Protoforms of Linguistic Database Summaries as a Human Consistent Tool for Using Natural Language in Data Mining

Janusz Kacprzyk, Polish Academy of Sciences, Poland

Sławomir Zadrożny, Polish Academy of Sciences, Poland

ABSTRACT

We consider linguistic database summaries in the sense of Yager (1982), in an implementable form proposed by Kacprzyk & Yager (2001) and Kacprzyk, Yager & Zadrożny (2000), exemplified by, for a personnel database, "most employees are young and well paid" (with some degree of truth) and their extensions as a very general tool for a human consistent summarization of large data sets. We advocate the use of the concept of a protoform (prototypical form), vividly advocated by Zadeh and shown by Kacprzyk & Zadrożny (2005) as a general form of a linguistic data summary. Then, we present an extension of our interactive approach to fuzzy linguistic summaries, based on fuzzy logic and fuzzy database queries with linguistic quantifiers. We show how fuzzy queries are related to linguistic summaries, and that one can introduce a hierarchy of protoforms, or abstract summaries in the sense of latest Zadeh's (2002) ideas meant mainly for increasing deduction capabilities of search engines. We show an implementation for the summarization of Web server logs.

Keywords: *computing with words and perceptions; data mining; fuzzy logic; fuzzy querying; linguistic summarization; protoform*

INTRODUCTION

Data summarization is one of basic capabilities needed by any "intelligent" system. Since for the human being the only fully natural means of communication is natural language, a linguistic summarization would be very desirable, exemplified by, for a data set on employees, a statement (linguistic summary) "almost all young and well qualified employees are well paid".

This may clearly be an instance of a paradigm shift that is advocated in recent time whose prominent example is the so-called "comput-

ing with words (and perceptions) paradigm" introduced by Zadeh in the mid-1990s, and extensively presented in Zadeh & Kacprzyk's (1999) books.

Unfortunately, data summarization is still in general unsolved a problem. Very many techniques are available but they are not "intelligent enough", and not human-consistent, partly due to a limited use of natural language.

We show here the use of linguistic database summaries introduced by Yager (1982, 1991, 1995, 1996), and then considerably advanced by Kacprzyk (2000), Kacprzyk & Yager (2001), and Kacprzyk, Yager & Zadrożny (2000, 2001), Zadrożny & Kacprzyk (1999), and implemented in Kacprzyk & Zadrożny (1998,2000a-d, 2001a-e, 2002, 2003, 2005). We derive here linguistic data summaries as linguistically quantified propositions as, e.g., "most of the employees are young and well paid", with a degree of truth (validity), in case of a personnel database.

We employ Kacprzyk & Zadrożny's (1998, 2000a-d, 2001) interactive approach to linguistic summaries in which the determination of a class of summaries of interest is done via Kacprzyk & Zadrożny's (1994, 1995a-b, 2001b) FQUERY for Access, a fuzzy querying add-in to Microsoft Access, extended to the querying over the Internet in Kacprzyk & Zadrożny (2000b). Since a fully automatic generation of linguistic summaries is not feasible at present, an interaction with the user is assumed for the determination of a class of summaries of interest, and this is done via the above fuzzy querying add-in.

Extending Kacprzyk & Zadrożny (2002), we show that by relating various types of linguistic summaries to fuzzy queries, with various known and sought elements, we can arrive at a hierarchy of prototypical forms, or – in Zadeh's (2002) terminology – protoforms, of linguistic data summaries. This seems to be a very powerful conceptual idea.

We present an implementation of the proposed approach to the derivation of linguistic summaries for Web server logs. This implementation may be viewed as a step towards the implementation of protoforms of linguistic summaries.

LINGUISTIC SUMMARIES USING FUZZY LOGIC WITH LINGUISTIC QUANTIFIERS

In Yager's (1982) approach, we have:

- V is a quality (attribute) of interest, e.g. salary in a database of workers,
- $Y = \{y_1, ..., y_n\}$ is a set of objects (records) that manifest quality V, e.g. the set of workers; hence $V(y_i)$ are values of quality V for object y_i,
- $D = \{V(y_1), ..., V(y_n)\}$ is a set of data (the "database" on question)

A *linguistic summary* of a data set (data base) consists of:

- a summarizer S (e.g. young),
- a quantity in agreement Q (e.g. most),
- truth T - e.g. 0.7,
- a qualifier R (optionally), i.e. another linguistic term (e.g. well-earning), determining a fuzzy subset of Y.

as, e.g., "T(*most* of employees are *young*)=0.7". The truth T may be meant more generally as, e.g., validity.

Given a set of data D, we can hypothetize any appropriate summarizer S and any quantity in agreement Q, and the assumed measure of truth will indicate the truth of the statement that Q data items satisfy S.

We assume that the summarizer S (and qualifier R) is a linguistic expression semantically represented by a fuzzy set as, e.g., "young" would be represented as a fuzzy set in {1, 2, ..., 90}. Such a simple one-attribute summarizer serves the purpose of introducing the concept of a linguistic summary but it can readily be extended to a confluence of attribute values as, e.g, "*young* and *well paid*". Clearly, the most interesting are non-trivial, *human-consistent* summarizers (concepts) as, e.g.: *productive* workers, involving complicated *combinations of attributes*, e.g.: a hierarchy (not all attributes are of the same importance), the attribute values

are ANDed and/or ORed, *k* out of *n*, *most*, etc. of them should be accounted for, etc. but they need some specific tools and techniques to be mentioned later.

Basically, two types of a linguistic quantity in agreement can be used: absolute (e.g., "about 5", "more or less 100", "several"), and relative (e.g., "a few", "more or less a half", "most", "almost all"). They are fuzzy linguistic quantifiers (cf. (Zadeh, 1983, 1985)) that can be handled by fuzzy logic.

The calculation of truth (validity) boils down to the calculation of the truth value (from [0,1]) of a linguistically quantified statement (e.g., "*most* of the employees are *young*") that can be done using Zadeh's (1983) calculus of linguistically quantified propositions (cf. (Zadeh & Kacprzyk, 1999)) or Yager's (1988) OWA operators [cf. (Yager & Kacprzyk, 1997)]; for a survey, see also Liu & Kerre (1998).

So, we have a linguistically quantified proposition, written "*Qy*'s are *F*",where *Q* is a linguistic quantifier (e.g., most), *Y*= {*y*} is a set of objects (e.g., experts), and *F* is a property (e.g., convinced). Importance *B* (later referred to with an accompanying attribute as a qualifier) may be added, yielding "*QBy*'s are *F*", e.g., "most (*Q*) of the important (*B*) experts (*y*'s) are convinced (*F*)".

We seek their truth values which are equal to: if *F* and *B* are fuzzy sets in *Y*, and a (proportional, nondecreasing) *Q* is assumed to be a fuzzy set in [0,1] as, e.g.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases} \quad (1)$$

then, due to Zadeh (1983)

$$\text{truth}(Qy's \text{ are } F) = \mu_Q[\tfrac{1}{n}\sum_{i=1}^{n}\mu_F(y_i)] \quad (2)$$

$$\text{truth}(QBy's \text{ are } F) = \mu_Q[\sum_{i=1}^{n}(\mu_B(y_i) \wedge \mu_F(y_i)) / \sum_{i=1}^{n}\mu_B(y_i)] \quad (3)$$

The OWA operators can also be used to calculate (1) and (2) – cf. (Yager, 1988; Yager & Kacprzyk (1997)). They offer a wide array of aggregation types based on various quantifiers, both crisp and fuzzy, though they may lead to more complicated calculation formulas. In the implementation presented later the user may choose between the OWAs and Zadeh's calculus.

The basic validity criterion, i.e. the degree of truth (validity), is conceptually important but often insufficient in practice. Some other quality (validity) criteria have been proposed in Kacprzyk & Yager (2001), and Kacprzyk, Yager & Zadrożny (2000), like degrees of: imprecision, covering, and appropriateness, and the length of a summary. An optimal summary is sought which maximizes the weighted average of the particular degrees.

FUZZY QUERYING, LINGUISTIC SUMMARIES, AND THEIR PROTOFORMS

The roots of our approach are our previous papers on fuzzy logic in database querying (cf. (Kacprzyk & Ziółkowski, 1986; Kacprzyk, Zadrożny & Ziółkowski, 1989)) in which we argued that the formulation of a precise query is often difficult for the end-user (see also (Kacprzyk et al., 2000)). For example, a customer of a real-estate agency looking for a house would rather use requirements using imprecise descriptions as *cheap*, *large* garden, etc. Also, to specify which combination of the criteria fulfillment would be satisfactory, one would often use, e.g., *most* or *almost all* of them. All such vague terms may be easily interpreted by fuzzy logic, hence the development of many fuzzy querying interfaces, notably our FQUERY for Access.

FQUERY for Access is an add-in that it makes possible to use fuzzy terms in queries, and the following terms are available:

- fuzzy values as by *low* in "profitability is *low*",
- fuzzy relations as by *much greater than* in "income is *much greater than* spending", and
- linguistic quantifiers as by *most* in "*most* conditions have to be met".

The first two are elementary building blocks of fuzzy queries in FQUERY for Access. They are meaningful in the context of numerical fields only. There are also other fuzzy constructs allowed which may be used with scalar fields. To use a field in a query in connection with a fuzzy value, it has to be defined as an *attribute* whose definition consists of its: lower (LL) and upper (UL) limit. They set the interval which the field's values are to belong to. This interval depends on the meaning of the given field. This makes it possible to universally define fuzzy values as fuzzy sets on [-10, +10]. Then, *the matching degree* $md(\cdot,\cdot)$ of a simple condition referring to attribute AT and fuzzy value FV against a record t is calculated by $md(\mathrm{AT} = \mathrm{FV}, t) = \mu_{\mathrm{FV}}(\tau(t(\mathrm{AT})))$, where: $t(\mathrm{AT})$ is the value of attribute AT in record t, μ_{FV} is the membership function of a fuzzy value FV, τ: $[\mathrm{LL}_{\mathrm{AT}},\mathrm{UL}_{\mathrm{AT}}] \rightarrow [-10,10]$ is a mapping from the interval defining AT onto [-10,10] so that we may use the same fuzzy values for different fields. A meaningful interpretation is secured by τ which makes it possible to treat all fields' domains as ranging over the unified interval [-10,10].

The elicitation (definition) of fuzzy sets corresponding to particular fuzzy values may be done using different methods. Normally, it involves an interface with the user(s) who provide responses to appropriate chosen questions.

Linguistic quantifiers provide for a flexible aggregation of simple conditions. In FQUERY for Access they are defined in Zadeh's (1983) sense, as fuzzy set on the [0, 10] instead of the original [0, 1]. They may be interpreted either using Zadeh's (1983) approach or via the OWA operators (Yager, 1988; or Yager & Kacprzyk, 1997); Zadeh's interpretation will be used here. The membership functions of fuzzy linguistic quantifiers are assumed piece-wise linear, hence two numbers from [0,10] are needed. Again, a mapping from [0,N], where N is the number of conditions aggregated, to [0,10] is employed to calculate the matching degree of a query. More precisely, the matching degree, $md(\cdot,\cdot)$, for *query* "Q of N conditions are satisfied" for record t is equal to

$$md(Q,\ \mathrm{condition}_i,\ t) = \mu_Q[\tau(\sum_i md(Q,\ \mathrm{condition}_i,\ t) \tag{4}$$

We can also assign different importances to particular conditions, and the aggregation formula is equivalent to (3). Importance is given as a fuzzy set defined on [0,1], and then treated as property B in (3) leading to

$$md(QB, \mathrm{condition}_i, t) = \mu_Q[\tau(\sum_i (md(\mathrm{condition}_i, t) \wedge \mu_B(\mathrm{condition}_i)) / / \sum_i \mu_B(\mathrm{condition}_i))] \tag{5}$$

FQUERY for Access has been designed so that fuzzy queries be syntactically correct Access's queries. This has been attained by using of parameters:

[FfA_FV *fuzzy value name*] is interpreted as a fuzzy value
[FfA_FQ *fuzzy quantifier name*] - as a fuzzy quantifier

First, a fuzzy term has to be defined and stored internally. This maintenance of dictionaries of fuzzy terms defined by users, strongly supports our approach to data summarization to be discussed next. In fact, the package has a set of predefined fuzzy terms but the user may always enrich the dictionary.

When the user initiates the execution of a query it is automatically transformed and

then run as a native query of Microsoft Access. Details can be found in (Kacprzyk & Zadrożny, 1994, 1995a-b, 2001b) and (Zadrożny & Kacprzyk, 1999).

In Kacprzyk & Zadrożny's (1998, 2001c) approach, *interactivity* is in the definition of summarizers (indication of attributes and their combinations), via a user interface of a fuzzy querying add-on. The queries (referring to summarizers) allowed are:

- *simple* as, e.g., "salary is *high*"
- *compound* as, e.g., "salary is *low* AND age is *old*"
- *compound with quantifier*, as, e.g., "*most* of {salary is *high*, age is *young*, ..., training is *well above average*}.

We will also use "natural" linguistic terms, i.e. (7±2!) like: *very low, low , medium, high, very high*, and also "comprehensible" quantifiers as: *most*, *almost all*, ..., etc.

Fuzzy queries directly correspond to summarizers in linguistic summaries. Thus, the derivation of a linguistic summary may proceed in an interactive way as follows:

- the user formulates a set of linguistic summaries of interest (relevance) using the fuzzy querying add-on described above,
- the system retrieves records from the database and calculates the validity of each summary adopted, and
- a most appropriate linguistic summary is chosen.

Fuzzy querying is very relevant because we can restate the summarization in the fuzzy querying context. First, (2) may be interpreted as:

"*Most records match query S*" (6)

where *S* replaces *F* in (2) since we refer here directly to the concept of a summarizer (this should be properly understood because *S* in (6) is in fact the whole condition, e.g., price = *high*, while *F* in (2) is just the fuzzy value, i.e. *high* in this condition; this should not lead to confusion).

Similarly, (3) may be interpreted as:

"*Most records meeting conditions B match query S*" (7)

In the database terminology, *B* corresponds to a (fuzzy) *filter* and (7) claims that *most* records passing through *B* match query *S* to a degree from [0,1].

Notice that the concept of a protoform in the sense of Zadeh (2002) is highly relevant here. First of all, a protoform is defined as an abstract prototype, that is, for the query (summary) (6) and (7), given as, respectively:

"*Most t's are S*" (8)

"*Most RBt's* are *S*" (9)

where *t* denotes "records", *R* is a filter, and *S* is a query.

Evidently, as protoforms may form a hierarchy, we can define higher level (more abstract) protoforms. For instance, replacing *most* by a general linguistic quantifier *Q*, we have:

"*Qt's* are *S*" (10)

"*QRt's* are *S*" (11)

The more abstract protoforms correspond to cases in which we assume less about summaries being sought. There are two extremes when we: (1) assume a totally abstract protoform, or (2) assume that all elements of a protoform are given on the lowest level of abstraction as specific linguistic terms. In case 1 data summarization is extremely time consuming but may produce an interesting, unexpected view on data. In case 2 the user has to guess a good candidate for a summary but the evaluation is fairly simple, equivalent to the answering of a (fuzzy) query. Thus, case 2 refers to *ad hoc queries*. This may be shown in Table 1 in

Table 1. Classification of linguistic summaries

Type	Given	Sought	Remarks
1	S	Q	Simple summaries Through ad-hoc queries
2	$S\ B$	Q	Conditional summaries Through ad-hoc queries
3	$Q\ S^{structure}$	S^{value}	Simple value oriented Summaries
4	$Q\ S^{structure}\ B$	S^{value}	Conditional value Oriented summaries
5	nothing	$S\ B\ Q$	General fuzzy rules

which 5 basic types of linguistic summaries are shown, corresponding to protoforms of a more and more abstract form.

Table 1 shows classifications where $S^{structure}$ denotes that attributes and their connection in a summary are known, while S^{value} denotes linguistic values that are sought and together with $S^{structure}$ fully define a summarizer.

Type 1 may be easily obtained by a simple extension of fuzzy querying. Basically, the user has to construct a query – candidate summary, and one has to determine the fraction of rows matching this query, and which linguistic quantifier best denotes this fraction. A Type 2 summary is a straightforward extension of Type 1 by adding a fuzzy filter. Type 3 summaries require much more effort. Their primary goal is to determine typical (exceptional) values of an attribute. So, query S consists of only one simple condition built of the attribute whose typical (exceptional) value is sought, the "=" relational operator and a placeholder for the value sought. For example, using the age-focused summary, S = "age=?" ("?" denotes a placeholder mentioned above) we look for a typical value of age. A Type 4 summary may produce typical (exceptional) values for some, possibly fuzzy, subset of rows. From the computational point of view, Type 5 summaries represent the most general form considered here: fuzzy rules describing dependencies between specific values of particular attributes. Here the use of B is essential, while previously it was optional. The summaries of Type 1 and 3 have been implemented as an extension to Kacprzyk & Zadrożny's (2000a-d) FQUERY for Access. Two approaches to Type 5 summaries have been proposed. First, a subset of such summaries may be produced taking advantage of similarities with *association rules* and using efficient algorithms for mining them. Second, a genetic algorithm may be employed to search the space of summaries. The results of the former case are briefly presented in the next section.

The protoforms are therefore a powerful conceptual tool because we can formulate many different types of linguistic summaries in a uniform way, and devise a uniform and universal way to handle different linguistic summaries. This may be viewed to confirm Zadeh's frequent claims of the power of protoforms.

IMPLEMENTATION

As a simple illustration of Type 5 summaries, an implementation is shown for the summarization of Web server logs (cf. Zadrożny and Kacprzyk, 2007).

Each request to a Web server is recorded in one or more of its log files. The recorded information usually comprises the fields listed in Table 2 (cf. a common log file format at http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format). There is also an extended format which includes more

fields but these will be of no interest for us in this paper.

There is a lot of software available (cf. e.g., Analog at http://www.analog.cx/) that reads a log file and produces various statistics concerning the usage of a given Web server. These statistics usually include the number of requests (or requested Web pages): per month/day/hour of the week, per country or domain of the requesting computer. Often the requests from specific sources (mostly search engines) are distinguished and related statistics are generated. Also the parameters of the requesting agent, such as the browser type or the operating system may be analyzed. These statistics may be computed in terms of the number of requests and/or the number of bytes of transferred data.

These simple analyses of log files refer to particular requests or requested Web pages (which may involve a number of requests for embedded multimedia files, style files etc.) More sophisticated analyses concern sessions, i.e. the series of requests send by the same agent. This type of analysis may help to model agents' behavior, identify the navigational paths and, e.g., reconstruct the Web site in order to enhance agents' experience.

It is easy to see that though those techniques, mostly involving some statistical analyses, are effective, and often powerful and efficient, their deficiency is that they are not human consistent enough. Namely, they produce numerical results that are often too voluminous and not comprehensible to an average human user who would welcome simple, intuitively appealing outcomes, possibly in a natural language. Such results are provided by linguistic summaries of data.

A Web server log file may be directly interpreted as a table of data with the columns corresponding to the fields listed in Table 2 and the rows corresponding to the requests. For instance, for the purposes of linguistic summarization attributes of requests can be as given in Table 3.

In this section we will discuss various linguistic summaries that may be derived using this data.

For efficiency, we look for a subclass of linguistic summaries that may be obtained using efficient algorithms for *association rules mining* taking into account the following correspondence between the concept of the association rule and of the linguistic summary: the condition and conclusion parts of an association rule correspond to qualifier R and summarizer S, respectively. This essentially constraints the structure of the qualifier and summarizer to a conjunction of simple conditions. However this simplification provides for the existence of efficient algorithms for rule generation. The

Table 2. Contents of a Web server log file

Field no.	Content
1	the requesting computer name or IP address
2	the username of the user triggering the request (often absent),
3	the user authentication data
4	the date and time of the request
5	the HTTP command related to the request which includes the path to the requested file
6	the status of the request (e.g., to determine whether a resource was correctly transferred, not found, etc.)
7	the number of bytes transferred as a result of the request
8	the software used to issue the request

Table 3. Attributes of the requests used for their linguistic summarization

Attribute name	Description
Domain	Internet domain extracted from the requesting computer name (if given)
Hour	hour the request arrived; extracted from the date and time of the request
Day of the month	as above
Day of the week	as above
Month	as above
Filename	name of the requested file, including the full path, extracted from the HTTP command
Extension	extension of the requested file extracted as above
Status	Status of the request
Failure	= 1 if status code is of 4xx or 5xx form, and =0 otherwise
Success	= 1 if status code is of 2xx form, and =0 otherwise
Size	number of bytes transferred as a result of the request
Agent	name of the browser used to issue the request (name for major browsers, "other" otherwise)

truth-value of the summary corresponds to the confidence measure of an association rule. Notice that we employ a restricted form of a Type 5 linguistic summary from Table 1.

The experiment was run on the log file of one of the Web servers of our institute. This is an Apache server and we used its access request log for the period of December 24, 2006 to January 16, 2007. with 352 543 requests. For extracting the data listed in Table 3, a simple Perl program was used. Then we imported the data to a Microsoft Access database and then employed our FQUERY for Access software to run the experiments by first defining a dictionary of linguistic terms. The system transformed the original access log data replacing the values of selected numerical attributes with their best matched linguistic terms (their labels or codes). Thus, in effect, a numerical attribute whose values are to be characterized in the summaries with the use of some linguistic terms is replaced with a set of binary attributes. For example, the attribute SIZE may be replaced by the artificial attributes: SIZEIsSmall, SIZEIsMedium and SIZEIsLarge. The semantics of these attributes is: an attribute, e.g., SIZEIsSmall, is said to appear in a table row (transaction, in terms of association rules mining) if the value of the original attribute, i.e., SIZE, in this row belongs to the fuzzy set representing the term *Small* to a high enough degree (which is determined by a threshold value, being a parameter controlled by the user of the FQUERY for Access interface).

The values of the remaining attributes, both numerical and textual, are also coded appropriately. Then the system executes an external program that looks for association rules. In these experiments we used an efficient implementation of the Apriori algorithm by Christian Borgelt (cf. http://www.borgelt.net/apriori.html). Finally, the obtained linguistic summaries are decoded by FQUERY for Access and presented to the user.

In our experiments the following linguistic terms were defined, among others, in the dictionary of linguistic terms.

The linguistic quantifier ``most'' is defined as in (1), i.e.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases}$$

and *small* for for the attribute *size of file* is defined by:

$$\mu_{small}(x) = \begin{cases} 1 & \text{for } x \le 50KB \\ -\frac{1}{30}x + \frac{8}{3} & \text{for } 50 \text{ KB} < x < 80KB \\ 0 & \text{for } x \ge 80KB \end{cases}$$

In our experiments we obtained a number of interesting linguistic summaries. Due to space limitations we can only show some of them. First:

All requests with the status code 304 ("not modified"') referred to *small* files' (T=1.0)

The next summary obtained concerns scalar (non-numerical) attributes:

Most files with the "gif" extension were requested from the domain "pl" (T=0.98)

and it is worth noticing it does not hold that "*Most* files were requested from the domain 'pl'" which is true to degree 0.4 only.

We obtain more convincing summaries when we add a condition concerning the status code of the request, namely:

Most files with the "gif" extension successfully fetched (with the status code 200) were requested from the domain "pl" (T=1)

Many more interesting linguistic summaries have also been obtained which can give much insight into requests coming into the Web server.

CONCLUDING REMARKS

We presented the idea and power of the concept of a linguistic data (base) summary, originated by Yager (1982) and further developed first, in a more conventional form, by Kacprzyk & Yager (2001), and Kacprzyk, Yager & Zadrożny (2000), and then, in a more general context of Zadeh's (2002) protoforms by Kacprzyk & Zadrożny (2005), and a more implementation oriented context of its relation to fuzzy database querying by Kacprzyk & Zadrożny (1998 – 2005). These summaries are some short sequence(s) in natural language that make it possible to readily capture, even by an inexperience and novice user, the essence of the very essence of data,

As an application we presented the summarization of Web server logs which is a very interesting and increasingly popular research topic in data mining. Results of such analyses can be very important by helping in the reporting, advertising, and generally decision making processes in a company. For instance, the resulting knowledge may help improve navigation paths, better organize paid search advertising, personalize Web site access, better design B2B interfaces, etc.

ACKNOWLEDGMENT

This works was partially supported by the Ministry of Science and Higher Education under the T-INFO Research Network.

REFERENCES

Kacprzyk, J. (2000). Intelligent data analysis via linguistic data summaries: a fuzzy logic approach. In R. Decker & W. Gaul (Eds.), *Classification and Information Processing at the Turn of the Millennium* (pp. 153 - 161). Berlin, Heidelberg and New York: Springer-Verlag.

Kacprzyk, J., Pasi, G., Vojtaš, P., & Zadrożny, S. (2000). Fuzzy querying: issues and perspective. *Kybernetika, 36, 605 - 616.*

Kacprzyk, J., & Yager, R.R. (2001). Linguistic summaries of data using fuzzy logic. *International Journal of General Systems, 30, 133 - 154.*

Kacprzyk, J., Yager, R.R., & Zadrożny, S. (2000). A fuzzy logic based approach to linguistic summaries of databases. *International Journal of Applied Mathematics and Computer Science, 10, 813 - 834.*

Kacprzyk, J., Yager, R.R., & Zadrożny, S. (2001). Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support. In W. Abramowicz & J. Żurada (Eds.), *Knowledge Discovery for Business Information Systems* (pp. 129-152). Boston: Kluwer.

Kacprzyk, J., & Zadrożny, S. (1994). Fuzzy querying for Microsoft Access. In *Proceedings of FUZZ-IEEE'94 (Orlando, USA): Vol. 1*, (pp. 167 – 171).

Kacprzyk, J., & Zadrożny, S. (1995a). Fuzzy queries in Microsoft Access v. 2. In *Proceedings of FUZZ-IEEE/IFES '95 (Yokohama, Japan), Workshop on Fuzzy Database Systems and Information Retrieval*, (pp. 61 – 66).

Kacprzyk, J., & Zadrożny, S. (1995b). FQUERY for Access: fuzzy querying for a Windows-based DBMS. In P. Bosc & J. Kacprzyk (Eds.), *Fuzziness in Database Management Systems* (pp. 415 – 433). Heidelberg: Physica-Verlag,.

Kacprzyk, J., & Zadrożny, S. (1998). Data mining via linguistic summaries of data: an interactive approach. In T. Yamakawa & G. Matsumoto (Eds.), *Methodologies for the Conception, Design and Application of Soft Computing - Proceedings of IIZUKA'98 (Iizuka, Japan)*, (pp. 668 – 671).

Kacprzyk, J., & Zadrożny, S. (1999) The paradigm of computing with words in intelligent database querying. In L.A. Zadeh and J. Kacprzyk (Eds.): *Computing with Words in Information/Intelligent Systems*.*(Part 2. Foundations*, (pp. 382 – 398), Heidelberg and New York: Physica-Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2000a). On combining intelligent querying and data mining using fuzzy logic concepts. In G. Bordogna & G. Pasi (Eds.), *Recent Research Issues on the Management of Fuzziness in Databases* (pp. 67 – 81), Heidelberg and New York: Physica – Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2000b). Data mining via fuzzy querying over the Internet. In O. Pons, M.A. Vila & J. Kacprzyk (Eds.), *Knowledge Management in Fuzzy Databases* (pp. 211 – 233), Heidelberg and New York: Physica – Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2000c). On a fuzzy querying and data mining interface. *Kybernetika*, *36, 657 - 670*.

Kacprzyk, J., & Zadrożny, S. (2000d). Computing with words: towards a new generation of linguistic querying and summarization of databases. In P. Sinčak & J. Vaščak (Eds.), *Quo Vadis Computational Intelligence?* (pp. 144 – 175), Heidelberg and New York: Physica-Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2001a). On linguistic approaches in flexible querying and mining of association rules. In H.L. Larsen, J. Kacprzyk, S. Zadrożny, T. Andreasen & H. Christiansen (Eds.), *Flexible Query Answering Systems. Recent Advances* (pp. 475 – 484), Heidelberg and New York: Springer-Verlag.

Kacprzyk, J., & Zadrożny, S. (2001b). Computing with words in intelligent database querying: standalone and Internet-based applications. *Information Sciences*, *34, 71 - 109*.

Kacprzyk, J., & Zadrożny, S. (2001c). Data mining via linguistic summaries of databases: an interactive approach. In L. Ding (Ed.), *A New Paradigm of Knowledge Engineering by Soft Computing* (pp. 325 – 345). Singapore: World Scientific.

Kacprzyk, J., & Zadrożny, S. (2001d). Fuzzy linguistic summaries via association rules. In A. Kandel, M. Last & H. Bunke (Eds.), *Data Mining and Computational Intelligence* ,(pp. 115 – 139), Heidelberg and New York: Physica-Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2001e). Using fuzzy querying over the Internet to browse through information resources. In B. Reusch and K.-H. Temme (Eds.), *Computational Intelligence in Theory and Practice* (pp. 235 – 262), Heidelberg and New York: Physica-Verlag (Springer-Verlag).

Kacprzyk, J., & Zadrożny, S. (2002). Protoforms of linguistic data summaries: towards more general natural - language - based data mining tools. In A. Abraham, J. Ruiz del Solar, M. Koeppen (Eds.), *Soft Computing Systems* (pp. 417 – 425), Amsterdam: IOS Press.

Kacprzyk, J., & Zadrożny, S. (2003). Linguistic summarization of data sets using association rules. In *Proceedings of FUZZ-IEEE'03 (St. Louis, USA)* (pp. 702 – 707).

Kacprzyk, J., & Zadrożny, S. (2005). Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Information Sciences, 173(4), 281-304.*

Kacprzyk, J., Zadrożny, S., & Ziółkowski, A. (1989). FQUERY III+: a 'human consistent` database que-

rying system based on fuzzy logic with linguistic quantifiers. *Information Systems*, *6, 443 - 453*.

Kacprzyk, J., & Ziółkowski, A. (1986). Database queries with fuzzy linguistic quantifiers. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC - 16, 474 - 479*.

Liu, Y., & Kerre, E.E. (1988). An overview of fuzzy quantifiers. (I) Interpretations. *Fuzzy Sets and Systems*, *95, 1 - 21*.

Yager, R.R. (1982). A new approach to the summarization of data. *Information Sciences*, *28, 69 - 86*.

Yager, R.R. (1988). On ordered weighted avaraging operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-18, 183 - 190*.

Yager, R.R. (1991). On linguistic summaries of data. In G. Piatetsky-Shapiro & B. Frawley (Eds.), *Knowledge Discovery in Databases* (pp. 347 – 363), Cambridge, USA: MIT Press.

Yager, R.R. (1995). Linguistic summaries as a tool for database discovery. In *Proceedings of FUZZ-IEEE'95/IFES'95, Workshop on Fuzzy Database Systems and Information Retrieval, (Yokohama, Japan)* (pp. 79 – 82).

Yager, R.R. (1996). Database discovery using fuzzy sets. *International Journal of Intelligent Systems, 11, 691 - 712*.

Yager, R.R., & Kacprzyk, J. (Eds.). (1997). *The Ordered Weighted Averaging Operators: Theory and Applications*. Boston: Kluwer.

Zadeh, L.A. (1983). A computational approach to fuzzy quantifiers in natural languages. *Computers and Mathematics with Applications*, *9, 149 - 184*.

Zadeh, L.A. (1985). Syllogistic reasoning in fuzzy logic and its application to usuality and reasoning with dispositions. *IEEE Transaction on Systems, Man and Cybernetics*, *SMC-15*, *754 - 763*.

Zadeh, L.A. (2002). A prototype-centered approach to adding deduction capabilities to search engines – the concept of a protoform. In *BISC Seminar, 2002, University of California, Berkeley*.

Zadeh, L.A., & Kacprzyk, J. (Eds.). (1999). *Computing with Words in Information/Intelligent Systems, 1. Foundations, 2. Applications*. Heidelberg and New York: Physica-Verlag (Springer-Verlag).

Zadrożny, S., & Kacprzyk, J. (1995) Fuzzy querying using the 'query-by-example' option in a Windows-based DBMS, *Proceedings of Third European Congress on Intelligent Techniques and Soft Computing - EUFIT'95 (Aachen, Germany), vol. 2 (pp. 733 – 736)*.

Zadrożny, S., & Kacprzyk, J. (1999). On database summarization using a fuzzy querying interface. In *Proceedings of IFSA'99 World Congress (Taipei, Taiwan R.O.C.)*, *Vol. 1, (pp. 39 – 43)*.

Zadrożny, S., & Kacprzyk, J. (2007) Summarizing the contents of Web server logs: a fuzzy linguistic approach. *Proceedings of FUZZ-IEEE'2007 – The 2007 IEEE Conference on Fuzzy Systems (London, UK, July 23-26, 2007), pp. 1860 – 1865*.

BIOS