précédente. Dans cette vision plus globale, l'algorithme exploite les liens hiérarchiques entre résumés.
Puisque la sélection doit prendre en compte tous les résumés désignés par la requête, il est nécessaire que l'exploration de la hiérarchie de résumés soit complète. Le parcours adopté est un parcours en profondeur qui s'appuie sur une propriété de la hiérarchie pour assurer la complétude de l'exploration. Le modèle SAINTETIQ garantit en effet d'une part, qu'un descripteur trouvé à un nœud existe dans au moins un nœud fils et d'autre part, que tout descripteur attaché à un résumé est également attaché à chaque parent de ce résumé. Il s'ensuit qu'un descripteur absent à un point de l'arbre est absent du sous-arbre ayant ce point comme

racine.
La propriété ainsi établie permet un élagage rapide des branches qui seront, pour une requête particulière, vides de tout résultat. Dans tous les cas, l'exploration se fait sur une portion de l'espace de recherche. On notera qu'elle est complète et exacte ; elle ne fournit que les résultats appropriés.
L'algorithme 1 décrit l'exploration et la fonction de sélection avec les hypothèses suivantes :
– la fonction *Sélection* retourne une liste indifférenciée de résumés en résultat ;
– la fonction *Corr* symbolise le test de correspondance de la Section 3.2 ;
– l'opérateur binaire '+' est l'opérateur de concaténation des listes ;
– la fonction *Ajouter* construit une liste ;
– L_{res} est une variable locale ; elle représente la liste, initialement vide au début de la fonction, des résultats trouvés dans le sous-arbre exploré.

Algorithme 1 Fonction Sélection(z, Q)

$L_{res} \leftarrow \langle\rangle$
si Corr$(z, Q) =$ *indécision* **alors**
  pour chaque résumé z_f fils de z **faire**
    $L_{res} \leftarrow L_{res}+$ Sélection(z_f, Q)
  fin pour
sinon
  si Corr$(z, Q) =$ *exacte* **alors**
    Ajouter(z, L_{res})
  fin si
fin si
retourner L_{res}

3.4 Expression des résultats

Les résultats présentent la liste des étiquettes qui décrivent les données sur les attributs de sortie (voir section 3.1). Dans l'éventualité où une requête fait intervenir plusieurs descripteurs requis pour deux ou plusieurs attributs, les résumés résultats peuvent différer quant aux « raisons » de leur sélection. Par exemple, pour une requête sur *épaisseur* et *dureté*, avec comme descripteurs requis {*fin*, *mince*} et {*mou*, *doux*} respectivement, certains résumés présenteront *fin* et *mou*, d'autres présenteront *fin* et *doux* et ainsi de suite.
Afin de faciliter la lecture des résultats, les résumés présentant les mêmes ensembles de descripteurs pour tous les attributs requis fournissent une liste de descripteurs par attribut :
– fin, mou ⇒ froid ;
– fin, doux ⇒ froid, bas ;
Cette méthode présente plusieurs avantages :
– l'existence de certaines combinaisons de descripteurs requis au sein des données peut être facilement vérifiée (*fin*, *mou*, *bas* est absent) ;
– les catégories de résumés résultats sont différenciées (*bas* n'est dû qu'à {{*fin*}, {*mou*}}) : on sait précisément à quels résumés on doit la présence d'un descripteur de sortie ;
– la présentation des résultats reste synthétique et permet si nécessaire, d'avoir une unique description globale.

4 Réparation de requêtes

L'objectif visé est d'offrir une réponse en l'absence de résumés correspondant strictement à la requête. La réparation de requêtes a déjà été implémentée dans le contexte d'un médiateur par Bidault et al. [1, 2]. De par son aptitude à faciliter la réécriture de requêtes, cette caractéristique fait partie des comportements coopératifs listés par Gaasterland et al. dans [4].
La réparation s'appuie sur l'idée qu'il pourrait exister des résultats sémantiquement proches de ceux recherchés par l'utilisateur. Pour trouver ces résultats, la requête est modifiée en utilisant des informations préétablies ou les résumés eux-mêmes.

4.1 Modification de requête

Cette procédure intervient à chaque fois que l'exploration ne peut plus progresser au-delà d'un nœud z. Dans ce cas, la procédure de sélection en 3.3 échoue pour le sous-arbre de z : la liste des résumés sélectionnés reste vide car certains descripteurs requis sont absents de z.
La première stratégie (section 4.3) consiste à remplacer les descripteurs absents par d'autres descripteurs, dans une limite fixée par la distance en section 4.2. On obtient alors une nouvelle requête Q^*, dite requête de substitution. Cependant, aucune garantie ne peut être don-

née quant à l'existence de résultats pour Q^*. Une possibilité de substitutions est liée aux variables linguistiques : les descripteurs absents sont remplacés par les descripteurs les plus proches, ce qui nécessite qu'un ordre existe sur le domaine de la variable concernée. Pour contourner cet obstacle, une autre possibilité, plus générale, consiste à définir explicitement une matrice de substitutions pour chaque variable. Dans ce cas, les poids affectés aux permutations permettraient un traitement plus fin des résultats en privilégiant certaines substitutions.

Dans tous les cas, un descripteur d remplacé par d^* au niveau d'un nœud z ne peut plus servir de substituant à un autre descripteur dans le sous-arbre de racine z. En effet, l'échec de d au résumé z garantit qu'il n'existe pas de résumé dans le sous-arbre de racine z où d serait positivement valué dans l'évaluation de la requête.

La deuxième stratégie (section 4.4), plus flexible car guidée par la hiérarchie de résumés, est adaptée à un mode interactif. Les résultats sont cette fois de nouvelles requêtes plutôt que des résumés.

4.2 Distance entre requêtes

Afin d'éviter une série de modifications conduisant à des résultats inappropriés, nous introduisons une mesure de distance entre requêtes.

Admettons qu'une requête Q peut être associée à une chaîne de bits S dans laquelle un bit marque la présence ou l'absence, dans la requête, du descripteur associé à cette position suivant un ordre fixé.

Définition : Soient Q et Q^* deux requêtes respectivement associées aux chaînes de bits S et S^*. La distance entre Q et Q^*, notée $d(Q, Q^*)$, est le nombre de bits non nuls de $S \oplus S^*$ (S XOR S^*). Ce nombre rend compte des modifications (insertions ou suppressions de descripteur) nécessaires pour obtenir Q^* à partir de Q. La distance ainsi définie, qui est une distance de Hamming [5], satisfait les propriétés suivantes :

1. $d(Q, Q) = 0$
2. $Q \neq Q^* \Rightarrow d(Q, Q^*) > 0$
3. $d(Q, Q^*) = d(Q^*, Q)$
4. $d(Q, Q^*) < \sum_{A_i \in C} |D_{A_i}|$ où D_{A_i} est l'ensemble des descripteurs de la variable linguistique sur l'attribut A_i.

Toutefois, cette distance n'est pas suffisamment fine pour permettre un processus automatique. Il peut s'avérer nécessaire que l'utilisateur guide la suite de la recherche : en effet, la distance ci-dessus place au même niveau des requêtes Q_1 et Q_2 par rapport à une requête Q sans distinguer la *proximité* des étiquettes. Par exemple, si une requête porte sur des matériaux à température décrite par « bas » (voir figure 1), des requêtes modifiées pour rechercher respectivement « froid » et « élevé » sont considérées équivalentes.

4.3 Algorithme de substitution de requêtes

Cette section présente la procédure de modification des requêtes décrite par l'algorithme 2. La procédure est une variante de l'algorithme 1 (voir section 3.3) car elle ajoute au cas 1 un traitement supplémentaire représenté par la fonction *Modifier*. L'emploi des informations disponibles (variables linguistiques ou matrice de substitution) intervient au niveau de cette fonction.

Q_{ref} représente la requête originale exprimée par l'utilisateur, et utilisée par la fonction *Modifiable* pour garantir que la propriété 4 (section 4.2) sera toujours vérifiée après la modification. Q est la requête en cours d'évaluation, initialement équivalente à Q_{ref}.

Algorithme 2 Fonction Sel-Mod(z, Q, Q_{ref})

$L_{res} \leftarrow \langle\rangle$
si Corr$(z, Q) =$ *indécision* **alors**
  pour chaque résumé z_f fils de z **faire**
    $L_{res} \leftarrow L_{res}+$ Sel-Mod(z_f, Q)
  fin pour
sinon
  si Corr$(z, Q) =$ *exacte* **alors**
    Ajouter(z, L_{res})
  sinon {pas de correspondance, mais le résumé peut être acceptable}
    si Modifiable(Q, Q_{ref}) = `TRUE` **alors**
      Q^* = Modifier(Q, z)
      $L_{res} \leftarrow L_{res}+$ Sel-Mod(z, Q^*, Q_{ref})
    fin si
  fin si
fin si

retourner L_{res}

4.4 Modification guidée par les résumés

Lorsqu'une requête échoue, l'évaluation permet de détecter les *raisons* de l'échec, une *raison* désignant un attribut requis. Il suffit de mettre en évidence les attributs requis dont les descripteurs n'apparaissent dans aucun résumé de l'espace exploré.
Pendant une recherche-sélection, si la même requête Q_0 échoue pour tous les fils d'un nœud z_0, on peut considérer z_0 comme la meilleure approximation d'une réponse à Q_0 dans la branche menant à z_0. L'exploration se faisant de proche en proche, du fait que z_0 est le dernier point examiné, on déduit qu'une requête Q_1 ayant z_0 comme résultat est également proche de Q_0 à condition de partager les mêmes attributs requis.
Les requêtes semblables à Q_1 (c'est-à-dire considérées comme proches de Q_0 et issues de points d'échec), offrent une garantie de résultat, ce qui n'était pas le cas pour Q_0.
Toutefois, ce fonctionnement requiert de déterminer, par l'algorithme 1, que la requête n'a pas de résultat, et, par la suite, rechercher les points d'échec afin de déduire les requêtes modifiées. L'algorithme 3 supprime la nécessité de deux explorations en constituant deux listes résultats : celle des résumés sélectionnés (L_{res}) et celles des résumés à la base de requêtes modifiées (L_Q), la dernière n'étant utile que si la première est vide.

4.5 Expression des résultats

La modification de requêtes, telle que décrite, rend possible l'obtention de résultats issus de plusieurs requêtes différentes. Ces requêtes-source sont plus ou moins proches de la requête initiale. Il devient possible d'effectuer un classement des résultats et donc d'exprimer une préférence de certains résultats par rapport à d'autres. Il semble naturel d'une part que le critère de classement soit la distance des requêtes relativement à la requête utilisateur, et d'autre part, que les résultats *héritent* de la distance associée à la requête dont ils sont un résultat.
On notera que la réparation apparaît comme un relâchement des contraintes de la recherche car la nouvelle requête obtenue est plus générale. Cependant, chaque substitution d'une requête Q par une requête Q^* est locale comme l'explicite l'exemple 5 ci-après. La « localité » de la réparation se justifie par le fait que l'échec survient à un point précis, et par le nombre des échecs qui, sans cette « localité », ferait perdre en efficacité.

Algorithme 3 Fonction Sel-Guidée(z, myCorr, Q)

$L_{res} \leftarrow \langle\rangle$
si myCorr $=$ *exacte* **alors**
  Ajouter(z, L_{res})
sinon
  si myCorr $=$ *indécision* **alors**
    pour chaque résumé z_f fils de z **faire**
      Corr$_f$ $=$ Corr(z_f, Q)
      $L_{res} \leftarrow L_{res} +$ Sel-Guidée(z_f, Corr$_f$, Q)
    fin pour
    si $L_{res} = \langle\rangle$ **alors**
      Ajouter(z_f, L_Q)
    fin si
  fin si
fin si
retourner L_{res}

Exemple 5 :
Considérons la requête initiale Q_0 adressée à une hiérarchie de racine z_0 et deux points d'échec arbitraires, z_1 et z_2. Les requêtes de substitution Q_1 et Q_2 sont alors composées pour chacun de ces points.
On distingue, suivant la requête dont ils sont un résultat, trois types de résumés sélectionnés : ceux de l'arbre z_0, ceux du sous-arbre z_1 et ceux du sous-arbre z_2.
Les résumés sélectionnés par Q_1 ne se retrouveront *que* dans le sous-arbre de sommet z_1 même s'il existe ailleurs dans l'arbre, des résumés qui auraient été sélectionnés si Q_1 avait été adressée à toute la hiérarchie (à partir de z_0).

En résumé, l'articulation des différentes sections pour l'algorithme 3, choisi pour ses avantages, est la suivante :

1. évaluation de la requête initiale Q ;
2. absence de résultat, exécution de la réparation ;
3. choix, par la mesure de distance ou par interaction, de la meilleure requête de substitution Q^* ;
4. évaluation de Q^* ;
5. expression des résultats.

5 Conclusion

Nous avons présenté une approche de l'interrogation basée sur des résumés linguistiques de

données structurées dans un but de caractérisation des données. Cette approche, motivée par l'augmentation des tailles des bases de données, tire profit des résumés et offre des résultats permettant de décrire les données sur plusieurs niveaux de détails.

D'un point de vue algorithmique, le mécanisme de recherche-sélection explore une hiérarchie de résumés. À chaque résumé visité, il effectue une comparaison ensembliste avec la requête sur la base de descripteurs issus d'un vocabulaire prédéfini. Le résultat de cette comparaison détermine si le résumé fera partie de la réponse mais il conditionne aussi l'exploration d'une partie de la hiérarchie. En conséquence de l'utilisation d'opérateurs binaires, l'outil d'interrogation reste booléen. Il se distingue par le recours, d'une part, aux résumés pour caractériser des données et d'autre part, aux liens hiérarchiques entre résumés. Un prototype a été développé et a permis de valider expérimentalement la méthode évoquée ici.

Une extension de cette méthode stricte est également présentée. Elle permet de rechercher les données, éventuellement satisfaisantes, qui pourraient être retournées en résultat à une requête sans réponse. Il en découle deux options, l'une consistant à exploiter des informations disponibles, et l'autre, plus sûre en termes de résultats, utilisant les résumés pour déterminer les meilleures alternatives à la requête initiale.

Le lien immédiat entre un résumé et les n-uplets relationnels qu'il couvre conduit à considérer nos travaux comme une étape intermédiaire vers un processus d'interrogation flexible, plus complet, mettant en jeu les enregistrements et tirant parti des degrés de satisfaction.

La poursuite de ces travaux ciblera en premier lieu l'expressivité du langage d'interrogation. Il s'agit notamment de déterminer comment exprimer des préférences ou des priorités, mentionnées par Rocacher dans [9] comme propres à l'interrogation flexible ou encore d'intégrer la description des données dans un langage.

Références

[1] A. Bidault, C. Froidevaux, and B. Safar. Repairing queries in a mediator approach. In *Proc. of ECAI'00*, pages 406 – 410.

[2] A. Bidault, C. Froidevaux, and B. Safar. Similarity between queries in a mediator. In *Proc. of ECAI'02*, pages 235–239.

[3] P. Bosc, D. Dubois, O. Pivert, and H. Prade. Résumés de données et ensembles flous - principes d'une nouvelle approche. In *LFA'2000*, pages 333–340.

[4] T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *J. of Intelligent Systems*, **1**(2):123–157, 1992.

[5] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, **27**(2):147–160, Apr. 1950.

[6] D. H. Lee and M. H. Kim. Database summarization using fuzzy ISA hierarchies. *IEEE Trans. on Systems, Man and Cybernetics-Part B : Cybernetics*, **27**:68–78, Feb. 1997.

[7] G. Raschia. SAINTETIQ *: une approche floue pour la génération de résumés à partir de bases de données relationnelles*. Thèse de doctorat, Université de Nantes, Dec. 2001.

[8] G. Raschia and N. Mouaddib. SAINTETIQ : a fuzzy set-based approach to database summarization. *Fuzzy Sets And Systems*, **129**:137–162, 2002.

[9] D. Rocacher. On fuzzy bags and their application to flexible querying. *Fuzzy Sets And Systems*, **140**:93–110, 2003.

[10] W. A. Voglozin, G. Raschia, L. Ughetto, and N. Mouaddib. Querying the SAINTETIQ summaries – a first attempt. In *Proc. of (FQAS2004)*, 2004.