

FUZZY QUERYING: ISSUES AND PERSPECTIVES

JANUSZ KACPRZYK, GABRIELLA PASI, PETER VOJTÁŠ
AND SŁAWOMIR ZADROŻNY

1. INTRODUCTION

The term *query* is widely used in the database as well as information retrieval communities. Basically, a query against a collection of information items (to be called later, for brevity, an information source) provides a formal description of the items of interest to the user posing this query. A source of information is meant here very generally. It may take the form of an archive of multimedia or textual documents, a database, or a knowledge base. In the three previous examples the information items are *documents*, *records* (rows in relational data model) and *facts*, respectively. In order to manage and access an information source, an appropriate system is defined which makes it possible to store, represent and retrieve information items by means of a formal *query language*. Information systems that make it possible to manage information items previously mentioned are *information retrieval systems*, *data base management systems* and *knowledge based systems*, respectively. Query languages of these systems usually refer to some features of entities represented by the items stored in an information source, e. g., keywords (index terms) in textual documents (documents archive), attributes (database) or arguments of facts (knowledge base). Thus, basically, a query may be seen as a set of selection conditions that should be met by an information item (its features) to be qualified as relevant with respect to the query. On the other hand, the *query processing* itself may be seen as consisting mainly of *matching* a query against the items of the information source. This process may be essentially more complex, as, e. g., in the case of knowledge bases where we deal with a whole chain of matching within the reasoning process.

Often, a user faces the problem of how to express her or his information requirements in a formal query language supported by a given information system interface. These formal languages usually require a *crisp* (*precise, unambiguous*) specification of a query, while, for human beings, a query is best expressed in terms of a natural language – a very powerful, but ambiguous and imprecise medium. Thus, adding some flexibility to traditional querying systems seems to be a critical issue for enhancing their effectiveness and efficiency.

In this paper, we discuss some recent advances and basic issues related to flexible

querying based on the application of fuzzy logic. We focus on two areas corresponding to the type of information source under consideration, namely: information retrieval in which we primarily deal with archives of textual documents and database querying. Both areas share the same interest in fuzzy (linguistic) queries and flexible matching against items of information. However, they have also their specific features, and these are pointed out in the next sections. The third area, that of very broadly meant knowledge bases querying is dealt with in the paper by Peter Vojtáš, in this special issue. Specifically, the concept of matching, essential for querying, may be identified to some extent with the *unification*. In the mentioned paper, the issues related to the fuzzy unification are discussed. The matching of fuzzy concepts, from a slightly different perspective, is also the subject of the paper by Andrejková, in this issue. Another contribution relevant for the flexible querying of knowledge bases is the paper by Ch. Marsala, in this issue.

Moreover, beside its application to querying itself, the concept of flexibility is usually extended to the representation of information to be queried. This is particularly evident in the area of information retrieval in which concepts of fuzzy logic fit very well into advanced indexing schemes for text documents. In case of database management systems, fuzzy logic based ideas have led to the development of imprecise/vague data representation models. These issues are also dealt with in the following sections.

This paper is structured in two sections dealing with information retrieval and database querying, respectively. The paper is meant to provide a synthetic description of the research area of the papers appearing in this special issue of the *Kybernetika*. This issue is comprised of extended versions of selected papers presented at the session on fuzzy querying at the FSTA'2000 Conference held in Liptovský Mikuláš (Slovak Republic) in the winter of 2000. We refer to the other papers in this issue indicating their relevance for the topics discussed here.

2. FLEXIBLE QUERYING IN INFORMATION RETRIEVAL SYSTEMS

The information source considered in this section consists of an archive of textual documents. Usually, in an Information Retrieval System (IRS), both documents and queries are formally represented by means of terms, named in the following index and query terms, respectively. The indexing procedure adopted by the IRS produces the formal documents' representations, while queries are directly specified by users by means of the system's query language. The so-called matching mechanism compares these two formal representations to the aim of estimating the relevance (or the probability of relevance in probabilistic models) of documents to the considered query. This is a complex task, pervaded with vagueness and uncertainty. Most of the existing IRSs are based on simple retrieval models with the main consequence of privileging efficiency at the expense of effectiveness. A promising direction to improve these systems is to model subjectivity intrinsic in the interpretation and selection of information and to make the systems adaptive, i. e. being able to learn the users' concept of relevance. Fuzzy set theory has provided a suitable and natural support to define information retrieval systems tolerant to vagueness. In particular,

it has been applied to different purposes that can be categorized as: definition of flexible interfaces (flexible query languages), flexible representation of structured documents, and definition of associative mechanisms such as thesauri, clustering and relevance feedback.

In the following we will describe some applications of fuzzy set theory for defining flexible query languages in generalized Boolean models. For a more extensive survey of the application of fuzzy set theory to information retrieval see [6, 23]. A more recent application of fuzzy set theory to information retrieval is aimed at defining linguistic information retrieval models in which both at the level of documents' representation and at the level of query language the importance of terms is linguistically specified [7].

Before describing some applications of fuzzy sets theory to define flexible query languages we briefly introduce the indexing problem. Usually, the indexing procedure is based on the selection of a set of index terms used to represent the document contents. In the Boolean model of information retrieval documents are represented as sets of index terms. The vector space model, some probabilistic models and fuzzy models adopt a weighted document representation that is based on the association of a numeric weight with each index term [6, 14, 17, 27, 28, 29]. This numeric weight is interpreted as an indicator of importance of an index term as a document's descriptor. The automatic computation of the index term weights is performed by a so-called indexing function F that is based on a count of the occurrences of a term in the document and in the whole archive. Function F is generally defined so as it is increasing with the frequency of a term t in a document d , and decreasing with the frequency of a term in all documents of the archive [4, 10, 11]. An example of definition of function F is the following: $F(d, t) = tf_{dt} \times IDF_t$ where: tf_{dt} is a normalized term frequency and IDF_t is an inverse document frequency [29, 33]. From a mathematical point of view, in fuzzy IR models a document is represented as a fuzzy set of terms: $R_d = \sum \mu_d(t) \setminus t$.

A more recent approach to weighted indexing is based on the consideration that the formal representation of a document should be generated by taking into account the subjective view of a user. In the case in which an archive is constituted by documents structured in logical sections, the usefulness of sections to the users as potential carriers of relevant information can vary depending on users' needs. To this aim, an adaptive indexing model has been proposed in [5], which allows users to determine the formal document representation according to their search interests. This indexing model produces a weighted document representation which can be "tuned" by the user in the phase of query formulation. In this special issue, the contribution by Bordogna and Pasi proposes a flexible query language, that allows users to express soft content-based selection conditions on the adaptive representation of structured documents. By means of this query language a user may indicate the preferred sections of documents, i. e. those which he/she estimates bearing the most interesting information. Further the user can linguistically quantify the number of sections which determine the global potential interest of the documents through the specification of a linguistic quantifier such as *all*, *most of*, *at least 70% of*, etc. expressing strict or more relaxed constraints [35, 37]. In this special issue, a flexible

query language is defined, that allows users to express soft content-based selection conditions on the structured documents. In this way, users can tune the view of the document content used by the retrieval mechanism to determine the document relevance.

It is important to underline that the adoption of a weighted document representation constitutes the basis for most of the fuzzy generalizations of the Boolean query language which we are going to describe in the following.

When using an IRS, the main problem in the query formulation phase is the reduction of a vague expression in natural language into an expression of a formal language. A general structure of the expressions of a query language can be described by means of a set of selection conditions connected through aggregation operators:

$$sel_1 \text{ AG}_1 sel_2 \text{ AG}_2 sel_3 \dots$$

where sel_i are the selection conditions and AG_i are aggregation operators. The selection conditions are constraints on the formal document representations: the matching mechanism verifies the satisfaction of the specified constraints by each document. Aggregation operators are employed to express more complex selection conditions. The most widely adopted query language is the Boolean language in which both the selection conditions and the aggregation operators are "crisp": a user can specify a set of query terms as descriptors of her/his needs, and aggregate them by the AND or the OR operators (let us think about the query languages of the search engines on the Internet). This kind of language does not allow any level of vagueness – on the contrary, it requires an extreme precision in the selection of search terms, and forces the user to request all or at least one of the query terms to be present in desired documents. The concept of relevance modelled here is binary, and corresponds to the presence of query terms in document representations. Other approaches to querying can be based either on the specification of selection conditions with no aggregation operators (which are assumed by default) or as an expression in natural language from which the query terms are automatically extracted.

Fuzzy sets theory has been successfully employed for generalizing the Boolean query language so as to define flexible query languages tolerant to vagueness and imprecision inherent in the human communication. Fuzzy set theory has been applied to both the levels of specification of selection conditions and of aggregation operators. With respect to the former aspect, the aim has been to define selection conditions as *soft constraints* on weighted document representations. In the very first proposals numeric query term weights have been defined in order to allow the specification of a different importance of search terms as descriptors of the sought documents [1, 2, 13, 23, 34]. Distinct semantics have been associated with these weights such as a relative importance (weights specify the relative importance of terms), threshold (weights specify threshold that the index term weights should overcome), and ideal index term weights (weights specify the desired values of index term). For details on these semantics see [6, 23].

The main limitation of numeric query weights is that they force the user to quantify the qualitative and vague concept of importance. It is much easier to directly use linguistic descriptors such as *important*, *very important*, *fairly important*, etc. The

objective of the first linguistic extension of the Boolean query language in [3, 22] has been to define a fuzzy retrieval model with linguistic query weights. In this approach, the linguistic descriptors are formalized through linguistic variables. A selection condition is specified by a pair $\langle t, l \rangle$, where t is a term belonging to the set T of terms and l is a value of the linguistic variable *Importance*, e. g. $\langle \textit{politics}, \textit{important} \rangle$. The linguistic variable *Importance* is defined over a base variable ranging over the set $[0,1]$. In this way the matching mechanism evaluates the compatibility between the linguistic query weights and the numeric index terms weights.

The definition of a set of linguistic weights, i. e. the term set $T(\textit{Importance})$, should be done so as to provide users with a few words by which they can naturally formulate their information needs. One primary term, *important*, can be defined; the other linguistic descriptors are obtained by attaching linguistic modifiers such as *very*, *fairly*, *not*, etc. to the primary term: $T(\textit{Importance}) = \{\textit{important}, \textit{very important}, \textit{not important}, \textit{fairly important}, \dots\}$. A context-free grammar has been defined in [3] for generating all values belonging to the term set $T(\textit{Importance})$.

The second level of definition of queries concerns the specification of aggregation operators. The use of a flexible aggregation operator may be helpful either when users do not have a precise idea of what they are looking for or when they can tolerate an undersatisfaction of some conditions. The AND and OR connectives allow only for crisp aggregations which do not capture any vagueness. For example, the AND used for aggregating n selection criteria does not allow to tolerate the non-satisfaction of a single criterion; this may cause the rejection of useful items.

In this respect, within the framework of fuzzy set theory, a generalization of the Boolean query language has been defined [4] based on the concept of linguistic quantifiers: they are employed to specify both crisp and soft aggregation of selection criteria. New aggregation operators, with a self-expressive meaning such as *at least n* and *most of*, are defined with a behavior between the two extremes corresponding to the AND and the OR connectives. By means of linguistic quantifiers the requirements of a complex Boolean query are more easily and intuitively formulated. For example when one wishes that at least 2 out of the three criteria “politics”, “economy” and “inflation” be satisfied, one should formulate the following Boolean query:

(politics AND economy) OR (politics AND inflation) OR (economy AND inflation)

which can be replaced by the simpler one:

at least 2(politics, economy, inflation).

The ordered weighted averaging (OWA) operators have been used to define the linguistic quantifiers [35]. Besides the quantifier *at least k* defined as a crisp threshold, other quantifiers with a vague meaning are defined. The quantifier *almost k* may be interpreted as a soft threshold, i. e., a document is deemed as relevant to the query even if satisfies slightly less than k criteria. The quantifier *more than k* specifies that the higher the number of the criteria above k which are satisfied, the higher the overall satisfaction value.

Another extension of the Boolean query language concerns the specification of optional selection conditions. In some cases, the order in which the selection conditions are connected through the AND operator reflects an implicit priority among them. The selection criteria that are listed first in the query are in some sense considered essential to characterize topics of interest, while those that appear later are considered optional. What is desired is that the retrieval results depend on the satisfaction of essential criteria, while the overall relevance must also be conditioned by the satisfaction of the optional criteria. The *and possibly* operator has been defined to ask for optional selection criteria in relation to essential ones. An optional criterion affects only the degree of relevance of documents retrieved; it acts as a filter on items that satisfy the essential criteria. For example, to express interest in documents dealing with “expert” “systems” (essential criteria), while declaring a greater interest for those of such documents that also deal with “fuzzy” or “ANN” (optional criteria), the following query can be formulated:

all(expert, systems) and possibly at least 1 (fuzzy, ANN).

The *and possibly* operator has been defined in [4] as a non-monotonic intersection and provides a further level of softening of the retrieval mechanism, not discarding documents which satisfy only the essential criteria.

3. FUZZINESS IN DATABASE QUERYING

Database management systems are meant for the maintenance and processing of highly structured data. We will limit our discussion here to the case of the *relational data model*. Nevertheless, it is worth noticing that the position of this model has been recently challenged by the *object-oriented data model*. The latter, no less than the former, requires special measures for the representation of imprecise, uncertain etc. data and flexible querying capabilities. Recent advances in this area are presented in [15].

In the relational database model, the main structure used for the representation of the class of real world entities is the *table*. A table is defined by a set of *columns* corresponding to the attributes of the modelled entities. Each entity is represented by a *row (tuple)* in the table. The relationship between classes of entities is represented either by a separate table or by two columns corresponding to each other and defined in the tables modelling these classes of entities.

Basically, we can distinguish three main groups of fuzzy logic based approaches to the representation of imperfect information within the framework of the relational data model. The first approach is a natural extension to the main notion laying ground for the relational data model, i. e., the notion of the relation. Namely, instead of the crisp relation it is postulated to employ the fuzzy relation that allows to express some imperfect information about the items represented by particular tuples of such a relation. The querying of such a fuzzy database relies on the algebra of fuzzy relations. This approach is mainly undermined by the lack of a clear, unambiguous interpretation of the grade of membership of a tuple in a relation, cf., e. g., [25].

Two other approaches preserve the crispness of the relations and represent the imperfect information directly at the level of values of the attributes. The value of an attribute is still assumed to be atomic, single element belonging to the attribute's domain, but information about this value may be imperfect. In the second approach, proposed by Buckles and Petry [12], this value is represented as a crisp subset of the attribute's domain. In the third approach, proposed by Prade and Testemale [26], this value is represented as a possibility distribution on the attribute's domain. In the approach of Buckley and Petry (and their followers further developing the idea cf., e.g., [30]) the domain of an attribute is additionally equipped with a fuzzy similarity relation indicating how similar to each other are all pairs of the values of a given attribute. In order to explain the role of a similarity relation we have to refer to the basic database concept of *redundancy*. Namely, two tuples are considered redundant when the values of all of their attributes are *identical*. The similarity relation is meant to replace the *identity* relation when deciding about the redundancy of two tuples. Thus, to decide if two tuples are redundant we first merge the values of all their attributes (i.e., the union of the sets of values is taken) and then check if a minimal similarity degree for each attribute is higher than a certain threshold. If the answer is positive, then these two tuples are considered redundant and are replaced by one tuple obtained by merging them. The redundant tuples are likely to appear in the results of queries containing the projection operator (present in most queries).

The relational data model encompasses two intrinsic theoretical formalisms for the queries, cf. [32]. The first one is based on *the relational algebra* and supports the query as a composition of relational operators, such as *selection*, *projection*, *join* etc., acting on some tables and producing the requested information. The second approach is based on a *relational calculus* and can have two flavours: *calculus of domains* and *calculus of tuples*. Basically, in both approaches the query is expressed as a formula in which the predicate symbols correspond to relations defined in a database and the variables represent elements of domains (in the former approach) or whole tuples of the relations (in the latter approach). The set of tuples satisfying the formula comprises the result of such a query. The relational algebra based approach is *procedural*. Namely, the user has to determine *how* the requested information has to be found in a database. The relational calculus, on the other hand, makes it possible to describe just *what* information is requested. Neither of the approaches, in their strict form, has turned out to be practical. Currently, the de-facto industry standard for the querying of relational databases is the Structured Query Language (SQL) that may be considered to be a mixture of both the above more theoretical approaches. In our further discussion of fuzzy querying we will refer mostly to the SQL related concepts.

Queries in SQL are expressed by just one but a very powerful command: *SELECT*. In its generic form, the *SELECT* command combines data from tables producing another table as the result. The command is built of several clauses specifying which tables and columns should be taken into account and how the tables should be joined (the *FROM* and *SELECT* clauses), what are the conditions the requested rows should satisfy (the *WHERE* clause), how the rows should be grouped and what

conditions should the groups meet (the GROUP BY and HAVING clauses) and how the resulting table should be ordered (the ORDER BY clause). In terms of relational algebra the SELECT, FROM and WHERE clauses correspond to the projection, join and selection operations, respectively. From the fuzzy querying perspective the most interesting is the WHERE clause. Here are specified the requirements the information sought should meet and, as we argue in the introduction, these conditions very often are prone to imprecision, vagueness etc.

A natural applicability of fuzzy logic based approaches in the context of database querying has been advocated by numerous researchers starting, presumably, with Tahani, [31]. The concept of fuzzy querying may be considered either in the context of a crisp, traditional database or a fuzzy database. Let us first discuss the latter case. Depending on the type of a fuzzy database we obtain various interpretations for fuzzy queries. In case of a similarity based approach, the query may be fuzzified through the use of terms of the type “*more or less*”. Namely, such a term indicates that a complete fulfillment of the query conditions is not required. For example, let the query contain the condition “**attribute more or less equal to value**”. Thus, the query evaluation mechanism may exploit the similarity relation defined for the **attribute** and retrieve not only rows for which its value is exactly as used in the query but also these rows where the value of the attribute is similar enough. An analogous approach to fuzzy querying in the context of a crisp database has been proposed by several authors, cf., e. g., [24].

In case of a possibility based approach it is usually assumed that a query may contain vague terms like “high”, “cheap” etc. These terms are represented by fuzzy sets defined in domains of corresponding attributes. Thus, the querying engine has to assess how a fuzzy set used in a query is compatible with the value of an attribute represented by the possibility distribution. Natural candidates for the compatibility assessment are the possibility and necessity measures [16]. We will discuss this issue later on. Now we will focus on the case of fuzzy querying in the context of a classical, crisp database. This direction seems to be more practical and promising.

The development of a fuzzy querying system may be meant as devising a query language directly referring to fuzzy concepts. Four basic, sometimes inter-related issues relevant for such a language may be considered:

- its syntax,
- representation of fuzzy (linguistic) terms to be directly used in the queries,
- semantics of the language, basically boiling down to the question of the calculation of a matching degree,
- efficiency of the implementation.

These issues are discussed in the following paragraphs.

Two leading application oriented approaches choose the SQL as the starting point for the construction of a flexible query language, cf. [9, 10] and [18, 19]. Both proposals advocate a direct use of linguistic terms within the SQL SELECT command. Bosc et al have proposed an extension of all constructs of this command that may be fuzzified in a reasonable way. These encompass the conditions of the WHERE clause including a graded membership in the result set of a subquery as well as the

fuzzy joins. Kacprzyk et al focused on the interpretation of fuzzy terms within the WHERE clause putting emphasis on the use of linguistic quantifiers as non-standard aggregation operators supplementing the classical AND and OR connectives.

Fuzzy (linguistic) terms of a fuzzy query language are represented as fuzzy sets. These may be *fuzzy values* (e. g., “low”), *fuzzy relations (comparators)*, exemplified by “*much greater than*” or *fuzzy (linguistic) quantifiers* (e. g., “*most*”). In case of the fuzzy values corresponding to the numeric attributes, they are usually represented by trapezoidal fuzzy numbers defined on the domains of the attributes. Kacprzyk et al introduced the concept of a *universal fuzzy value* that may be defined on a unified interval, e. g., $[-10, 10]$. Additionally, for each numeric attribute its range, i. e., lowest and highest possible values, has to be provided (these values may be subjectively assessed). Then, during the query processing a fuzzy value’s membership function is automatically adapted to the range of the attribute associated with this fuzzy value in the query condition. Fuzzy relations are defined on the Cartesian products of appropriate domains. In case of numeric attributes, a fuzzy relation may also be defined on the universe being an interval whose lower and upper bound are, respectively, the smallest and the highest possible differences of the involved attributes. In such a case it is also possible to introduce *universal* definitions of fuzzy relations. The fuzzy (linguistic) quantifiers may be interpreted in the sense of Zadeh [37] or via various aggregation schemes, notably Yager’s OWA operators.

One of the distinguishing features of fuzzy querying is the concept of a matching degree belonging to the $[0,1]$ interval. The fuzzy query evaluation against a crisp database may be considered as a special case of a more general and complex case of fuzzy, possibility based databases [16]. In the latter case we deal with imperfect information both in the query and in the database. Namely, the query may contain linguistic terms represented by the fuzzy sets and the values of attributes in the database may be represented by possibility distributions. Then, a simple query condition may be expressed as the requirement that a numeric attribute value, represented by the possibility distribution $\pi(u)$, matches a soft constraint represented by the fuzzy set P . The matching degree is evaluated using two measures:

$$\text{Possibility of matching: } \Pi(P) = \sup_{u \in U} \min(\pi(u), \mu_P(u)) \quad (1)$$

$$\text{Necessity of matching: } N(P) = \inf_{u \in U} \max(1 - \pi(u), \mu_P(u)). \quad (2)$$

In case of a crisp database the possibility distribution is replaced by a single value u_0 what corresponds to the following possibility distribution: $\pi(u_0) = 1$ and $\pi(u) = 0 \forall u \neq u_0$. Then, both (1) and (2) reduce to $\Pi(P) = N(P) = \mu_P(u_0)$. Hence, while calculating the matching degree of a simple condition of the form: *attribute IS fuzzy set*, exemplified by “*price IS low*”, we take the membership degree of the attribute value in the fuzzy set employed in the condition. Obviously, this is applicable for both numeric and scalar attributes. A simple condition referring to a fuzzy relation is processed in the way described earlier. A special treatment is required in case of subqueries – for a detailed discussion of a possible approach see [10]. Finally, a required aggregation operator is applied to the partial matching degrees computed for simple conditions. Usually, it is assumed that a certain threshold is specified in

the query limiting the rows in the result set to those matching the query to a degree that is above that threshold.

During the last two or three decades of the research in the field of database management systems the efficiency of querying engines has been one of crucial issues. Many effective indexing schemes, query optimization algorithms etc. have been devised. Processing of fuzzy queries introduces an additional computational burden due to a substantial amount of metadata to be processed (linguistic terms), more complex matching evaluation etc. These topics have been studied but to a rather limited extent. A naive approach consists in the full search of the database or, more precisely, tables referred to in a query. Bosc et al propose a derivation procedure that aims at constructing a crisp query producing results that cover the result set of the original fuzzy query, see e. g., [11]. Such a crisp query may employ a full power of the query optimization provided by the host database management system. Then, a fuzzy matching degree is computed for the particular rows of the, usually substantially smaller, result set of this crisp query. Yazici [36] studies the construction of indices taking into account fuzzy (linguistic) terms. Kacprzyk and Zadrozny have proposed an experimental implementation of fuzzy querying capabilities on top of the popular, commercial, desktop database management system, Microsoft Access. The same authors have also studied issues related to the implementation of a similar approach to fuzzy querying in the Internet environment.

This short presentation does not exhaust a wide spectrum of problems discussed in relation to the concept of fuzzy querying. For example, Bosc et al have recently [8] proposed a novel category of fuzzy queries posed against a possibility based fuzzy database. Namely, so far we have assumed that a simple condition of the query identifies a constraint on the *value* of an attribute while the newly proposed approach sets a constraint on the “*shape*” of the value (more precisely: on the form of the possibility distribution representing the value). Kacprzyk and Zadrozny (see this issue) investigate the possibility of combining fuzzy querying with data mining tasks. Damiani et al (see this issue) discuss the application of fuzzy logic based approaches for querying meant in a different perspective. They assume that a query is posed against a collection of information sources (here, the XML documents but the basic ideas of their approach are applicable also in the context of databases) that, while collecting data on the objects of the same type, may represent them in a different way. A fuzzy logic based approach is applied to circumvent this heterogeneity of data representation when processing a query.

(Received October 6, 2000.)

REFERENCES

- [1] A. Bookstein: Fuzzy requests: an approach to weighted boolean searches. *J. Amer. Soc. Inform. Science* 31 (1980), 4, 240–247.
- [2] G. Bordogna, P. Carrara and G. Pasi: Query term weights as constraints in fuzzy information retrieval. *Inform. Process. Management* 27 (1991), 1, 15–26.
- [3] G. Bordogna and G. Pasi: A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation. *J. Amer. Soc. Inform. Science* 44 (1993), 2, 70–82.

- [4] G. Bordogna and G. Pasi: Linguistic aggregation operators in fuzzy information retrieval. *Internat. J. Intelligent Systems* 10 (1995), 2, 233–248.
- [5] G. Bordogna and G. Pasi: Controlling Information Retrieval through a user adaptive representation of documents. *Internat. J. Approx. Reason.* 12 (1995), 317–339.
- [6] G. Bordogna and G. Pasi: The Application of Fuzzy Set Theory to Model Information Retrieval. In: *Soft Computing in Information Retrieval: Techniques and Applications* (F. Crestani and G. Pasi, eds.), Physica-Verlag, Heidelberg 2000.
- [7] G. Bordogna and G. Pasi: Linguistic granules to express importance in an ordinal information retrieval model. In: *Proceedings of the Eighth International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'2000)*, Madrid 2000, pp. 470–476.
- [8] P. Bosc, L. Duval and O. Pivert: Value-based and representation-based querying of possibilistic databases. In: *Recent Issues on the Management of Fuzziness in Databases* (G. Bordogna and G. Pasi, eds.), Physica-Verlag, Heidelberg 2000, pp. 3–28.
- [9] P. Bosc and O. Pivert: Fuzzy querying in conventional databases. In: *Fuzzy Logic for the Management of Uncertainty* (L. A. Zadeh and J. Kacprzyk, eds.), Wiley, New York 1992, pp. 645–671.
- [10] P. Bosc and O. Pivert: SQLf: a relational database language for fuzzy querying. *IEEE Trans. Fuzzy Systems* 3 (1995), 1–17.
- [11] P. Bosc and O. Pivert: SQLf query functionality on top of a regular relational database management system. In: *Knowledge Management in Fuzzy Databases* (O. Pons, M. A. Vila and J. Kacprzyk, eds.), Physica Verlag, Heidelberg 2000, pp. 171–190.
- [12] B. P. Buckles and F. E. Petry: A fuzzy model for relational databases. *Fuzzy Sets and Systems* 7 (1982), 213–226.
- [13] D. A. Buell and D. H. Kraft: A model for a weighted retrieval system. *J. Amer. Soc. for Inform. Science* 32 (1981), 3, 211–216.
- [14] F. Crestani, M. Lalmas, C. J. van Rijsbergen and I. Campbell: “Is this document relevant? ... probably”: a survey of probabilistic models in information retrieval. *ACM Comput. Surveys* 30 (1998), 4, 528–552.
- [15] R. De Caluwe (ed.): *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*. *Adv. in Fuzzy Systems – Appl. and Theory* 13 (1998). World Scientific Pub Co.
- [16] D. Dubois and H. Prade: Tolerant fuzzy pattern matching: an introduction. In: *Fuzziness in Database Management Systems* (P. Bosc and J. Kacprzyk, eds.), Physica-Verlag (Springer-Verlag) 1995, pp. 42–58.
- [17] N. Fuhr: Models for retrieval with probabilistic indexing. *Inform. Process. Management* 25 (1989), 1, 55–72.
- [18] J. Kacprzyk and S. Zadrożny: Fuzzy querying for Microsoft Access. In: *Proceedings of Third IEEE Conference on Fuzzy Systems Orlando 1994*, Vol. 1, pp. 167–171.
- [19] J. Kacprzyk and S. Zadrożny: FQUERY for Access: fuzzy querying for a Windows – based DBMS. In: *Fuzziness in Database Management Systems* (P. Bosc and J. Kacprzyk, eds.), Physica-Verlag, Heidelberg 1995, pp. 415–433.
- [20] J. Kacprzyk, S. Zadrożny and A. Ziółkowski: FQUERY III+: a ‘human consistent’ database querying system based on fuzzy logic with linguistic quantifiers. *Inform. Systems* 6 (1989), 443–453.
- [21] J. Kacprzyk and A. Ziółkowski: Database queries with fuzzy linguistic quantifiers. *IEEE Trans. Systems Man Cybernet.* SMC-16 (1986), 474–479.
- [22] D. H. Kraft, G. Bordogna and G. Pasi: An extended fuzzy linguistic approach to generalize Boolean information retrieval. *J. Inform. Sci. Appl.* 2 (1995), 3, 119–134.
- [23] D. Kraft, G. Bordogna and G. Pasi: Fuzzy Set Techniques in Information Retrieval. In: *Fuzzy Sets in Approximate Reasoning and Information Systems* (J. C. Bezdek,

- D. Dubois and H. Prade, eds.), The Handbooks of Fuzzy Sets Series, Kluwer Academic Publishers, Boston – Dordrecht – London 1999, pp. 469–510.
- [24] A. Motro: VAGUE: A user interface to relational database that permits vague queries. *ACM Trans. Office Inform. Systems* 6 (1988), 3, 187–214.
- [25] F. E. Petry: *Fuzzy Databases. Principles and Applications*. Kluwer Academic Publishers, Boston – Dordrecht – London 1996.
- [26] H. Prade and C. Testemale: Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. *Inform. Sci.* 34 (1984), 115–143.
- [27] G. Salton: *Automatic text processing: The transformation, analysis and retrieval of information by computer*. Addison Wesley, Reading 1989.
- [28] G. Salton and C. Buckley: Term weighting approaches in automatic text retrieval. *Inform. Process. Management* 24 (1988), 5, 513–523.
- [29] G. Salton and M. J. McGill: *Introduction to modern information retrieval*. McGraw-Hill, New York 1983.
- [30] S. Shenoj and A. Melton: Proximity relations in the fuzzy relational database model. *Fuzzy Sets and Systems* 31 (1989), 285–296.
- [31] V. Tahani: A conceptual framework for fuzzy query processing: a step toward very intelligent database systems. *Inform. Process. Management* 13 (1977), 289–303.
- [32] J. D. Ullman: *Principles of Database Systems*. Computer Science Press, Rockville 1982.
- [33] C. J. Van Rijsbergen: *Information Retrieval*. Butterworths & Co., Ltd, London 1979.
- [34] R. R. Yager: A note on weighted queries in information retrieval systems. *J. Amer. Soc. Inform. Sci.* 38 (1987), 1, 23–24.
- [35] R. R. Yager: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems Man Cybernet.* 18 (1988), 1, 183–190.
- [36] A. Yazici and D. Cibiceli: An access structure for similarity-based databases. *Inform. Sci.* 115 (1999), 1–4, 137–163.
- [37] L. A. Zadeh: A computational approach to fuzzy quantifiers in natural languages. *Computers and Math. Appl.* 9 (1983), 149–184.

*Prof. Dr. Janusz Kacprzyk and Dr. Sławomir Zadrozny, Systems Research Institute, Polish Academy of Sciences, and University of Information Technology and Management, ul. Newelska 6, 01-447 Warszawa. Poland.
e-mail: kacprzyk,zadrozny@ibspan.waw.pl*

*Dr. Gabriella Pasi, Istituto per le Tecnologie Informatiche Multimediali CNR, via Ampère 56, 20131 Milano. Italy.
e-mail: gabriella.pasi@itim.mi.cnr.it*

*Doc. RNDr. Peter Vojtáš, DrSc., Department of Computer Science, Faculty of Science, P. J. Šafárik University, Jesenná 5, 041 54 Košice. Slovakia.
e-mail: vojtas@kosice.upjs.sk*