Combining Quantifications for Flexible Query Result Ranking

Christophe Billiet, Guy De Tré
Department of Telecommunications and Information Processing, Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
Email: Christophe.Billiet@UGent.be, Guy.DeTre@UGent.be

***Abstract*—Databases contain data and database systems governing such databases are often intended to allow a user to query these data. On one hand, these data may be subject to imperfections, on the other hand, users may employ imperfect query preference specifications to query such databases. All of these imperfections lead to each query answer being accompanied by a collection of quantifications indicating how well (part of) a group of data complies with (part of) the user's query. A fundamental question is how to present the user with the query answers complying best to his or her query preferences. The work presented in this paper first determines the difficulties to overcome in reaching such presentation. Mainly, a useful presentation needs the ranking of the query answers based on the aforementioned quantifications, but it seems advisable to not combine quantifications with different interpretations. Thus, the work presented in this paper continues to introduce and examine a novel technique to determine a query answer ranking. Finally, a few aspects of this technique, among which its computational efficiency, are discussed.**

I. Introduction and Opening Example

Generally, databases contain data. Usually, such data are the results of measurements, descriptions or calculations intended to capture those properties of real-world objects or concepts, that are deemed necessary to be preserved by humans. In this way, database systems managing databases model real-world objects or concepts and thus (parts of) reality. For example, a database containing data representing rental cars might contain the numerical datum '2010' to represent in what year a car was made.

Obviously, one of the most important purposes of database systems is to allow humans (and other systems) to query data or retrieve information or knowledge from the databases they control. For example, the database system controlling the database mentioned above might allow users to query its data, from which can be learned in which years the 10 most often rented cars were made.

In the last decades, both the way in which properties of real-world objects or concepts may be represented in databases and the way in which data may be queried or information or knowledge may be retrieved from databases have been severely enhanced by proposals concerning applications of soft computing techniques.

Many of the data in databases are (directly, or through human-made equipment) produced by humans or represent information or knowledge (directly or indirectly) produced by humans. In the last decades, researchers have started to acknowledge the observation that such human-made data, information or knowledge can be subject to imperfections, often due to the imperfect nature of humans and human reasoning or to imperfections in measuring equipment. Such imperfections may take the form of uncertainties [1]–[9], imprecisions [1], [8], [9], vaguenesses [1], [9], contradictions, etc [8]–[10]. To allow the representation of such imperfections or ways to deal with them, many existing approaches propose to extend the data contained in databases to also contain descriptions of the determined intensity levels of such imperfections or such ways of handling imperfections [1], [5], [6], [11].

For example, consider table I, which is the visualization of an example relational database relation containing data representing 3 properties of each of 5 rental cars. For every such car, data representing a unique ID number, the color of the car and an ill-known time interval [1]–[3], [12]–[15] determining when the car might be available for rent, are stored. The first row in the table contains these properties' (or in the context of databases: attributes') names, every other row visualizes the values representing these properties for one car. Every value for the attribute 'Availability' represents an ill-known time interval, which is a time interval subject to uncertainty caused by a (partial) lack of knowledge, about which all the available knowledge is contained in a possibility distribution on the set of all existing time intervals. The interpretation is: an exact time interval during which a car is available, is intended, but due to a (partial) lack of knowledge about the circumstances determining this interval, it is uncertain exactly which interval is intended. In order to model confidence about which interval is intended, every existing time interval is given a possibility degree, which is interpreted as the degree of how plausible it is, given all available knowledge, that the corresponding time interval is the intended interval [1]–[3], [12]–[15].

While querying a database, a user may convey his or her query preferences in different ways. The historically oldest way is the *regular* way, following which the user determines the data which he or she finds desired or satisfactory and thus wants to retrieve, by *perfectly* describing the allowed values of these data. For example, a user might be interested in all rental cars build 'in the exact year 2010'.

In the last decades, several proposals were made, which present approaches to allow users to use a *fuzzy* way to convey preferences, following which the user determines the data which he or she finds desired or satisfactory and thus wants to retrieve, by *imperfectly* describing the allowed values of these data [4], [10], [16]–[19]. For example, a user might be interested in all rental cars build 'around the start of this century'.

In recent years, several proposals were made, which present approaches to allow users to use a *bipolar* way to convey

TABLE I. AN EXAMPLE RELATION.

ID	Color	Availability
001	red	IKI_1
002	teal	IKI_2
003	blue	IKI_3
004	light green	IKI_4
005	green	IKI_5

TABLE II. THE RESULT OF THE EXAMPLE QUERY.

ID	Color (Dis)satisfaction	Availability
001	0.0	1.0
002	0.5	0.5
003	0.1	0.1
004	0.8	0.9
005	1.0	0.0

preferences. Two main types of such approaches exist [20]. Approaches of one type allow the user to determine the data which he or she finds acceptable and among this acceptable data, to determine the data which he or she finds really desired, both by describing the allowed values of these data. Approaches of the other type allow the user to independently determine both the data which he or she finds desired or satisfactory and the data which he or she finds undesired or unsatisfactory, both by describing the allowed values of these data. The descriptions used in bipolar querying may contain imperfections [3], [4], [11], [16], [18]–[23].

For example, a user might 'prefer a green car, but dislike red cars' and might 'desire the car to be available during a certain week'.

Usually, when querying a database not containing data subject to imperfections (neither data representing information or knowledge subject to imperfections) using a regular approach towards conveying preferences, the query result is a set of data collections, each containing coherent data which fully and perfectly comply with the user's query preferences. Thus, a similar data collection is considered an answer to the query.

However, if a database contains data subject to imperfections or it is queried using a fuzzy way, the query result will usually be a set of data collections, each containing coherent data which comply with the user's query preferences to any degree. Each of these data collections is then accompanied by a corresponding set of gradual, usually numerical, indications, where each indication is a quantification of how well (part of) the accompanying data collection complies with (part of) the user's query preferences. In this paper, a *quantification* is exactly this: a numerical value expressing a valuation. Thus, a similar data collection is again considered a query answer, but, compliance to the user's preferences is now a matter of degrees (quantifications) [2]–[7], [10]–[13], [15]–[17], [19].

For example, the result set of the last example query applied to the example relation visualized in table I might take the form of the set of tuples visualized in table II. In this table, the first row contains labels of which every label in a column refers to the meaning of the values visualized in that column and every other row corresponds to a different tuple of the example relation. The values visualized in the column labeled 'ID' hereby refer to the ID's of the example cars, as visualised in table I. The value visualized in the column with label Color (Dis)satisfaction for a tuple is a quantification of how well the car with this tuple's ID complies with the user's preferences about color, based on the example relation's data. The value is a number between 0 and 1, a bigger number indicating higher satisfaction. The value visualized in the column with label Availability for a tuple is a quantification of how plausible it is, given all available knowledge, that the car with this tuple's ID is available during the entire preferred week. Hence, this value is a possibility degree between 0 and 1.

It is clear that, although every quantification accompanying a query answer quantifies a level of compliance to the user's preferences of the same group of data, different quantifications may have distinctly different interpretations and semantics. A fundamental question arises now: should one consider combining such quantifications with different interpretations? For example, should the values for a tuple visualized in table II in the columns with labels Color (Dis)satisfaction and Availability be combined?

On one hand, such quantifications have distinctly different semantics and it would not be clear what exactly the meaning or interpretation would be of the result of such a combination or what the semantically most coherent ways would be to further process such combination results. For example, would the combination mentioned above result in a quantification of satisfaction, of possibility or even of something else? Would the processing of such combination result employ operators of possibility theory or others? Thus, it is important, for every query answer presented to the user, to certainly preserve all different types of quantifications as separate (meta)data.

On the other hand, without an unambiguous and straightforward ranking of the query result tuples, a user cannot clearly or easily discern the query answers which comply well with his or her demands from those which don't. This would defeat the purpose of querying. For example, it is somehow clear that the result tuple with ID '004' visualized in table II should be offered more prominently to the user than the query answer with ID '003'. Reasonably, such a ranking should be based on how well the result tuples comply with the user's different preferences. As there cannot exist a ranking between quantifications with different semantics, a combination of quantifications with different semantics seems to be required.

In an attempt to bring a solution to this problem more within reach, in this paper, a general technique is presented to create an unambiguous and straightforward ranking of query result tuples, based on how well these query answers comply with the user's preferences, without actually combining the corresponding quantifications. Succinctly, the idea behind this approach is to first consider the domain of each quantification with a distinct interpretation as an orthogonal dimension in a Cartesian space made up of these dimensions. Next, the vector of all quantifications corresponding to a query answer is considered a point in this space. Next, a reference point is chosen. Finally, a distance measure is chosen and for each point corresponding to a result tuple, the distance between this point and the reference point is measured. The tuples are then ranked according to these distances. Next to the presentation of this technique, this paper contains two smaller contributions in the form of a determination of acceptable guidelines for the creation of a similar ranking of query answers (and a motivation for these guidelines) on one hand and a concise reasoning about the choices of the aforementioned reference point and distance measure on the other.

To the knowledge of the authors, only a few papers have acknowledged the existence of the aforementioned problem [4], but no paper has ever focussed on attempting to solve it. In the opinion of the authors, this is quite astonishing, as they assess the impact of this problem for soft computing in information retrieval to be quite big. Indeed, it is the authors' opinion that for soft computing in information retrieval to reach one of its goals, namely to allow people to retrieve possibly imperfect data, information or knowledge in a possibly imperfect way, it is essential that both the representation and handling of imperfection in data in databases, (or in information or knowledge represented in databases) and the representation and handling of imperfection in ways of retrieving data, information or knowledge from such databases, is supported. Moreover, the presence of such support should not impede the usefulness and execution speed of such database systems. It is the authors' opinion that the work presented in this paper is an important step in allowing this support.

The rest of this paper is structured as follows. In section III, the problem of creating a result ranking under the aforementioned constraints is presented and briefly examined, resulting in a set of acceptable guidelines for the creation of such a ranking. In section IV, an outline for the general technique which is the main contribution of this work, is presented and briefly motivated. In section V, a concise reasoning is presented about some choices which must be made in the context of the presented general technique. A few interesting choices are presented and discussed. In section VI, a brief examination of the efficiency of the presented technique is described. Finally, in section VII, some conclusions and future work in the light of this research are discussed. First however, the next section (section II) presents and describes some of the most prevalent types of imperfections encountered in information retrieval.

II. Preliminaries: Types of Imperfections

Data, information or knowledge may be subject to different types of imperfections. In the following sections, the most prevalent types of imperfections and the ways in which such imperfections or knowledge about such imperfections is usually represented or modeled, are shortly described.

A. Uncertainty

In some cases, it is known that a datum is intended, but this datum is somehow unknown. Often, a set of data exists, each of which could be the intended datum. In those cases, it is said that the datum is subject to uncertainty [1], [2], [5], [8], [12], [21], [24]–[27]. Usually, for a datum subject to uncertainty, an attempt is made to model the available knowledge about the intended datum, by assigning each of the data which could be the intended datum a degree of confidence an agent has that the corresponding datum is the intended datum. Thus, existing knowledge about the intended datum often takes the form of a distribution of such degrees of confidence on existing data. Depending on the source of the circumstances of the uncertainty, such confidence can take different forms, where different forms have different interpretations and different handling rules. When the source is variability, confidence usually takes the form of chance and probability theory is used [1], [12], [26], [27]. When the source is a (partial) lack of knowledge, confidence usually takes the form of possibility and possibility theory is used [1], [8], [12], [21], [24], [25]. Other sources and forms of confidence exist.
For example, it might be known that a certain instantaneous event took place during an hour of a given day, but not during which hour. In this case, one could attach a degree of confidence to each hour of the given day expressing one's confidence that the event took place in that hour.

B. Imprecision and Vagueness

In existing literature, imperfections of different types have been named imprecisions, hence different views on the interpretation of imprecisions exist.
Some authors consider imprecision to be the imperfection to which a datum is subject if this datum is described with a precision which is coarser than the precision needed [8], [9], [21], [24], [28]. For example, if the height of a person is required as an amount of centimeters and is described to be between 1.8 and 1.9 meters, the number representing this height is said to be subject to imprecision. According to this view on the interpretation of imprecision, confidence about the intended datum may be modeled just like confidence about a datum subject to uncertainty is modeled [8], [9].
Other authors consider imprecision to be the imperfection to which a datum is subject if crisp boundaries for this datum do not exist, often because it is bounded in a gradual way [1], [21], [28], [29]. For example, the time interval indicated by the words 'the Industrial Revolution' has no crisp boundaries, as it gradually came into existence and gradually faded out. Usually, a datum subject to imprecision which is interpreted following this view, is represented by a fuzzy set with a conjunctive interpretation, where the gradualness of the datum is reflected in the gradualness of the fuzzy set.
Both views on the interpretation of imprecisions approach vagueness in the same way: vagueness is considered to be the same imperfection as imprecision, except for the fact that the description of a datum subject to vagueness is always linguistic.

In the next section, the main issues arising when attempting to create a presentation of query answers in the presence of such imperfections are presented, described and discussed.

III. Combining Quantifications and Requirements for Ranking

As mentioned in section I, if a database contains data subject to imperfections or it is queried using a fuzzy way, the query result will usually be a set of data collections, each containing coherent data which comply with the user's query preferences to some degree(s), each accompanied by one or more gradual indication(s), where each gradual indication is a quantification of how well (part of) the accompanying data collection complies with (part of) the user's query preferences. Obviously, if the data in the database is subject to several different types of imperfections, if several different types of imperfections are allowed to be used in querying the database or if a type of imperfection to which the data in the database is subject differs from a type of imperfection allowed to be used in querying that database, each of the aforementioned query answers will be accompanied by not one quantification, but a corresponding collection of several different quantifications where each quantification indicates either how well that part of the corresponding data collection which is subject to a certain imperfection complies with the corresponding perfect part of the user's query preferences, or how well that perfect part of the corresponding group of data complies with the corresponding part of the user's query preferences expressed in a fuzzy way, or how well that imperfect part of the corresponding group of data complies with the corresponding part of the user's query preferences expressed in a fuzzy way, where both imperfections in data and query have the same type. For example, if a relational database relation containing data subject to uncertainty is queried using query condition specifications using vague terms, each result tuple will be given a quantification indicating the confidence about the data in this tuple in the context of this uncertainty and another quantification indicating how well data in this tuple fit the vaguely specified query conditions. A similar thing can be observed in table III, which is a visualization of the set of tuples resulting from the querying of the example relation visualized in table I using a query expressing in a fuzzy way that: *the user searches a car to rent and prefers a green car, but dislikes red cars and desires the car under consideration to be available during a given, precisely specified week.* As many different shades of green and red exist, the user's query condition with respect to the car color is subject to vagueness. Thus, every result tuple is given a quantification indicating how well the color representation in the tuple represents the user's color-related query condition. These quantifications are visualized in the column labeled 'Metadata: Color Satisfaction Degree' in table III, for every result tuple visualized by a row in this table.

The uncertainty about the availability of the cars during the week indicated in the user's query preferences, gives rise to a possibility degree indicating how possible it is that a car is available during the indicated week. In table III, this possibility degree for a car corresponding to a result tuple is visualized in the row corresponding to that tuple, in the column labeled 'Metadata: Possibility of Availability'.

The main problem examined in the work presented in this paper is how to present the query answers in a query result, of which each is accompanied by a collection of quantifications with different interpretations, resulting from a user query, to said user. It is the opinion of the authors that in doing so, two main concerns must be taken into account.

On one hand, it must be taken into account that such quantifications with different interpretations have clearly different semantics and as a result of this, combining quantifications with one interpretation with one another may require the employment of different rules than the ones employed to combine quantifications with another, different interpretation with one another. As a result of this observation, it seems logical to discourage the combination of two or more different quantifications with different interpretations, assigned to the same query answer. Indeed, it is the opinion of the authors that it cannot always be clear what the semantics of such a combination would be or what rules would be employed to further compare or combine the result of such a combination with the results of other, similar combinations. For example, if one would try to combine a quantification of chance with a quantification of possibility, it would not be clear if the resulting combination would be a quantification of chance or a quantification of possibility or even something else. Based on this argument, the authors would like to conclude that a straightforward combination of two or more different quantifications with different interpretations should be avoided, and that it can be argued that, for each query answer, preserving the distinct quantifications with different interpretations associated to it (for example as metadata corresponding to the query result set) might be very useful.

On the other hand, it must be taken into account that the amounts of data contained in existing databases and the amounts of real-life concepts or objects represented by these data are usually great and even strongly increasing. As mentioned before, if a database contains data subject to imperfections or it is queried using a fuzzy way, the query result will usually be a set of data collections, each containing coherent data which comply with the user's query preferences to some degree(s). Thus, this result set, of which elements may be presented to the user, usually contains a great many elements. However, it is reasonable to assume that among the answers in such a result set, the user is most or even only interested in the ones that comply best with this user's query preferences. Hence, previous proposals suggested to rank the query answers in the result set according to how well they comply with the user's query preferences and either only or most prevalently present the user with the top-k query answers (where k is a positive integer) based on this ranking. Following this reasoning, the authors would like to conclude that it can be argued that ranking the answers to a user's query before presenting them to this user is fundamental to the usability of the retrieval of data, information or knowledge.

In situations where each of the answers in a result set originating from a user's query is accompanied by a single quantification of how well that answer complies to this query, where all of these quantifications have the same interpretation, many existing approaches suggest to base the determination of the rank of each of these answers on the size of its corresponding quantification. Although a similar ranking expresses some kind of comparison between answers, because the interpretation and semantics of the aforementioned quantifications are the same for all answers, these quantifications may be used as a basis for this comparison. Moreover, as such a quantification corresponding to an answer is a quantification of how well the answer's data complies with the user's query preferences, it is logical to assume that a similar ranking reflects the way in which the user would evaluate the answers, based on his or her query preferences.

Consider the situation described in the beginning of this section, where each of the answers in a result set originating from a user's query is accompanied by a collection of different quantifications with different interpretations. As argued above, it is necessary to rank these answers before presenting (some of) them to the user and it seems logical to determine a similar ranking based on the quantifications corresponding to the answers, where the rank of an answer reflects how well the answer complies with the user's query preferences, by being based on all of the collection of quantifications corresponding to the answer. However, this is not a straightforward task.

Taking all of the arguments presented above into account, it is clear that the answers to a user's query should be ranked before presentation to the user, and it is the proposal of the authors that this ranking should have the following properties,

TABLE III. THE RESULT SET FOR THE EXAMPLE QUERY APPLIED TO THE EXAMPLE RELATION.

ID	Color	Availability	Metadata: Color Satisfaction Degree	Metadata: Possibility of Availability
001	red	IKI_1	0.0	1.0
002	teal	IKI_2	0.5	0.5
003	blue	IKI_3	0.1	0.1
004	light green	IKI_4	0.8	0.9
005	green	IKI_5	1.0	0.0

based on the arguments given above.

1) The determination of the rank of a query answer must take into account all quantifications corresponding to this answer.
2) The determination of the rank of a query answer must not contain a combination of quantifications with different interpretations.
3) The way of ranking the answers to a user's query must reflect the assumed intention of the user, i.e. the highest ranks must be assigned to the query answers best complying with the biggest part of the user's query preferences.
4) The determination of the ranks of the query answers must not significantly slow down the process of selecting which answers will be presented to the user.

In the next section, the general technique which is the main contribution of the work presented in this paper, is described. This technique is intended to rank query answers according to the properties enumerated above.

IV. THE PROPOSED RANKING APPROACH

Consider a database and a user's query applied to this database. Now consider the result set

$$R = \{(R_1, Q_1), (R_2, Q_2), \cdots, (R_i, Q_i), \cdots, (R_n, Q_n)\}$$

of this query, where every $(R_i, Q_i), i \in \mathbb{N} \wedge 1 \leq i \leq n$ consists of an answer R_i to the user's query and an ordered collection of quantifications Q_i corresponding to R_i. Let each such ordered collection of quantifications Q_i consist of m quantifications, i.e. $Q_i = (q_{i,1}, q_{i,2}, \cdots, q_{i,j}, \cdots, q_{i,m})$, and let the quantification $q_{i,j}$ with the same index j have the same interpretation, for every collection of quantifications Q_i, and thus correspond to the same type of quantification T_j.

The work presented in this paper now proposes the following steps to determine a rank for every query answer $R_i, i \in \mathbb{N} \wedge 1 \leq i \leq n$.
Firstly, a vector space V is constructed, with m dimensions, where every dimension $D_j, j \in \mathbb{N} \wedge 1 \leq j \leq m$ corresponds to a type of quantifications T_j, which is the type to which the quantifications $q_{i,j}, i \in \mathbb{N} \wedge 1 \leq i \leq n$ belong. This vector space obviously has an origin, which is the vector $O = (o_1, o_2, \cdots, o_j, \cdots, o_m)$, where, for every $j \in \mathbb{N} \wedge 1 \leq j \leq m$, the value o_j is the smallest allowed value for the type of quantification T_j. In this vector space V, a subspace V_{sub} is considered, where V_{sub} contains every vector $v = (v_1, v_2, \cdots, v_j, \cdots, v_m)$ of which the value v_j is an allowed value for quantification type T_j, and no other vectors. Hence, through this construction, every such vector v_j corresponds to a possible ordered collection of quantifications.
Secondly, a reference vector $V_{ref} = (v_{ref,1}, v_{ref,2}, \cdots, v_{ref,j}, \cdots, v_{ref,m})$ in this vector space V is chosen. The intention is that this reference vector is the vector to which all vectors corresponding to query answers are compared in order to determine their ranks.
Thirdly, for every vector $VR_i = (vr_{i,1}, vr_{i,2}, \cdots, vr_{i,j}, \cdots, vr_{i,m})$ corresponding to a collection of quantifications Q_i, $i \in \mathbb{N} \wedge 1 \leq i \leq n$, the Euclidean distance d_i between the chosen reference vector and vector VR_i is calculated, i.e.:

$$\begin{aligned} d_i \quad = \quad & ((v_{ref,1} - vr_{i,1})^2 + (v_{ref,2} - vr_{i,2})^2 + \cdots \\ & + (v_{ref,j} - vr_{i,j})^2 + \cdots + (v_{ref,m} - vr_{i,m})^2)^{\frac{1}{2}} \end{aligned}$$

Fourthly, for each result $(R_i, Q_i) \in R, i \in \mathbb{N} \wedge 1 \leq i \leq n$, depending on the reference vector chosen, this Euclidean distance d_i determined for vector VR_i corresponding to Q_i is used as a basis to determine the rank of the corresponding query answer R_i.

With respect to this technique, three remarks should be made.

1) This technique makes the assumption that every type of quantification used to indicate how well (a part of) a query answer complies with (a part of) the user's query, is equipped with a total ordering. This assumption makes sense: indeed, if a type of quantification would not be equipped with a total ordering, it would not be possible to rank query results based on quantifications of this type, even if the only imperfection in both the database and the query would be of this type. Hence, it would not be possible to rank these query results based on how well the answers comply with the user's query preferences, which would defeat the purpose of ranking whatsoever. Therefore, it is the opinion of the authors that such a type of quantification would not make sense, and is therefore not used.
2) Essential to the presented technique, is the fact that it does not combine quantifications with different interpretations. Indeed, in the determination of the distance between a vector VR_i and reference vector V_{ref}, the calculation develops in the following way. First, for every value $vr_{i,j}$ corresponding to a quantification $q_{i,j}$, the square $(v_{ref,j} - vr_{i,j})^2$ of the distance between both values is calculated. Here, as both $v_{ref,j}$ and $vr_{i,j}$ correspond to quantifications of the same interpretation, this squared distance calculation is a type of combination of two quantifications, but they have the same interpretation. This squared distance itself, however, does not have this interpretation, but it acquires the interpretation of 'a squared distance between two quantifications'. For this squared distance, the interpretation of the quantifications is no longer important. Next, the sum of these squared distances is calculated, followed

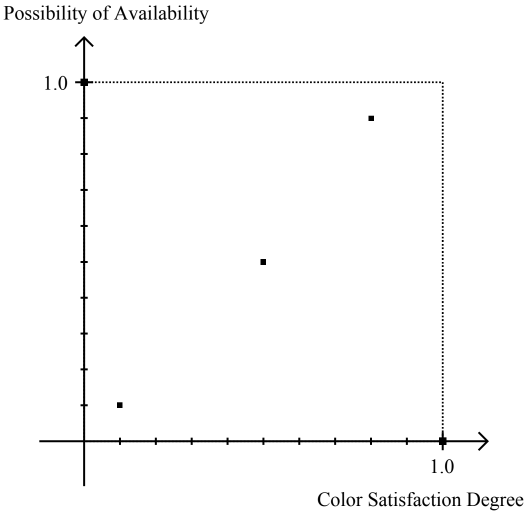

Fig. 1. A vector space corresponding to the example result set.

by the square root of this sum. Thus, this entire calculation is not based upon the combination or comparison of quantifications with different interpretations, but rather on the combination or comparison of the quantifications Q_i of the query result (R_i, Q_i) with the values of a reference vector V_{ref}, where every quantification is compared to or combined with the corresponding value of the reference vector for the dimension corresponding to the same type of quantification.

3) In theory, distance metrics differing from the Euclidean distance could be used to calculate the distance between the reference vector and a vector representing the collection of quantifications corresponding to a query answer. In the work presented in this paper, the Euclidean distance is chosen because it best represents the way in which human beings would assess a direct distance between two vectors in a vector space. Through what is deemed by the authors as a choice of distance metric representing 'natural' human reasoning, the authors hope to construct a more 'natural' ranking.

Figure 1 contains a visualization of a possible vector space V chosen in the context of the query result illustrated in table III. The chosen dimensions are named 'Color Satisfaction Degree', referring and corresponding to the satisfaction degrees quantifying compliance with respect to car color, and 'Possibility of Availability', referring and corresponding to the possibility degrees quantifying compliance with respect to car availability. A Cartesian coordinate system is installed in this space and the chosen subspace V_{sub} is illustrated using a dotted bounding box. V_{sub} is chosen based on the assumption that both given types of quantifications have the same weight and govern quantifications in the unit interval $[0, 1]$. The vectors corresponding to the collections of quantifications assigned to the query answers visualized in table III are visualized as points in figure 1.

In the next section, those choices for the reference vector, as part of the technique introduced above, which are deemed interesting by the authors, are presented, discussed and illustrated. A small discussion is added about their differences.

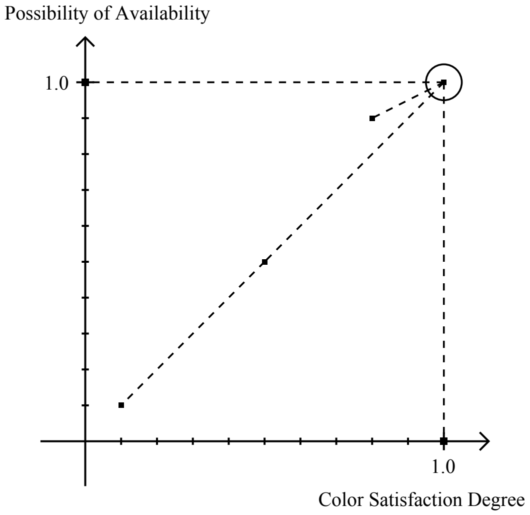

Fig. 2. Distances from the ideal vector. The visualization of the ideal vector is encircled.

V. Points of Reference

As mentioned in section IV, the second step in the proposed technique consists of choosing a reference vector $V_{ref} = (v_{ref,1}, v_{ref,2}, \cdots, v_{ref,j}, \cdots, v_{ref,m})$ in the constructed vector space V with m dimensions. In the following subsections, two choices for this reference vector, which are deemed interesting by the authors, are presented, discussed and illustrated.

A. The Ideal Vector

It would be possible to choose the so-called 'ideal vector' as a reference vector. This ideal vector is a vector $V_{ideal} = (v_{ideal,1}, v_{ideal,2}, \cdots, v_{ideal,j}, \cdots, v_{ideal,m})$ in the constructed vector space, where each value $v_{ideal,j}$ is the value of the highest quantification allowed by the quantification type corresponding to dimension D_j, for every $j \in \mathbb{N} \wedge 1 \leq j \leq m$. Hence, this ideal vector corresponds to a collection of quantifications where each quantification expresses perfect compliance of the collection's corresponding answer to the user's query preferences. The intention behind the choice of the ideal vector as reference vector is to force a comparison of every answer's collection of quantifications with the best possible collection of quantifications, where answers that comply better with the user's query preferences correspond to a vector that lies closer to the ideal vector. Thus, if the ideal vector is chosen as reference vector, the rank of a query answer should be inversely proportional to the distance between the vector corresponding to its collection of quantifications and the ideal vector. For the vectors corresponding to the answers visualized in table III, these distances are visualized by dashed lines in figure 2.

B. The Worst Vector

It would be possible to choose the so-called 'worst vector' as a reference vector. This worst vector is a vector

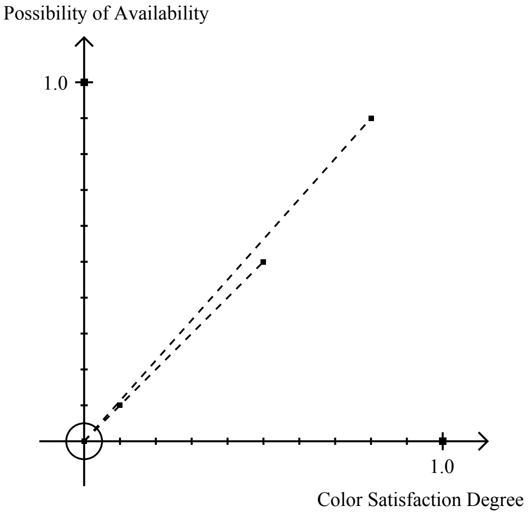

Fig. 3. Distances from the worst vector. The visualization of the worst vector is encircled.

$V_{worst} = (v_{worst,1}, v_{worst,2}, \cdots, v_{worst,j}, \cdots, v_{worst,m})$ in the constructed vector space, where each value $v_{worst,j}$ is the value of the lowest quantification allowed by the quantification type corresponding to dimension D_j, for every $j \in \mathbb{N} \wedge 1 \leq j \leq m$. Through the construction described in section IV, the worst vector coincides with the origin of the vector space. Hence, this worst vector corresponds to a collection of quantifications where each quantification expresses absolutely no compliance of the collection's corresponding answer to the user's query preferences. The intention behind the choice of the worst vector as reference vector is to force a comparison of every answer's collection of quantifications with the worst possible collection of quantifications, where answers that comply better with the user's query preferences correspond to a vector that lies further away from the worst vector. Thus, if the worst vector is chosen as reference vector, the rank of a query answer should be proportional to the distance between the vector corresponding to its collection of quantifications and the worst vector. For the vectors corresponding to the answers visualized in table III, these distances are visualized by dashed lines in figure 3.

C. *Differences*

It should be clear that, because of the use of the Euclidean distance in the presented approach, all equidistant vectors lie upon the surface of a single (hyper-)sphere, with the reference vector as a midpoint. Hence, changing the midpoint results in changing the elements of the classes of equidistant vectors and thus in changing which collections of quantifications are deemed equivalent by the ranking technique. At the moment this paper is written, further research is required to determine which choices for reference vector and distance metric result in rankings best reflecting human reasoning.

In the next section, a brief discussion about the computational efficiency associated with the presented approach is presented.

VI. Computational Efficiency

Consider a result set to a user's query, containing n elements, where each element is assigned m quantifications of how well the element's answer complies with the user's query. To determine the computational efficiency of the approach to determine a ranking for these elements, as introduced in section IV, two arguments must be taken into account.

First, it is important to notice that, for a given result set element, the determination of its rank is independent from the other elements of the result set or their quantifications, but only depends on the quantifications in its own collection. Hence, for a single result set element, the determination of its rank has an execution time of $O(m)$.

Second, to determine a ranking for the complete result set, the rank for every result set element must be determined. Hence, the determination of the rank of every element in the result set may be done in $O(m * n)$.

Notoriously left out of this reasoning is the computational efficiency of ordering the result set elements based on their assigned ranks before presenting them to the user. However, every querying technique in which result set elements are assigned a rank, needs to provide for such ordering, and the order of computational efficiency presented above only has the intention to highlight the computational burden added by the approach presented in section IV.

VII. Conclusions and Future Work

The work presented in this paper introduces a general technique to present a user with these answers to his or her query which are deemed to comply best with his or her query preferences, where compliance with the user's query preferences is indicated through the use of quantifications with different interpretations and where such quantifications with different interpretations are never combined. As mentioned before, it is the authors' opinion that for soft computing in information retrieval to reach the goal of allowing people to retrieve possibly imperfect data, information or knowledge in a possibly imperfect way, it is essential that an unambiguous presentation of the query answers best complying with the user's preferences is achieved, which is inevitably based on the aforementioned quantifications. As the authors deem it unacceptable to calculate a straightforward combination of quantifications with different interpretations, the technique presented in this paper presents another way of yet achieving an unambiguous query answer ranking.

Many aspects of the presented approach require further investigation. An inquiry about the impact of different reference vectors to which query answers are compared and about the distance metrics used in this comparison is strongly suggested, preferably with respect to the intention of mimicking human reasoning in the best possible way. Another interesting idea to investigate is the use of optimization techniques to determine a query answer's rank from its collection of quantifications.

References

[1] C. Billiet and G. De Tré, "Combining Uncertainty and Vagueness in Time Intervals," in *Advances in Intelligent Systems and Computing*, P. Angelov, K. T. Atanassov, L. Doukovska, M. Hadjiski, V. Jotsov,

J. Kacprzyk, N. Kasabov, S. Sotirov, E. Szmidt, and S. Zadrozny, Eds. Warsaw, Poland: Springer, 2014, pp. 353–364.

[2] C. Billiet, J. E. Pons Frias, O. Pons Capote, and G. De Tré, "A Comparison of Approaches to Model Uncertainty in Time Intervals," in *Advances in Intelligent Systems Research*, G. Pasi, J. Montero, and D. Ciucci, Eds., no. Eusflat. Milano, Italy: Atlantis Press, 2013, pp. 626–633.

[3] ——, "Bipolar Querying of Valid-Time Intervals Subject to Uncertainty," in *Lecture Notes in Computer Science*, H. Legind Larsen, M. Martin-Bautista, M. Amparo Vila, T. Andreasen, and H. Christiansen, Eds. Granada, Spain: Springer, 2013, pp. 401–412.

[4] C. Billiet, J. E. Pons Frias, O. Pons, and G. De Tré, *Bipolarity in the Querying of Temporal Databases*. SRI PAS/IBS PAN, 2013, no. 1, pp. 21–37.

[5] P. Bosc and O. Pivert, "Modeling and Querying Uncertain Relational Databases: A Survey of Approaches Based on the Possible Worlds Semantics," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 18, no. 5, pp. 565–603, 2010.

[6] B. P. Buckles and F. E. Petry, "A fuzzy representation of data for relational databases," *Fuzzy Sets and Systems*, vol. 7, no. 3, pp. 213–226, 1982.

[7] J. M. Medina, O. Pons, and M. A. Vila, "Gefred: A Generalized Model of Fuzzy Relational Databases," *Information Sciences*, vol. 76, no. 1-2, pp. 87–109, 1994.

[8] A. Motro and P. Smets, *Uncertainty Management in Information Systems: from Needs to Solutions*, A. Motro and P. Smets, Eds. Kluwer Academic Publishers, 1997.

[9] S. Parsons, "Current Approaches to Handling Imperfect Information in Data and Knowledge Bases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 3, pp. 353–372, 1996.

[10] P. Bosc, D. Kraft, and F. Petry, "Fuzzy Sets in Database and Information Systems: Status and Opportunities," *Fuzzy Sets and Systems*, vol. 156, no. 3, pp. 418 – 426, 2005.

[11] G. De Tré, S. Zadrozny, and A. J. Bronselaer, "Handling Bipolarity in Elementary Queries to Possibilistic Databases," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 599–612, 2010.

[12] A. Bronselaer, J. E. Pons, G. De Tré, and O. Pons, "Possibilistic evaluation of sets," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 21, no. 3, pp. 325–346, 2013.

[13] J. E. Pons, N. Marín, O. Pons, C. Billiet, and G. D. Tré, "A Relational Model for the Possibilistic Valid-time Approach," *International Journal of Computational Intelligence Systems*, vol. 5, no. 6, pp. 1068–1088, 2012.

[14] J. E. Pons, C. Billiet, O. Pons Capote, and G. De Tré, "A Possibilistic Valid-Time Model," in *Proceedings of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part I*, S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. R. Yager, Eds. Catania, Italy: Springer, 2012, pp. 420–429.

[15] J. E. Pons Frias, C. Billiet, O. Pons, and G. De Tré, *Aspects of Dealing with Imperfect Data in Temporal Databases*. Springer, 2013, ch. 9, pp. 189–220.

[16] G. De Tré, S. Zadrozny, T. Matthé, J. Kacprzyk, and A. J. Bronselaer, "Dealing with Positive and Negative Query Criteria in Fuzzy Database Querying Bipolar Satisfaction Degrees," in *Proceedings of the eighth International Conference on Flexible Query Answering Systems*, T. Andreasen, R. R. Yager, H. Bulskov, H. Christiansen, and H. L. Larsen, Eds. Denmark: Springer Verlag Berlin, 2009, pp. 593–604.

[17] J. Kacprzyk and S. Zadrozny, "Computing with Words in Intelligent Database Querying: Standalone and Internet-based Applications," *Information Sciences*, vol. 134, no. 1-4, pp. 71–109, 2001.

[18] T. Matthé and G. De Tré, "Bipolar query satisfaction using satisfaction and dissatisfaction degrees Bipolar satisfaction degrees," in *Proceedings of the 2009 ACM Symposium on Applied Computing*. ACM New York, 2009, pp. 1699–1703.

[19] T. Matthé, G. De Tré, S. Zadrozny, and J. Kacprzyk, "Bipolar Database Querying Using Bipolar Satisfaction Degrees," *International Journal of Intelligent Systems*, vol. 26, no. 10, pp. 890–910, 2011.

[20] T. Matthé, J. Nielandt, S. Zadrozny, and G. De Tré, *Constraint-Wish and Satisfied-Dissatisfied: An Overview of Two Approaches for Dealing with Bipolar Querying*. Springer, 2014, ch. 2, pp. 21–44.

[21] D. Dubois and H. Prade, "Gradualness, Uncertainty and Bipolarity: Making Sense of Fuzzy Sets," *Fuzzy Sets and Systems*, vol. 192, no. 1, pp. 3–24, 2012.

[22] ——, *Bipolar Representations in Reasoning, Knowledge Extraction and Decision Processes*. Springer, 2006, pp. 15–26.

[23] M. Dziedzic, C. Billiet, J. Kacprzyk, S. Zadrozny, and G. De Tré, "Inner and Outer Bipolarity in Database Querying," in *Proceedings of the IEEE 2014 Conference on Norbert Wiener in the 21st Century*. Boston, U.S.A.: IEEE, 2014, p. 8.

[24] D. Dubois and H. Prade, *Formal Representations of Uncertainty*. London, U.K.: Wiley, 2009, ch. 3, p. 59.

[25] ——, "The Three Semantics of Fuzzy Sets," *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 141–150, 1997.

[26] "Publications of a. n. kolmogorov," *The Annals of Probability*, vol. 17, no. 3, pp. 945–964, 07 1989.

[27] A. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, 1933.

[28] D. Dubois, F. Esteva, L. Godo, and H. Prade, *An Information-based Discussion of Vagueness: Six Scenarios Leading to Vagueness*. Elsevier, 2005, ch. 40, pp. 891–909.

[29] M. Black, "Vagueness. An Exercise in Logical Analysis," *Philosophy of Science*, vol. 4, no. 4, pp. 427–455, 1937.