Efficient Generation of Reliable Estimated Linguistic Summaries

Grégory Smits, Pierre Nerzic, Olivier Pivert, Marie-Jeanne Lesot

Efficient Generation of Reliable Estimated Linguistic Summaries

Grégory Smits[1], Pierre Nerzic[1], Olivier Pivert[1] and Marie-Jeanne Lesot[2]
[1] Univ Rennes, IRISA - UMR 6074, F-22305 Lannion, France
Email: {gregory.smits,pierre.nerzic,olivier.pivert}@irisa.fr
[2] Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
Email: marie-jeanne.lesot@lip6.fr

***Abstract*—Summarizing data with linguistic statements is a crucial and topical issue that has been largely addressed by the soft computing community. The goal of summarization is to generate statements that linguistically describe the properties observed in a dataset. This paper addresses the issue of efficiently extracting these summaries and rendering them to the final user, in the case where the data to be summarized are stored in a relational data base: it proposes a novel strategy that leverages the statistics about the data distribution maintained by the database system. This paper shows that reliable summaries can be very efficiently estimated based on these statistics only and without any costly data access. Additionally, it proposes a visualization of the set of extracted summaries that offers a fruitful interactive exploration tool to the user. Experiments performed on two real data bases show the relevance and efficiency of the proposed approach: with a negligible loss of accuracy, we provide the first linguistic summarization approach whose processing time does not depend on the size of the dataset. The generation of estimated linguistic summaries takes less than one second even for dataset containing millions of tuples.**

I. Introduction

Many professional activities rely on the analysis of data with the aim of making decisions on the basis of facts or item descriptions. As the volume and heterogeneity of the datasets a user has to deal with are continuously increasing, the generation of meaningful but concise views of the data now constitutes a crucial task. Data summarization is a problem that has been deeply studied by both the DataBase (DB) community and the soft-computing community. Roughly speaking, the DB community has mainly addressed this issue in two ways: schema summarization [26] and data/table selection [7], [25].

The soft-computing community possesses a long history in data summarization, see e.g. the recent overview in [2]. The specificity of its approach comes from the fact that the trends observed in the data are expressed in a linguistic way. These trends are usually structured wrt. a syntactic protoform, i.e. a template, that is generally of one of the two following forms in the case of numerical data [24], [12]: *Q of the X's are/have P* or *Q of the PX's are/have also P′*, where X is the universe of discourse, i.e. the data to summarize, Q is a quantifier and the Ps are summarizers generally expressed by means of conjunctions of linguistic labels. For example, for a data set describing commercial flights, these protoforms can be instantiated as *most of the flights have short airtime* or *a minority of the flights from western airports have a very long arrival delay*.

As detailed in Section II, that also provides a more formal description of the linguistic summarization task, the latter faces two major, related, issues, namely the computational cost of the summary extraction and the huge quantities of sentences that can be extracted.

This paper proposes to address these challenges in the case where the data to be summarized are stored in a Relational DataBase (RDB). When the summarization task concerns massive relational data, it is obviously inconceivable to scan or query the data to assess all the candidate linguistic statements. Moreover, contrary to DB querying where the exact answer to the user's query has to be computed, users expect from a data summary to obtain an insight about the data distribution and are thus willing to accept some controlled imprecision for the benefit of a better scalability.

We introduce in this paper a summarization strategy that aims at efficiently providing the user with linguistic descriptions of the data. Instead of scanning the data or executing queries to precisely quantify the cardinality of the different possible summarizers, we propose to take advantage of the RDB storage and to estimate these cardinalities using statistics about the data distribution that are maintained by any RDB Management System (RDBMS). Such statistics are mainly used by the RDBMS to derive query execution plans and to optimize selection queries. We study in this paper their interest in estimating the cardinality of a summarizer composed of a conjunction of fuzzy terms. Our goal is to show that reliable linguistic summaries can be generated in a very efficient way and without any access to the data themselves, thus allowing the management of massive data.

After having formalized the linguistic summarization task and put our approach into perspective with existing works in Section II, preliminary notions about RDBs are introduced in Section III. Section IV presents the proposed approach, detailing how DB statistics can be used to estimate the cardinalities required to extract summaries. Section V describes the derived extraction process and a measure to quantify the confidence of the generated summaries. Section VI gives the results of experiments performed on real data that show the relevance and efficiency of the proposed approach. Finally, we describe in Section VII a possible interactive tool to visualize the summaries generated by the proposed approach that eases

the identification of the data relevant characteristics.

II. Context and Related Work

This section briefly recalls the formal definition of protoform-based linguistic summaries of data, before presenting some existing approaches dealing with its main challenge of efficient extraction.

A. Linguistic Summaries Based on Protoforms

1) General Principle: Protoform-based summarization strategies [24], [10] take as input a set of data to be summarized, here denoted R, that contains the description of m tuples, $R = \{t_1, t_2, \ldots, t_m\}$ wrt. n attributes $\{A_1, A_2, \ldots, A_n\}$ that may be of a numerical or categorical type. Moreover users usually provide the definitions of linguistic labels corresponding to imprecise concepts they are interested in: they are defined as fuzzy linguistic variables, as detailed below.

The summarization process consists in projecting the data to summarize onto different conjunctive combinations of the linguistic terms. The cardinality associated with each combination of terms is also linguistically described using quantifiers, usually taken from standard a predefined list. In addition, a degree of truth is generally attached to each summary to quantify the extent to which it faithfully represents the data.

2) Vocabulary and Quantifiers: Formally, the considered vocabulary, denoted by $\mathcal{V} = \{V_1, \ldots, V_n\}$, consists of a set of linguistic variables, associated with each attribute: V_j is a triple $\langle A_j, \{v_{j1}, \ldots, v_{jq_j}\}, \{l_{j1}, \ldots, l_{jq_j}\}\rangle$ where q_j denotes the number of modalities associated with attribute A_j, v_{js} denote their respective membership functions defined on domain D_j and l_{js} their respective linguistic labels, generally adjectives from the natural language. For instance, an attribute A describing prices may be associated with $q_A = 3$ modalities, in turn associated with the labels l_{A1} ='cheap', l_{A2} ='reasonable' and l_{A3} ='expensive'.

It is assumed that the linguistic variables associated with an attribute, say A_j, define a strong partition [18]: $\forall y \in D_j$, $\sum_{s=1}^{q_j} v_{js}(y) = 1$. As a consequence, any value y can be rewritten in terms of V and y can partially satisfy only up to two modalities, that besides have to be adjacent in the case of a numerical value.

Fuzzy quantifiers [9] linguistically describe relative or absolute cardinalities; in the relative case, they can for instance be taken from the classic partition illustrated in Figure 1.

3) Protoforms and Truth Degree: Many different types of protoforms have been proposed, in particular depending on the considered type of data, e.g. numerical vs time series [3], [8], [4], [1], [15], and of the information a user may be interested in. In the case of numerical data, the two most classic syntactic structures of linguistic statements, introduced by Yager [24], [10], are defined as $Q\,R$ *are/have* P and $Q\,P\,R$ *are/have also* P', where R is the considered data, P and P' are conjunctions of terms taken from $\mathcal{V}$ and Q is a quantifier. The selection between the verbs "are" and "have" depends on the considered terms, to make the sentence as natural as possible from a linguistic point-of-view.

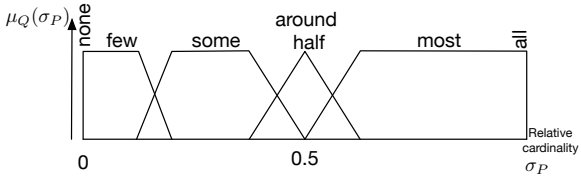

Fig. 1. Example of possible relative quantifiers

The relevance of a candidate instantiated protoform for the considered data is then measured by a truth degree that quantifies its validity with respect to R. This truth degree may be computed in the following way that relies on Zadeh's interpretation of quantified statements [27], for the first type of considered protoform:

$$\tau(Q\,R \text{ are/have } P) = \mu_Q\left(\frac{\sum_{r\in R}\mu_P(r)}{|R|}\right) \qquad (1)$$

B. Efficient Summarization

Classically, two main, related, issues arise when extracting linguistic summaries to describe a given data set: the mining computational cost and the result size. Indeed, the number of output summaries in a naive setting is overwhelming and can make it very difficult for the user to get any benefit from the produced result. Whatever the considered protoform, the search space of possible summaries is huge and considering the volume of data to manage, efficient strategies have to be defined to reduce this search space or to avoid too many costly data access and to generate the most interesting summaries only.

Many methods have been proposed to tackle these two issues: the initial approach (see e.g. [17], [11]) consisted in adopting a user-guided methodology, focusing on the instantiation of partially specified protoforms, limiting the number of data scans to identify the unspecified parts of the summaries. The exploitation of genetic algorithms has also been proposed to explore the whole search space efficiently [6].

More recent approaches focus on the issue of the result size and propose methods to *a posteriori* reduce the set of extracted summaries to the most relevant ones: they do not consider summaries individually, assessing their quality for instance through the truth degree recalled in Equation (1), but globally, in particular taking into account their relative redundancy, so as to output a compact final result. They differ by the criteria and rules they propose to prune them [16], [21].

Other proposals aim at avoiding the generation of non-relevant summaries that are later discarded, using integrated methods: some of them exploit the relations between linguistic summaries and fuzzy or quantitative association rules [13], [14], others exploit the same principle of anti-monotonicity of quality criteria, e.g. considering degree of focus or a measure inspired by the degree of appropriateness, beside the truth degree [22], [5], [23].

To the best of our knowledge, no method has been proposed to exploit the specific principles of a DB storage, as it is the case in this paper.

III. Preliminaries: Relational Data Bases

This paper focuses on the case where the data to be summarized are stored by an RDBMS in a set of relations linked by foreign key constraints. For the sake of clarity and simplicity, we thus consider that the data to summarize come from a relation R that may result from joins between different relations.

This section recalls the principle of the statistics maintained by the RDBMS and in particular the notion of selectivity degree.

A. Meta-data Table about the Tuple Distribution

For each attribute, the RDBMS maintains, transparently for the user, a meta-data table, stored in the so-called DB catalog, that describes the data distribution on the definition domain of the different attributes.

In the case of a numerical attribute, say A, the data distribution is described by a histogram of k intervals denoted by $H_A : \{h_1, h_2, \ldots, h_k\}$ and defined by their boundary values. These k intervals can be of different widths as their bounds are dynamically defined so as to guarantee that they all cover the same amount of data: narrow intervals thus describe dense areas of the definition domain.

In the case of categorical attributes, the data distribution is described by the list of the k most frequent values possessed by the tuples.

B. Associated Selectivity Degrees

For each interval (for numerical attributes) and for each frequent value (for categorical attributes), say h for simplicity, the RDBMS additionally stores and maintains a so-called *selectivity degree* that corresponds to a relative cardinality: this value in $[0, 1]$, denoted by σ_h, corresponds to an estimation of the proportion of tuples whose value falls in the interval or is equal to the given categorical value.

In the case of numerical attributes, as mentioned above, the interval widths are defined so that they cover the same amount of data. Therefore, for a histogram $H_A : \{h_1, h_2, \ldots, h_k\}$, one should theoretically have $\sigma_{h_1} = \sigma_{h_2} = \ldots = \sigma_{h_k}$. In practice, the RDBMS tries to maintain these selectivity degrees as close as possible, without achieving equality.

Given the selectivity degree of an interval h as computed by the RDBMS, the number of tuples associated with h, denoted by $|h|$, can then be computed as: $|h| = \sigma_h \times |R|$.

The RDBMS then considers that these $|h|$ tuples are uniformly distributed within the interval covered by h.

In the case of categorical attributes, the number of tuples concerned by a frequent value, say h, of selectivity σ_h, is computed as for the numerical case: $|h| = \sigma_h \times |R|$. The selectivity degree of non-frequent values is then deduced: denoting H the set of frequent values for a given attribute A, the selectivity of a non-frequent value, say h', is computed as $\sigma_{h'} = \frac{1}{N}(1 - \sum_{h \in H} \sigma_h)$, where N is the number of non frequent distinct values for A.

In the rest of the paper, the term RDB statistics is used to name these histograms and lists of frequent values associated with their selectivity degrees.

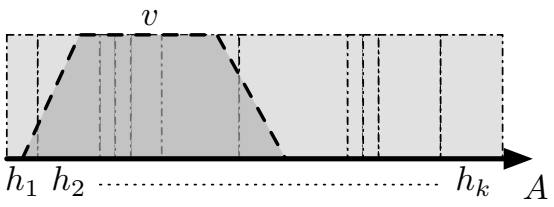

Fig. 2. Cardinality estimation of a term using a histogram about data distribution

IV. Cardinality Estimation from RDB Statistics

In this section, we show how RDB statistics maintained by any RDBMS can be used to efficiently derive interesting estimations of the cardinalities of a candidate summarizer P: it first considers the case of an atomic P, i.e. a summarizer defined as one of the linguistic labels, and the case of conjunctive summarizers, i.e. a combination of atomic labels. These estimations are then used in the algorithm proposed in Section V to establish relevant linguistic summaries of the data.

A. Case of Atomic Summarizers

This section first considers an atomic summarizer, denoted by P, associated with attribute A and fuzzy subset v. The task is to estimate the relative (wrt. the size of the considered relation) fuzzy cardinality of v based on the selectivity degrees attached to histograms (for the numerical case) and frequent categorical values (for the categorical case) in the DB meta-data. Figure 2 provides a graphical representation of this issue for the case of a numerical attribute A.

We first explain how the relative cardinality of v, denoted by σ_v, is computed in the case of a numerical attribute, thus involving the use of a histogram, to then show that the computation is mainly the same in the case of categorical values.

Let $H : \{h_1, h_2, \ldots, h_k\}$ be the histogram maintained by the RDBMS for attribute A. The principle of the computation is to determine the extent to which each interval of the histogram contributes to the computation of σ_v, and to thus sum up their selectivity degrees weighted by their respective contributions. For example, for the case represented in Figure 2, the selectivity of the third interval should be fully taken into account, whereas that of the second one should only contribute with an approximate weight of two thirds.

In order to have a tractable approach for this computation, an α-cut method is proposed: let $\alpha_1 = 1 > \alpha_2 > \ldots > \alpha_q = 0^+$ be a scale of membership degrees used to build α-cuts of v, respectively denoted by v^α.

Then for each α_i of the considered scale, one builds the fuzzy set of histogram intervals intersecting with v^{α_i}. We denote by $H_v^{\alpha_i}$ such fuzzy set of the intervals from H concerned by v^{α_i}. Each histogram interval in $H_v^{\alpha_i}$, say h, is associated with a membership degree denoted by $\mu_{H_v^{\alpha_i}}(h)$ that tells us about the overlapping ratio between h and v^{α_i}.

Based again on the hypothesis of uniform distribution of the tuples inside each histogram interval, we compute the selectivity of an α-cut in the following way:

$$f(H_v^{\alpha_i}) = \sum_{h \in H_v^{\alpha_i}} \mu_{H_v^{\alpha_i}}(h) \times \sigma_h. \quad (2)$$

It is straightforward to show that f is a capacity, as $f(\emptyset) = 0$, $f(H) = 1$ and the use of the sum aggregator makes f a non-decreasing function of its argument. The estimation of the overall selectivity of v may thus be simply computed by a Choquet integral:

$$\sigma_v = \sum_{i=2}^{q} \alpha_i \times [f(H_v^{\alpha_i}) - f(H_v^{\alpha_{i-1}})]. \quad (3)$$

In the case of a categorical attribute and thus the use of a list of frequent values and their associated selectivity degrees instead of histograms, the only difference concerns the definition of the fuzzy set $H_v^{\alpha_i}$. Instead of gathering histogram intervals overlapping with v^{α_i}, $H_v^{\alpha_i}$ contains the categorical values present in v^{α_i}. Let x be one these values, then the membership degree of x in $H_v^{\alpha_i}$ is simply equal to its membership degree in v^{α_i}, i.e. $\mu_{H_v^{\alpha_i}}(x) = \mu_v(x)$.

B. Case of Conjunctive Summarizers

When P is of a conjunctive type, $P = P_1 \wedge \ldots \wedge P_u$, the selectivity σ_P has to be estimated from the individual selectivities of its conjuncts σ_{P_i}, $i = 1..u$, respectively estimated using Equation (3) defined in the previous section. Indeed, an RDBMS only maintains statistics about tuples distribution on the different attributes individually, but not on the Cartesian product of their domains.

To estimate the cardinality of an intersection between two histogram intervals h_a and h_b of respective cardinality σ_{h_a} and σ_{h_b}, the RDBMS again applies the hypothesis of a uniform distribution of the tuples that are covered by h_a on the definition domain of the attribute concerned by h_b, hence the use of the probabilistic norm to compute the selectivity of $h_a \wedge h_b$.

Without any additional statistics about data distribution on conjunctions of properties, we cannot do better than using the same hypothesis of a uniform distribution of the tuples. The relative cardinality of a conjunctive summarizer is thus computed in the following way, for $P = P_1 \wedge \ldots \wedge P_u$:

$$\sigma_P = \prod_{i=1..u} \sigma_{P_i}. \quad (4)$$

V. Linguistic Summarization using RDB Statistics

This section first details the proposed RDB-statistics-based summarization process (Section V-A) exploiting the estimation described in the previous section. Section VI-A3 then proposes a criterion to assess the reliability of this estimation, defining a confidence degree.

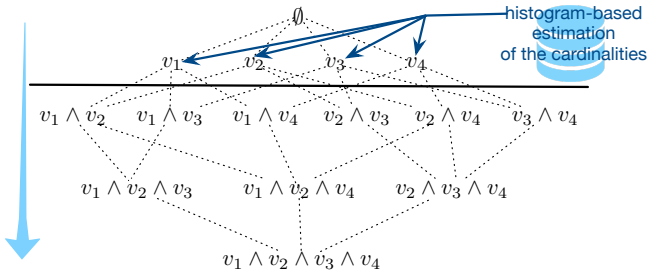

Fig. 3. Apriori-like summarization process

A. Summarization Process

As reminded in Section II, the goal of a linguistic summarization process is to generate a set of sentences that describe the different data trends. These sentences are of a predefined syntactic structure, *Q R are/have P* in our case, and use terms from the user vocabulary to form the summarizer P that may be of an atomic or conjunctive nature.

The summarization algorithm follows a principle similar to the one of the classic frequent itemset extraction algorithm APRIORI, it relies on a classical breadth first search of the lattice of the possible summarizers, lattice structured by means of an inclusion relation between the conjunctive terms. Each element of this lattice constitutes a summarizer P for which one has to compute its cardinality to then identify the quantifier that best describes it. Considering the fact that the available quantifiers form a Ruspini partition (Figure 1), we mean by "best quantifier" for a relative cardinality σ_P, the relative quantifier Q such that $\mu_Q(\sigma_P) \geq 0.5$. In the special case where two best quantifiers exist, i.e. when σ_P is the middle point between two adjacent quantifiers, then one adopts a pessimistic choice by selecting the one describing lowest cardinalities.

The search space of the possible summarizers is depicted in Figure 3. Contrary to existing approaches that rely on a scan or a querying of the data, the strategy proposed in this paper uses the statistics maintained by any RBDMS about the data distribution to estimate the cardinality of each atomic summarizers (see Section IV-A), i.e. the cardinalities σ_{v_1}, σ_{v_2}, σ_{v_3} and σ_{v_4} in Figure 3. Once these cardinalities about each term from the vocabulary estimated, no more access to the DB statistics are needed as the cardinalities for conjunctive summarizers are estimated by aggregating the estimated cardinalities of their conjuncts (see Section IV-B and Equation (4)).

The relative cardinality of a conjunctive summarizer being estimated as the product of the cardinalities of its conjuncts, it is always strictly positive. A cardinality threshold κ is thus used to stop the exploration process when the current conjunction has a very low relative cardinality, e.g. below 0.01. However, it must be underlined that contrary to association rules that focus on main trends in the data, linguistic summaries also aim at extracting information about surprising or rare summarizers, in the case of summaries associated with the quantifier 'Few'. A low value of κ should thus be considered, that can be related to the fuzzy quantifier 'almost none'.

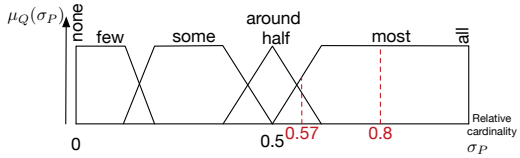

Fig. 4. Illustration of the confidence measure

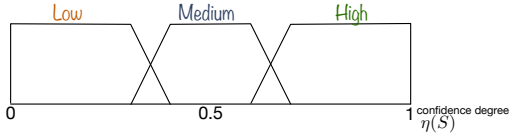

Fig. 5. Linguistic rewriting of the confidence degree

B. Confidence degree: Quantifier vs. Estimated Cardinality

The proposed linguistic statements being generated from estimated cardinalities, we consider important to be able to inform the user about the confidence the system has in each of these statements.

To this aim, we define in this section a measure that takes into account the possible imprecision of an estimated cardinality to quantify the confidence one can have in its associated linguistic statement. The confidence degree attached to a statement of the form S = $Q\,R$ *are/have* P is denoted by $\eta(S)$.

This degree depends on the estimated relative cardinality of P (i.e. σ_P). We propose to measure it as a function of its location in the membership function of the quantifier Q that best describes it: the closer this cardinality is to the middle of the core of the membership function of Q and the larger this core is, the higher the confidence. Indeed, if this cardinality falls close to the middle of the core of Q, then the imprecision of the estimated cardinality has a reduced impact on the choice of its attached quantifier . This principle is illustrated in Figure 4: one can be more confident in the linguistic rewriting of the estimated cardinality 0.8 by the quantifier *most* than for the cardinality 0.57. Indeed, a small variation in the estimation of 0.57 may turn it to 'around half'.

Formally, let Q be a quantifier of a trapezoidal shape whose 0.5-cut is the interval $[a, b]$, then the confidence measure η is defined as follows:

$$\eta(Q\,R \text{ are/have } P) = 1 - \frac{|\frac{b-a}{2} - \sigma_P|}{b-a}. \quad (5)$$

The confidence degree η defined in the unit interval may then be translated in a more interpretable way using a linguistic variable such as the one illustrated in Figure 5. We leverage this linguistic rewriting of the confidence degree in the graphical interface provided to the user described in Section VII.

VI. Experimentations

Experimentations have been conducted to assess both the relevance and the efficiency of the proposed summarization strategy based on estimated cardinalities.

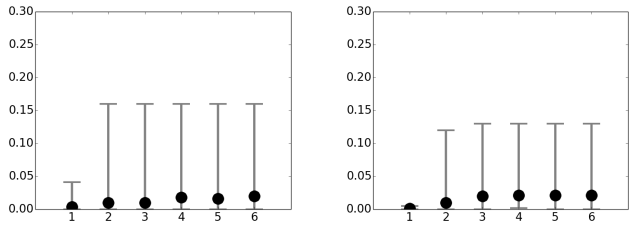

Fig. 6. Cardinality comparison (left *flights*, right *cars*)

To do so, the approach has been implemented and tested on two sets of real data[1]. The first one, denoted *flights*, contains the description of 123,534,991 commercial flights in the US from 1987 to 2008[2]. A fuzzy-partition-based vocabulary composed of 75 terms has been defined on the attributes: {*DayOfWeek, DepTime, AirTime, ArrDelay, DepDelay, Distance, Month, DayOfMonth, TaxiIn, TaxiOut, CarrierDelay, WeatherDelay, SecurityDelay, LateAircraftDelay, Origin, Dest*}. The partitions defined on the attributes *DayOfWeek*, *Month* and *DayOfMonth* respectively indicate the part of the week concerned (beginning, middle, end, weekend), the season and the part of the month.

The second dataset, named *cars*[3], contains the description of 98,562 secondhand cars. A common sense vocabulary has also been defined on the six attributes {*nbDoors, price, mileage, year, horsePower, initialNewPrice*}.

A. Relevance of the RDB-statistics-based Cardinality Estimation

We successively assess the relevance of the proposed estimation based on selectivity degrees in terms of accuracy of the estimated value, accuracy of the selected quantifier and relevance of the computed confidence degree, as detailed in turn below.

1) Accuracy of the Estimated Relative Cardinalities: For the two datasets, we first compare the cardinality estimated from the RDB statistics and the real one (computed directly from the data) for each vocabulary term and also for each possible conjunctive combination of up to six terms (the cardinality threshold κ is set to 0.02). Figure 6 clearly shows that the difference of cardinality (y-axis) is rather low, especially for atomic properties (value 1 on the x-axis). Moreover, it stays low even for conjunctive properties (other values of the x-axis), showing no significant difference with the atomic case. The worst estimation indeed only differs from the real cardinality of 0.16 for the *flights* dataset and 0.13 for the *cars* dataset, and the difference is in average 0.01 for *flights* and 0.015 for *cars*.

2) Accuracy of the Summaries Generated from Estimated Relative Cardinalities: We then observe the impact of the use

[1]Stored in a PostgreSQL server v.9.6 running on a 3,1 GHz i7 with 16 GB 1867 MHz DDR3.

[2]This set is published by Research and Innovative Technology Administration (RITA) and Bureau of Transportation Statistics (BTS) http://stat-computing.org/dataexpo/2009/the-data.html

[3]Extracted from real classified ads published on leboncoin.fr

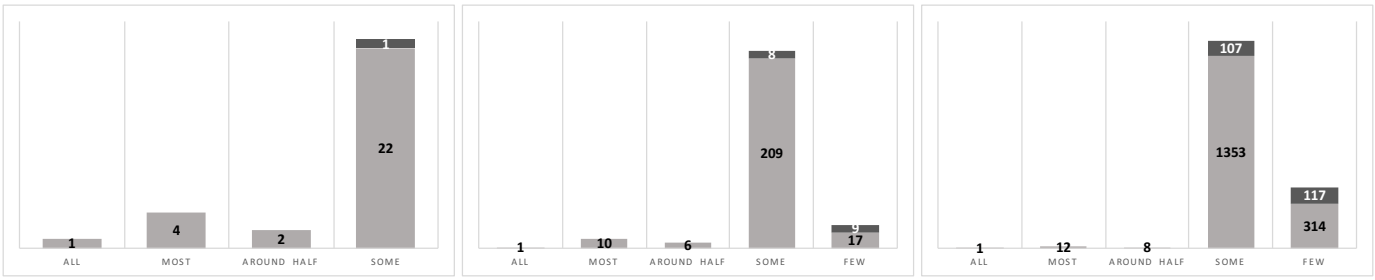

Fig. 7. Comparison of the selected quantifier, for the real and the estimated cardinality (*flights*data set): summarizers with one (left), two (center) and four conjuncts (right)

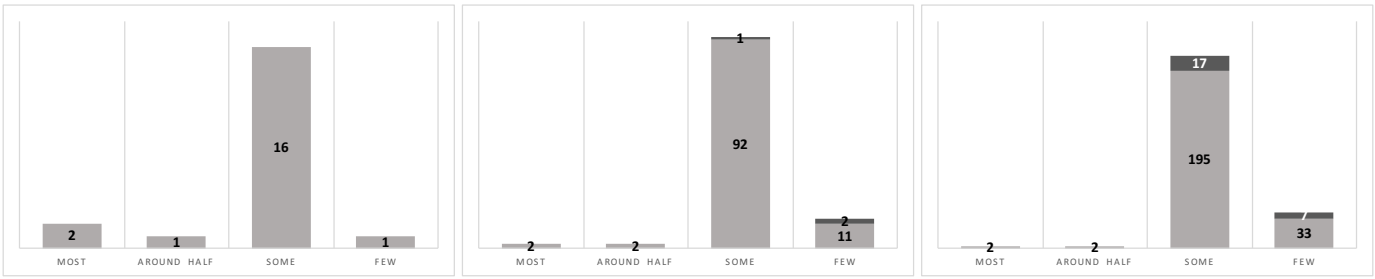

Fig. 8. Comparison of the selected quantifier, for the real and the estimated cardinality (*cars* data set): summarizers with one (left), two (center) and four conjuncts (right)

of estimated cardinalities on the target task, i.e. the linguistic summarization of data. To do so, we compare the linguistic summaries generated based on the real cardinality with the ones generated based on the estimated ones. More precisely, for a given summarizer P, we check whether the quantifiers that best describe the real cardinality of P and its estimated one are the same.

The result of this comparison for atomic summarizer as well as summarizers of two and four conjuncts are given in Figure 7 for the *flights* dataset and in Figure 8 for the *cars* dataset. The light grey part in each bar represents the number of summaries to find for the different quantifiers (i.e. summaries computed using the real cardinalities). The top dark grey part of each bar gives the number of summaries not found using RDB-statistics-based estimated cardinalities. For example, among the 22 expected atomic summaries of the form $SOME\ flights\ are/have\ P$, (left graph of Figure 7), only one is not correctly generated by our estimation-based approach. In this case and in all other error cases as well, the estimated cardinality is very close to the transition point between two adjacent quantifiers and a small difference of cardinality in such areas may lead to the choice of the wrong quantifier (e.g. *most* instead of *few* or vice-versa). But, as we will see in Section VI-A3, a confidence degree is computed to identify these cases.

The results of this study clearly show that cardinalities estimated from RDB statistics can be reliably used to identify the main trends in the data as all the summaries describing *all*, *most* or *around half* of the dataset are correctly generated by our approach. All of the errors concern inversions between *some* and *few*, i.e. minor trends; the average error rate for summaries involving up to six conjuncts is only 7% for the flights and 5% for the cars.

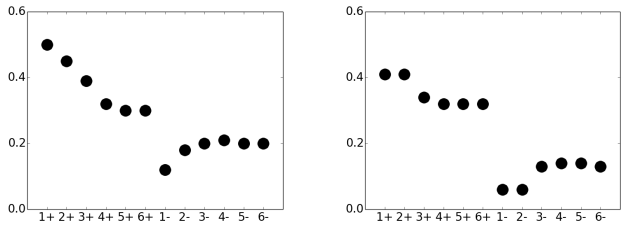

Fig. 9. Average confidence degree in the estimation, as a function of the summarizer size, for correctly (+ part) and incorrectly selected quantifiers (i part), left: *flights* data set, right: *cars* data set

3) Relevance of the Confidence Degree: We also check that the confidence degree (as defined in Section VI-A3) attached to each estimated summary makes it possible to discriminate between the correct and the incorrect estimated summaries. It is worth recalling that by *incorrect summaries* we mean that a wrong quantifier is used to describe the estimated cardinality of a summarizer and that in all of the erroneous cases the quantifier used is adjacent to the correct one.

Figure 9 indicates the average confidence degree attached to correctly estimated summaries involving up to six conjuncts (values $1+$, $2+$, $3+$, $4+$, $5+$ and $6+$ on the x-axis) and the one attached to incorrect summaries involving up to six conjuncts (values $1-$, $2-$, $3-$, $4-$, $5-$ and $6-$ on the x-axis). For the two datasets, the confidence degree attached to each summary appears to be meaningful as it clearly distinguishes between the correctly and incorrectly estimated summaries.

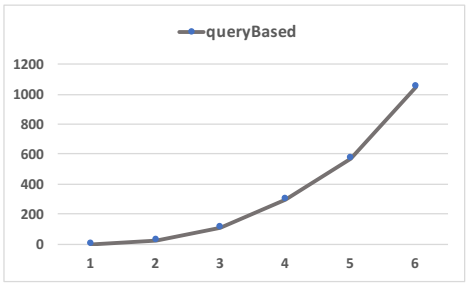

Fig. 10. Computation time of the summarization wrt. the size of the search space: Method *queryBased*

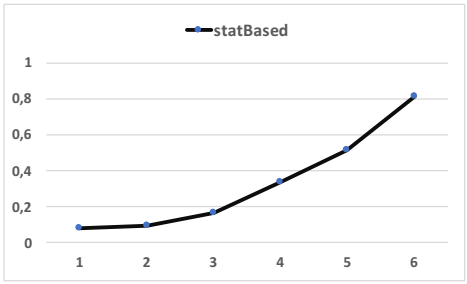

Fig. 11. Computation time of the summarization wrt. the size of the search space: Method *statBased*

B. Efficiency of the RDB-statistics-based Summarization Approach

Whereas the previous experiments study the relevance of the proposed approach, this section considers the efficiency issue, in terms of needed computational time, with respect to the search space and the data sizes: we compare the time needed to generate the linguistic summaries of a dataset using cardinalities estimated from RDB-statistics, method denoted *statBased* hereafter, and using real cardinalities computed by querying the DB, method denoted *queryBased*. This last approach relying on DB queries integrates the pruning criteria of the search space (Figure 3) detailed in [5] (pruning of conjunctions based on the monotonicity of the cardinality computation, implied quantifiers and summarizers).

1) Processing Time wrt. the Size of the Search Space: To observe the impact of the size of the search space (i.e. the lattice illustrated in Figure 3), we first compare the time needed (y-axis in second) by the two approaches, *statBased* vs. *queryBased*, to summarize the *cars* dataset wrt. the maximal number of conjuncts considered in the summarizers, from 1 to 6 conjuncts (x-axis).

Figure 10 and 11 clearly show that the time needed to summarize a relation (storing the *cars* dataset composed of 98,562 tuples in this case) grows exponentially wrt. the size of the search space, for both approaches.

2) Processing Time wrt. the Size of the Relation to Summarize: Another efficiency comparison criterion considers the processing time as a function of the data base size, it leads to a clear and indisputable conclusion. Indeed, an approach based on data access, as the *queryBased* one, is not scalable and cannot be envisaged to summarize large datasets such as the total *flight* ones. However, as confirmed by Figure 12,

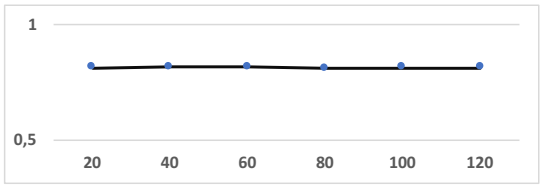

Fig. 12. Computation time of the *statBased* approach, for datasets of various sizes, defined as subparts of the *flights* dataset

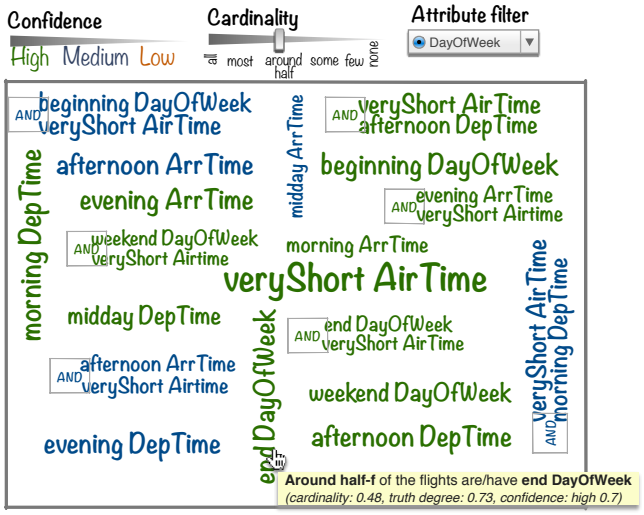

Fig. 13. Graphical rendering of the extracted linguistic summaries with their confidence level.

the proposed strategy exploiting the statistics about the data distribution available in any RDBMS without any additional computation, does not depend on the size of the dataset. Figure 12 indeed shows that the time needed (y-axis in second) by the *statBased* approach to generate all the linguistic summaries (still using $\kappa = 0.02$) is constant whatever the size of the *flights* dataset (x-axis in million of tuples).

As a final comparison, it takes 4597 seconds to linguistically summarize a table of 7,000,000 flights using the *queryBased* approach (with indexes defined on each attribute to speed up the queries execution); it only takes 0.814 second for the *statBased* approach to generate the 333 linguistic statements with only 12 erroneous quantifiers (9 uses of *some* instead of *few*, and 3 uses of *few* instead of *some*).

VII. Visualization of the Extracted Linguistic Summaries

Soft-computing-based strategies to data summarization generally generate a list of linguistic statements that are given to the user, in a decreasing order of their cardinality and truth degree. As mentioned in Section II, one of the challenges is the huge amount of produced results. We propose to address this issue using a graphical representation, that is in line with the user demands: domain experts faced with data analysis tasks are nowadays mainly looking for graphical views of the data and of the extracted knowledge [19], [20].

Instead of providing the user with an ordered list of linguistic phrases of the form Q R *are/have* P, we propose to only display the summarizers as a tag cloud. Figure 13 illustrates the proposed graphical rendering: the position and orientation is randomly set, the size of the font is proportional to the cardinality of the data subset it covers and the font color depends on the confidence in the estimated cardinality (Section VI-A3). The cloud simultaneously contains atomic and conjunctive summarizers.

On the top of the view, an interactive exploration tool offers the user the possibility to filter the set of summarizers according to their confidence, their cardinality (in Figure 13 only the properties describing at least around half of the data are shown), or/and the attribute they describe.

VIII. Conclusion and Future Works

In order to extract linguistic summaries, we propose a new approach in the prospect of handling large amounts of data. Instead of scanning or querying the data to identify relevant terms and conjunctions of terms from the user vocabulary that describe the data we propose to make the most of statistics about data distribution that are maintained by any RDBMS, to estimate the cardinality of the candidate summarizers, making it possible to identify the ones that best describe the data. In addition to this novel strategy of data summarization, we define a measure to quantify a confidence degree for each estimated cardinality. Experiments conducted on two different real data sets show the relevance of the estimated cardinalities and the efficiency of the summarization process that does not depend any more on the size of the dataset: it achieves a 10,000 speed up factor as compared to a query-based approach and it takes the same amount of time to summarize a dataset of one thousand tuples and a dataset containing millions of tuples. We also propose to graphically render and explore the linguistic statements that form the data summary.

An interesting aspect of this work comes from the intersection created between the soft computing, data mining and DB domains, and we plan to continue making the most of the complementarity of these fields for other data mining tasks, such as the related association rules discovery for instance. Another direction for future works concerns the summarization of tuples using other types of protoforms, in particular Q *of the* P R*'s are/have also* P', which may require to precompute additional statistics and imply the proposal of dedicated strategies.

Acknowledgments

This work has been partially funded by the French DGE (Direction Générale des Entreprises) under the project ODIN (Open Data INtelligence).

References

[1] R. J. Almeida, M.-J. Lesot, B. Bouchon-Meunier, U. Kaymak, and G. Moyse. Linguistic Summaries of Categorical Time Series Patient Data. In *Proc. of FUZZ-IEEE*, 2013.

[2] F. E. Boran, D. Akay, and R. R. Yager. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, 61:356–377, 2016.

[3] P. Cariñena, A. Bugarín, M. Mucientes, and S. Barro Ameneiro. A language for expressing fuzzy temporal rules. *Mathware & soft computing*, 7(2):213–227, 2000.

[4] R. Castillo-Ortega, N. Marin, and D. Sanchez. Linguistic local change comparison of time series. In *Proc. of FUZZ-IEEE*, pages 2909–2915, 2011.

[5] R. Dijkman and A. Wilbik. Linguistic summarization of event logs–a practical approach. *Information Systems*, 67:114–125, 2017.

[6] R. George and R. Srikanth. Data summarization using genetic algorithms and fuzzy logic. In F. Herrera and J.-L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 599–611. Physica-Verlag, 1996.

[7] M. Gyssens and L. V. Lakshmanan. A foundation for multi-dimensional databases. In *Proc of VLDB*, volume 97, pages 106–115, 1997.

[8] J. Kacprzyk, A. Wilbik, and S. Zadrożny. Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets and Systems*, 159(12):1485–1499, 2008.

[9] J. Kacprzyk and R. Yager. "Softer" optimization and control models via fuzzy linguistic quantifiers. *Information Sciences*, 34(2):157–178, 1984.

[10] J. Kacprzyk and R. Yager. Linguistic summaries of data using fuzzy logic. *Int. Journal of General Systems*, 30(2):133–154, 2001.

[11] J. Kacprzyk and S. Zadrożny. On combining intelligent querying and data mining using fuzzy logic concepts. In G. Bordogna and G. Pasi, editors, *Recent Research Issues on the Management of Fuzziness in Databases*, pages 67–81. Physica-Verlag, 2000.

[12] J. Kacprzyk and S. Zadrożny. Protoforms of linguistic data summaries: Towards more general natural-language-based data mining tools. *Soft Computing Systems*, pages 417–425, 2002.

[13] J. Kacprzyk and S. Zadrożny. Linguistic summarization of data sets using association rules. In *Proc. of the 12th IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE'03*, pages 702–707. IEEE, 2003.

[14] J. Kacprzyk and S. Zadrożny. Derivation of linguistic summaries is inherently difficult: Can association rule mining help? In C. Borgelt, M. A. Gil, J. M. Sousa, and M. Verleysen, editors, *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, pages 291–303. Springer, 2013.

[15] G. Moyse, M.-J. Lesot, and B. Bouchon-Meunier. Linguistic summaries for periodicity detection based on mathematical morphology. In *Proc. of IEEE SSCI*, pages 106–113, 2013.

[16] D. Pilarski. Linguistic summarization of databases with quantirius: a reduction algorithm for generated summaries. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(3):305–331, 2010.

[17] D. Rasmussen and R. Yager. Finding fuzzy and gradual functional dependencies with summarySQL. *Fuzzy Sets and Systems*, 106:131–142, 1999.

[18] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22 – 32, 1969.

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

[20] G. Smits, R. R. Yager, and O. Pivert. Interactive data exploration on top of linguistic summaries. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, pages 1–8. IEEE, 2017.

[21] A. Wilbik and R. M. Dijkman. On the generation of useful linguistic summaries of sequences. In *Proc. of the IEEE Int. World Conf. on Computational Intelligence, WCCI*, pages 555–562. IEEE, 2016.

[22] A. Wilbik and J. Kacprzyk. Towards an efficient generation of linguistic summaries of time series using a degree of focus. In *Proc. of the 28th North American Fuzzy Information Processing Society Annual Conf., NAFIPS'09*, 2009.

[23] A. Wilbik, U. Kaymak, and R. Dijkman. A method for improving the generation of linguistic summaries. In *Proc. of the Int. Conf. on Fuzzy Systems, FUZZ-IEEE'17*. IEEE, 2017.

[24] R. Yager. A new approach to the summarization of data. *Information Sciences*, 28:69–86, 1982.

[25] X. Yang, C. M. Procopiuc, and D. Srivastava. Summarizing relational databases. *Proceedings of the VLDB Endowment*, 2(1):634–645, 2009.

[26] C. Yu and H. Jagadish. Schema summarization. In *Proc. of the 32nd Int. Conf. on Very Large Data Bases*, pages 319–330. VLDB Endowment, 2006.

[27] L. A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2):111–127, 1997.