Neighborhood Systems and Approximate Retrieval

Y.Y. Yao

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: yyao@cs.uregina.ca

An approximate retrieval model is proposed based on the notion of neighborhood systems. The knowledge used in the model consists of an information table, in which each object is represented by its values on a finite set of attributes, and neighborhood systems on attribute values, which provide semantic similarity or closeness of different values. An information table can be used for exact retrieval. With the introduction of neighborhood systems to information tables, one is able to perform approximate retrieval. Operations on neighborhood systems are introduced based on power algebras. An ordering relation representing the information of a neighborhood system is suggested and examined. Approximate retrieval is carried out by the relaxation of the original query using neighborhood systems, and the combination of intermediate results using neighborhood system operations. The final retrieval results are presented according to the proposed ordering relation. In contrast to many existing systems, a main advantage of the proposed model is that the retrieval results are a non-linear ordering of objects.

Key words: Approximate retrieval, concept hierarchies, information tables, neighborhood systems, power algebras, similarity measures.

1 Introduction

Traditional database systems with the underlying Boolean logic rely on an exact matching method for retrieving objects from a database. An object either satisfies the query, which is retrieved, or does not satisfy the query, which is not retrieved. The same is also true for the Boolean model of information retrieval [25]. With exact matching, it may happen that a query returns either too many or too few items, as the query may be too general or too specific. A user is required to have detailed knowledge about problem domain and database schema to properly construct a query [7]. The semantic relationships between attribute values are not taken into consideration in traditional

database systems. It is therefore difficult to find objects that are close enough to an object satisfying the query. Several proposals have been made to resolve some of the above mentioned problems.

Different fuzzy database models have been proposed and studied as a generalization to traditional database systems [14]. For example, fuzzy similarity relations are defined on attribute values, which are in fact quantitative measures representing closeness of attribute values. An object is retrieved if it is sufficiently similar to the one requested by a query, which is normally done through some thresholds on similarity values. Fuzzy database systems can also be designed to rank objects according to their similarities to the query. A ranked list of retrieval results is, in fact, provided by many information retrieval systems and Web search engines [2,25]. Although a linear ordering of objects is more general than an unsorted list produced by traditional database systems, it still may not provide enough flexibility to a user. The requirement of quantitative similarity measures may also be a problem. In some situations, it may not be entirely meaningful to describe similarity in quantitative terms. The interpretation of a similarity value may not necessarily be clear [38]. One may argue that usages of linear ordering and quantitative measures are more for mathematical simplicity, rather than for providing a more realistic model.

In addition to quantitative measures, Chu and Chen [7,8] used type abstraction hierarchies to describe relationships between attribute values. A special class of type abstraction hierarchy, known as concept hierarchy, has also been used by many authors in the context of data mining [11,19]. By using type abstraction hierarchies, one may relax a user query to achieve neighborhood query answering. Different retrieved sets of objects can be obtained, depending on particular relaxations of the original query. Due to the non-linear structure of type abstraction hierarchies, it is possible to arrange the retrieved objects into a non-linear ordering. Along the same line but in a different setting, Lin [17] adopted neighborhood systems from topological spaces to describe relationships between objects in database systems. Some intuitive ideas have been discussed for approximate retrieval using neighborhoods. The proposals and studies of Chu and Chen [7,8], and Lin [17] demonstrated the necessity of, and possible solutions of, approximate retrieval.

The problem of approximate retrieval is also related to the notion of nearest-neighbor queries studied in database systems in the context of geographic information systems [10]. With nearest-neighbor queries, one can find the closest object to a given object. In general, it is possible to rank a set of objects according to their closeness to a given object. The notion of neighborhood queries has been explored recently by some authors for web information retrieval. For example, Stojanovic studied the problem of query refinement by searching among neighborhood queries [29,30].

The main objectives of this paper are two folds. Based on the well established notion of neighborhood systems, an approximate retrieval model is proposed. Methods for approximate retrieval using neighborhood system are suggested and investigated. The rest of the paper is organized as follows. Section 2 overviews the basic concepts of neighborhoods and neighborhood systems. It is argued that the notion of neighborhood systems provides a convenient and flexible tool for representing similarity between attribute values. Many widely used representation methods of similarity can be explained in terms of neighborhood systems. Section 3 introduces set-theoretic operations on neighborhood systems based on the concept of power algebras. Section 4 investigates an ordering relation on objects implied by a neighborhood system. Section 5 proposes an approximate retrieval model. The task of approximate retrieval is achieved by relaxing the original query using neighborhood systems, finding a family of subsets of objects using a combination of the traditional exact matching method and neighborhood system operations, and arranging the retrieved objects according to the ordering relation implied by the retrieved family of subsets of objects.

2 Neighborhoods and Neighborhood Systems

Sierpinski introduced the concept of neighborhood systems for the study of Féchet (V)spaces [26]. It is originated from the abstraction of geometric notion of closeness. Two points in a space are *close to*, or *approximate to*, each other if one point is in a neighborhood of the other [18]. Lin [17] adopted neighborhood systems to describe relationships between objects in database systems for the purpose of approximate retrieval.

In this section, we present a brief overview of neighborhood systems defined on a finite universe, with reference to approximate retrieval. A few simple methods for constructing neighborhood systems are given. Intuitive interpretations of neighborhoods are discussed.

2.1 Basic concepts

Let U denote a finite and non-empty set called the universe. For an element $x \in U$, one may associate with it a subset $n(x) \subseteq U$ called a *neighborhood* of x. By associating a non-empty family of neighborhoods $\mathrm{NS}(x) \subseteq \mathcal{P}(U)$ to x, one obtains a *neighborhood system* of x, where $\mathcal{P}(U)$ is the power set of U. A neighborhood system may be formally interpreted as an operator from U to $\mathcal{P}(\mathcal{P}(U))$ that maps each element of U to a family of subsets of U. The collection of neighborhood systems of all elements in U, denoted by

$\mathrm{NS}(U)$, determines a Fréchet (V)space $(U, \mathrm{NS}(U))$. There are no additional requirements on neighborhood systems. A neighborhood of x may or may not contain x. If $x \in n(x)$, $n(x)$ is called a reflexive neighborhood of x. If every neighborhood in a neighborhood system is reflexive, the system is called a reflexive neighborhood system. If a neighborhood system consists of only one neighborhood, it is called a 1-neighborhood system [36]. If a neighborhood system consists of a sequence of nested neighborhoods, it is called a nested neighborhood system.

Neighborhood systems represent the information or knowledge about relationships between elements of a universe. Intuitively speaking, elements in a neighborhood of an element are somewhat close to, or similar to, that element. Elements of $n(x)$ are drawn towards x by indistinguishability, similarity, or functionality [18]. A neighborhood system $\mathrm{NS}(x)$ of x groups the universe into classes. Distinct neighborhoods of x consist of elements having different types of, or various degrees of, similarity to x. Elements in the same neighborhood $n(x)$ are regarded to be indiscernible or at least not noticeably distinguishable from x. In the present study, we focus on similarity based interpretations of neighborhood systems.

Example 1 For a universe $U = \{a, b, c, d\}$, consider the following neighborhood systems of elements of U:

$$\begin{aligned}
&\mathrm{NS}(a) = \{\{a\}, \{a, b\}\},\\
&\mathrm{NS}(b) = \{\{a, b\}, \{a, b, c\}\},\\
&\mathrm{NS}(c) = \{\{c\}\},\\
&\mathrm{NS}(d) = \{\{d\}, \{a, d\}, \{b, d\}\}.
\end{aligned}$$

All neighborhoods are reflexive. For a and d, a is in a neighborhood of d, but d is not in any neighborhood of a. This implies that neighborhood relationships is not necessarily symmetric. One may observe two types of similarity from the neighborhood system of a, one suggests that only a is similar to itself, the other suggests that b is also similar to a. One may also interpret the neighborhood system as expressing three levels of similarity, a is more similar to a than b to a, and b is more similar to a than c and d to a. This implies a linear ordering $\{a\}, \{b\}, \{c, d\}$. In contrast, such a linear ordering cannot be inferred from the neighborhood system of d. One may not say that a is more similar to d than b to d, or b is more similar to d than a to d. Furthermore, one may not say that a is as similar to d as b to d. That is, one may not compare a and b based on their similarities to d. The neighborhood system of c is a 1-neighborhood system. The neighborhood systems of a, b, and c are nested neighborhood systems.

This example shows some advantages and flexibility of neighborhood systems. In the rest of this section, we argue that the notion of neighborhood sys-

tems provides a more general and meaningful tool for describing relationships between objects of a universe. The commonly used methods, such as binary relations, fuzzy binary relations, distance functions, dissimilarity and similarity measures, and hierarchic structures, can be understood as special classes of neighborhood systems. The neighborhood systems can be applied to situations where the meaning of a distance or a similarity function is not clear. It may also be applied when only the qualitative information (for example, the ordering implied by the numerical values) is useful rather than the precise numeric values. Although there is no constraint on neighborhood systems, in real applications one may in fact use very simple neighborhood systems. For instance, a neighborhood system may only contain a few, rather than a very large number of, neighborhoods.

Neighborhood systems offer a concrete model of an emerging theory known as granular computing [1,20,23,24,37,40,45,46]. Granular computing attempts to extract the commonalities from existing fields to establish a set of generally applicable principles, to synthesize their results into an integrated whole, and to connect fragmentary studies in a unified framework. Granular computing is a multi-disciplinary study with the objectives to investigate and model a way of thinking, a family of granule-oriented problem solving methods, and a paradigm of information processing. Granular computing at philosophical level concerns structured thinking, and at the application level concerns structured problem solving. While structured thinking provides guidelines and leads naturally to structured problem solving, structured problem solving implements the philosophy of structured thinking [40].

2.2 Neighborhood systems induced by binary relations

Approximate retrieval mainly deals with the problem of finding information relevant to a query by using inexact matching, based on a semantic distance or similarity [7,8]. In addition to the exactly matched items, items that are similar to, or in a neighborhood of, such items are also retrieved. The notion of neighborhood systems provides a convenient and flexible tool for representing similarity, and can be used to describe both quantitative and qualitative information. With the neighborhood system based representation of the underlying concept of similarity or distance, one may establish a solid theoretical basis for approximate retrieval.

A simple way of describing similarity is the use of a reflexive and symmetric relation called a compatibility or tolerance relation. The reflexivity and symmetry reflect our intuitive understanding of similarity. An element should be similar to itself. If an element a is similar to another element b, then it is natural to infer that b is also similar to a. In general, one may assume that

similarity is at least reflexive, but not necessarily symmetric [27]. Different neighborhoods can be defined from a reflexive binary relation [36].

Suppose $R \subseteq U \times U$ is a binary relation on a universe U. Given two elements $x, y \in U$, if xRy, we say that x is similar to y. For each element $x \in U$, elements in the following set,

$$R_p(x) = \{y \mid yRx\}, \tag{1}$$

are similar to x, and the set is called the predecessor neighborhood of x. The corresponding neighborhood system of x consists of one neighborhood and is given by $\mathrm{NS}(x) = \{R_p(x)\}$. For a reflexive binary relation R, one obtains a reflexive neighborhood system. The family of neighborhoods $\{R_p(x) \mid x \in U\}$ of a reflexive binary relation is a covering of the universe. It is a single-layered clustering of the universe. Such a granulated view of the universe is useful in granular computing [39,44]. Each neighborhood is a granule that may be considered as one unit instead of many individuals [37,45]. In the special case where an equivalence relation is used, one obtains a partition of the universe.

In contrast to the predecessor neighborhood, x is similar to each member of its successor neighborhood defined by:

$$R_s(x) = \{y \mid xRy\}. \tag{2}$$

If R is a symmetric binary relation, then $R_p(x) = R_s(x)$. By combining predecessor and successor neighborhoods, we obtain the predecessor-and-successor neighborhood $R_{p\wedge s}(x)$ and the predecessor-or-successor neighborhood $R_{p\vee s}(x)$. An element y belongs to $R_{p\wedge s}(x)$ if both x is similar to y and y is similar to x. Obviously, we have $R_{p\wedge s}(x) \subseteq R_p(x) \subseteq R_{p\vee s}(x)$ and $R_{p\wedge s}(x) \subseteq R_s(x) \subseteq R_{p\vee s}(x)$. Neighborhood systems can be constructed from these neighborhoods.

A binary relation, predecessor neighborhoods, and successor neighborhoods uniquely determine each other [36]. One may use any one of these representations. For instance, in information retrieval an index term is associated with a thesaurus consisting of other index terms with similar meanings [25]. A thesaurus may be considered as a predecessor neighborhood of that index term. Other types of binary relations considered include neighborhoods constructed from broader-terms, narrower-terms, synonyms, and related terms [16].

The notion of neighborhood systems defined by a binary relation has been studies extensively in the context of generalized rough set theory. Wu, Mi and Zhang studied various types of generalized fuzzy rough sets based on arbitrary fuzzy binary relations [31]. Mi and Zhang gave an axiomatic characterization of fuzzy generalization of rough sets [21]. Wu and Zhang examined both constructive and axiomatic approaches of fuzzy approximation operators [33].

Yang and Li investigated minimal sets of axioms defining various types of generalized approximation operators [35].

The dichotomous interpretation of similarity is a major drawback of the binary relation based view. Two elements are viewed as being either similar or not similar without considering the degrees of similarities. For a non-transitive relation R, we may have xRy and yRz and $\neg(xRz)$. By R, x is not similar to z. On the other hand, one may still infer a certain degree of similarity from the fact that xRy and yRz. In fact, we can build a nested neighborhood systems based on a non-transitive binary relation [42]. The composition of binary relation R with itself is defined by:

$$R \circ R = \{(x, z) \mid \text{there exists a } y \in U \text{ such that } xRy \text{ and } yRz\}. \tag{3}$$

In general, we write:

$$R^2 = R \circ R, \quad R^k = R^{k-1} \circ R. \tag{4}$$

Consider the following sequence of nested binary relations used in the construction of the transitive closure of R:

$$R \subseteq R \cup R^2 \subseteq \ldots \subseteq R \cup R^2 \cup \ldots \cup R^m. \tag{5}$$

Their predecessor neighborhoods of x define a nested sequence of neighborhoods of x:

$$R_p(x) \subseteq (R \cup R^2)_p(x) \subseteq \ldots \subseteq (R \cup R^2 \cup \ldots \cup R^m)_p(x). \tag{6}$$

The sequence in turn defines a nested neighborhood system of x. Every element in the predecessor neighborhood $R_p(x)$ is similar to x. When R is a reflexive relation, we have $R \subseteq R^2$. The sequence can be simply written as $R \subseteq R^2 \subseteq \ldots \subseteq R^m$.

For an element $z \in (R \cup R^2)_p(x)$, z is either similar to x, or z is similar to another element y and y is similar to x. The same argument can be applied to the entire sequence of binary relations. One can represent a relation by a graph and define a distance function between two nodes by the number of arcs in the shortest path connecting the two nodes. The set $R_p(x)$ contains all nodes from which x can be reached in one step, $(R \cup R^2)_p(x)$ contains all nodes from which x can be reached in not more than two steps, and so on. This clearly provides us with an intuitive interpretation of nested neighborhood systems. The use of the nested sequence of binary relations implies a linear ordering on the set of elements from which x can be reached. Based on the sequence of

relations given in equation (5), Wu and Zhang studied k-step neighborhood operator systems and approximations [32].

Kortelainen presented a method for defining a binary relation on the universe based on a fuzzy set [15]. More specifically, an element is ordered ahead of another element by the binary relation, if the membership of the former is greater or equal to the membership of the latter. The neighborhoods defined by this ordering relation are related to α-cuts of the fuzzy set. In general, ordering relations provide a formal way to describe user preferences. Neighborhood systems induced by ordering relations are important in many applications.

2.3 Neighborhood systems induced by quantitative similarity measures

One may use a fuzzy binary relation, a distance function, a dissimilarity measure, or a similarity measure to describe the degrees of similarity or closeness. In fact, these quantitative measures have been widely used in information retrieval [25,34], database systems [7,8], data analysis [28], and mathematical taxonomy [13].

Consider a normalized similarity measure $s : U \times U \longrightarrow [0, 1]$ on U satisfying the condition $s(x, x) = 1$ for all $x \in U$. The similarity measure is not necessarily symmetric. For each number $\alpha \in [0, 1]$, we may define the following neighborhood of $x \in U$:

$$n_\alpha(x) = \{y \mid s(y, x) \geq \alpha\}.$$

A neighborhood system of x is given by $\mathrm{NS}(x) = \{n_\alpha(x) \mid \alpha \in [0, 1]\}$. It is a reflexive and nested neighborhood system. For any two numbers $\alpha, \beta \in [0, 1]$ with $\alpha \geq \beta$, $n_\alpha(x) \subseteq n_\beta(x)$. In this way, we may interpret quantitative knowledge expressed using a similarity measure in terms of neighborhood systems. Elements of each of the neighborhoods show a specific level of similarity to x. A similarity measure may produce too many neighborhoods. One may not be interested in such very detailed information. Specific neighborhoods may therefore be constructed by some threshold values from $[0, 1]$. By using a similar formulation, one may construct neighborhood systems from a fuzzy binary relation or a distance function.

A similarity measure suggests that any two elements can be compared. If $s(y, x) \geq s(z, x)$, then y is similar to x at least as z is similar to x, and y can be put ahead of z. One obtains a linear ordering of elements in U. As shown by Example 1, this may not necessarily reflect our intuitive understanding of similarity. It may not be entirely meaningful to require that every pair of elements are comparable. In some applications, a quantitative function may

not be readily available, as the relationships between elements of universe may be qualitative. For example, clusters of universe may also be given by an expert based on domain knowledge without explicitly referring to a quantitative function. The semantic interpretation of a quantitative function may also be a difficult problem.

2.4 Neighborhood systems induced by hierarchies

A quantitative measure can be used to derive some clusters of elements so that similar elements of U are put together. Among many clustering methods, a very important class is known as stratified clusterings, which can be further divided into hierarchic (non-overlapping) clustering methods and non-hierarchic (overlapping) clustering methods [13]. In contrast to single-layered clustering, they produce multi-layered clusters of the universe. The multi-layered granulations of the universe offer a more useful view for granular computing [39].

A precise mathematical formulation of stratified clustering can be found in a book by Jardine and Sibson [13]. Given a dissimilarity measure, the output of a hierarchic clustering process is a dendrogram, which is a tree diagram in which numerical levels are associated with the branch points. The clusters given by a particular level form a partition of the universe. By using the one-to-one correspondence between partitions and equivalence relations, one may also conveniently describe hierarchic clusters by nested equivalence relations. A similar approach was used by Lin and Hadjimichael for the interpretation of concept hierarchy [19].

A hierarchy on a universe can be conveniently described by a tree structure such that each node represents a cluster [34]. Conceptually, a hierarchy may be viewed as a successive top-down decomposition of a universe U. The root is the largest cluster consisting of all elements from U. The root is decomposed into a family of pairwise disjoint clusters. That is, the children clusters of the root form a partition of the root. Similarly, each cluster can be further divided into smaller disjoint clusters. The leaves are clusters of singleton subsets, which are equivalent to the elements of the universe U. Alternatively, a hierarchy may also be viewed as a successive bottom-up combination of some smaller clusters to form a larger cluster.

In a hierarchy, all elements of a cluster at a lower level are included in every node between that cluster and the root, which form a sequence of nested clusters. A cluster containing $x \in U$ may be regarded as a neighborhood of x. Suppose $C_1(x) \subset \ldots \subset C_h(x)$ is the sequence of clusters containing x. A neighborhood system of x is given by $\mathrm{NS}(x) = \{C_k(x) \mid 1 \leq k \leq h\}$. Neighborhood systems induced by a hierarchy may be understood as a discrete view of the

neighborhood systems defined by a distance function, in which only a finite number of neighborhoods are considered. It offers a reasonable compromise between binary relation based and distance function based neighborhood systems. In a hierarchy, one typically associates a name with a cluster such that elements of the cluster are instances of the named category or concept [13,19]. Suppose U is the domain of an attribute in a database. A hierarchic clustering of attribute values produces a concept hierarchy [11]. A concept given to a cluster in a higher level is more general than a concept given to a cluster in a lower level, while the latter is more specific than the former. The notion of concept hierarchy has been used in data mining for discovering various levels of association rules [5,11]. It should be noted that concept hierarchy may be interpreted as a special class of type abstraction hierarchy introduced by Chu and his colleagues for approximate retrieval in database systems [7,8]. In these studies, the hierarchy is given directly by experts based on domain knowledge.

One may generalize the notion of hierarchy by removing the requirement of disjoint clusters in each level [13,19]. The resultant structure may be similarly explained in terms of neighborhood systems. Given an element x, its neighborhood system is the family of clusters containing x. In general, neighborhood systems can be used to describe a more general class of knowledge.

3 Operations on Neighborhood Systems

For an element of the universe, different neighborhood systems may be defined. Each of them represents the available knowledge about the universe from a particular point of view. For example, different neighborhood systems may be supplied by a group of experts. Neighborhood systems can be combined by set-theoretic operations based on the notion of power algebras [4].

Let X be a set and $\circ$ a binary operation on X. One can define a binary operation $\circ^+$ on subsets of X as follows [4]:

$$A \circ^+ B = \{x \circ y \mid x \in A, y \in B\}, \tag{7}$$

for any $A, B \subseteq X$. In general, one may lift any operation f on elements of X to an operation f^+ on subsets of X, called the power operation of f. Suppose $f : X^n \longrightarrow X$ $(n \geq 1)$ is an n-ary operation on X. The power operation $f^+ : \mathcal{P}(X)^n \longrightarrow \mathcal{P}(X)$ is defined by [4]:

$$f^+(A_0, \ldots, A_{n-1}) = \{f(x_0, \ldots, x_{n-1}) \mid x_i \in A_i \text{ for } i = 0, \ldots, n-1\}, \tag{8}$$

for any $A_0, \ldots, A_{n-1} \subseteq X$. This provides a universal-algebraic construction

approach. For any algebra $(X, f_1, \ldots, f_k)$ with the base set X and operations $f_1, \ldots, f_k$, its power algebra is given by $(\mathcal{P}(X), f_1^+, \ldots, f_k^+)$. The power operation f^+ may carry some properties of f. For example, for a binary operation $f : X^2 \longrightarrow X$, if f is commutative and associative, f^+ is commutative and associative, respectively. If e is an identity for some operation f, the set $\{e\}$ is an identity for f^+. If a unary operation $f : X \longrightarrow X$ is an involution, i.e., $f(f(x)) = f(x)$, f^+ is also an involution. On the other hand, many properties of f are not carried over by f^+. For instance, if a binary operation f is idempotent, i.e., $f(x, x) = x$, f^+ may not be idempotent. If a binary operation g is distributive over f, g^+ may not be distributive over f^+.

A neighborhood system is a family of subsets of the universe U, i.e., it is a subset of $\mathcal{P}(U)$. By applying the idea of power algebras, we may lift set-theoretic operations on sets to neighborhood systems. For two neighborhood systems $\mathrm{NS}_1(x)$ and $\mathrm{NS}_2(x)$, the complement, intersection and union are defined by:

$$\begin{aligned}
\neg\mathrm{NS}_1(x) &= \{\sim n_{i_1}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x)\}, \\
\mathrm{NS}_1(x) \sqcap \mathrm{NS}_2(x) &= \{n_{i_1}(x) \cap n_{i_2}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x), n_{i_2}(x) \in \mathrm{NS}_2(x)\}, \\
\mathrm{NS}_1(x) \sqcup \mathrm{NS}_2(x) &= \{n_{i_1}(x) \cup n_{i_2}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x), n_{i_2}(x) \in \mathrm{NS}_2(x)\}.
\end{aligned} \tag{9}$$

They may be interpreted as extensions of set-theoretic operations in a framework of set-based computations [43]. The extended operations reduce to the standard set-theoretic operations for 1-neighborhood systems.

The neighborhood system operations satisfy the following properties:

Commutativity :
$\mathrm{NS}_1(x) \sqcap \mathrm{NS}_2(x) = \mathrm{NS}_2(x) \sqcap \mathrm{NS}_1(x)$,
$\mathrm{NS}_1(x) \sqcup \mathrm{NS}_2(x) = \mathrm{NS}_2(x) \sqcup \mathrm{NS}_1(x)$;
Associativity :
$(\mathrm{NS}_1(x) \sqcap \mathrm{NS}_2(x)) \sqcap \mathrm{NS}_3(x) = \mathrm{NS}_1(x) \sqcap (\mathrm{NS}_2(x) \sqcap \mathrm{NS}_3(x))$,
$(\mathrm{NS}_1(x) \sqcup \mathrm{NS}_2(x)) \sqcup \mathrm{NS}_3(x) = \mathrm{NS}_1(x) \sqcup (\mathrm{NS}_2(x) \sqcup \mathrm{NS}_3(x))$;
De Morgan′s law :
$\neg(\mathrm{NS}_1(x) \sqcap \mathrm{NS}_2(x)) = \neg\mathrm{NS}_1(x) \sqcup \neg\mathrm{NS}_2(x)$,
$\neg(\mathrm{NS}_1(x) \sqcup \mathrm{NS}_2(x)) = \neg\mathrm{NS}_1(x) \sqcap \neg\mathrm{NS}_2(x)$;
Double negation law :
$\neg\neg\mathrm{NS}_1(x) = \mathrm{NS}_1(x)$;
Unit element of $\sqcap$:
$\mathrm{NS}_1(x) \sqcap \{U\} = \mathrm{NS}_1(x)$;
Zero element of $\sqcup$:
$\mathrm{NS}_1(x) \sqcup \{\emptyset\} = \mathrm{NS}_1(x)$.

Operations $\sqcap$ and $\sqcup$ are not idempotent. They are not distributive over each

other. In general, $\mathrm{NS}(x) \sqcap \neg\mathrm{NS}(x)$ is not necessarily equal to $\{\emptyset\}$, and $\mathrm{NS}(x) \sqcup \neg\mathrm{NS}(x)$ is not necessarily equal to $\{U\}$. Nevertheless, $\emptyset \in \mathrm{NS}(x) \sqcap \neg\mathrm{NS}(x)$ and $U \in \mathrm{NS}(x) \sqcup \neg\mathrm{NS}(x)$. A detailed study of such a system can be found in a paper by Brink [3] on second-order Boolean algebras.

4 Orderings Induced by Neighborhood Systems

In the previous sections, we examined a specific semantic interpretation of neighborhood systems. Neighborhood systems summarize the available information about the similarity or closeness of elements. Different neighborhoods represent different types of, or various degrees of, similarity. As briefly mentioned in Example 1, this implies some elements are closer to each other than others. An ordering relation on the universe is introduced to model such a "closer to" relationships.

For a neighborhood system $\mathrm{NS}(x) = \{n_1(x), n_2(x)\}$, one may say that elements in both neighborhoods are closer to x than those in only one neighborhood. That is, elements in $n_1(x) \cap n_2(x)$ are closer to x than elements in $n_1(x) \cup n_2(x) - n_1(x) \cap n_2(x)$. When the argument is extended to an arbitrary neighborhood system, we need to introduce the notion of $\cap$-closure. For a neighborhood system $\mathrm{NS}(x)$, its $\cap$-closure $\mathrm{NS}^*(x)$ is defined to be the minimum subset of $\mathcal{P}(U)$, which contains $\mathrm{NS}(x)$ and the entire set U, and is closed under set intersection $\cap$. The $\cap$-closure of a neighborhood system is a complete lattice under the ordering given by set inclusion. The meet is given by set intersection, but the join is different from the set union. It is commonly referred to as a closure system [9]. Based on the $\cap$-closure $\mathrm{NS}^*(x)$, an ordering relation $\prec$ on U is defined as:

$$\begin{aligned} a \prec b \Longleftrightarrow \text{there exist } n_1(x), n_2(x) \in \mathrm{NS}^*(x) \text{ such that } n_1(x) \subset n_2(x), \\ a \in n_1(x), \text{ and } b \in n_2(x) - \bigcup_{n(x) \subset n_2(x)} n(x). \end{aligned} \tag{10}$$

The relation $a \prec b$ states that a is closer to x than b. Obviously, if $a \prec b$, then $b \in n(x) \Longrightarrow a \in n(x)$ for all $n(x) \in \mathrm{NS}^*(x)$. That is, if a is considered to be more similar to x than b, a must be in every neighborhood of x that contains b. However, the reverse is not necessarily true. The relation $\prec$ shows the structure imposed by a neighborhood system. More specifically, $\prec$ is asymmetric and transitive. In fact, a diagram for relation $\prec$ can be easily obtained from the Hasse diagram of the lattice $\mathrm{NS}^*(x)$. For a subset $n(x) \in \mathrm{NS}^*(x)$, one only needs to delete all elements in subsets in $\mathrm{NS}^*(x)$ proceeding $n(x)$.

In order to gain more insights into the relation $\prec$, we consider two special cases. Consider a 1-neighborhood system $\mathrm{NS}(x) = \{n(x)\}$ of element x. Suppose

$n(x) \neq U$. The $\cap$-closure is $\mathrm{NS}^*(x) = \{n(x), U\}$. The ordering relation is defined by a binary relation on U:

$$a \prec b \Longleftrightarrow a \in n(x) \text{ and } b \in U - n(x), \tag{11}$$

Intuitively, it divides U into two disjoint classes, $n(x)$ and $U - n(x)$. An element in $n(x)$ is closer to x than an element in $U - n(x)$. By extending the relation $\prec$ to subsets of U, we have:

$$n(x) \prec U - n(x), \tag{12}$$

where the extended relation $\prec$ is defined by: for two subsets $A, B \subseteq U$, $A \prec B$ if and only if $x \prec y$ for all $x \in A$ and $y \in B$. The argument can be easily applied to a nested neighborhood system $\mathrm{NS}(x) = \{n_1(x), \ldots, n_k(x)\}$ satisfying the condition:

$$n_1(x) \subset n_2(x) \subset \ldots \subset n_k(x). \tag{13}$$

Assume $n_k(x) \neq U$, we have $\mathrm{NS}^*(x) = \mathrm{NS}(x) \cup \{U\}$. In this case, the ordering relation on $\mathcal{P}(U)$ is given by:

$$n_1(x) \prec n_2(x) - n_1(x) \prec \ldots \prec n_k(x) - n_{k-1}(x) \prec U - n_k(x). \tag{14}$$

A nested neighborhood system is a chain of subsets of the universe [12]. One can also see that $a \prec b$ if and only if there exist two neighborhoods $n_1(x), n_2(x) \in \mathrm{NS}^*(x)$ with $n_1(x) \subset n_2(x)$ such that $a \in n_1(x)$ and $b \in n_2(x) - n_1(x)$. These observations provide further support for the definition and interpretation of the relation $\prec$.

Example 2 Consider a universe $U = \{a, b, c, d, e, f, g\}$ and a neighborhood system of a defined by:

$$\mathrm{NS}(a) = \{\{a, e\}, \{a, b, c\}, \{a, b, d, f\}\}.$$

The $\cap$-closure of $\mathrm{NS}(a)$ is given by:

$$\mathrm{NS}^*(a) = \{\{a\}, \{a, b\}, \{a, e\}, \{a, b, c\}, \{a, b, d, f\}, U\}.$$

The lattice $\mathrm{NS}^*(a)$ is given by:

$$\{a\} \subseteq \begin{matrix} \{a,b\} \subseteq \begin{matrix} \{a,b,c\} \\ \{a,b,d,f\} \end{matrix} \subseteq U, \\ \{a,e\} \end{matrix}$$

and the induced relation $\prec$ is:

$$\{a\} \prec \begin{matrix} \{b\} \prec \begin{matrix} \{c\} \\ \{d,f\} \end{matrix} \prec \{g\}. \\ \{e\} \end{matrix}$$

From this example, we can see that e is not comparable with b, c, d, and f regarding their closeness to a. Similarly, c is not comparable with d and f.

This example shows that the relation $\prec$ is not necessarily a linear ordering. Thus, neighborhood systems offer more flexibilities in describing relationships between elements. When finding elements similar to x, a user may examine elements by following the ordering induced by $\prec$. In contrast to a linear ordering produced by many existing retrieval systems, one is able to examine elements in various sequences.

5 An Approximate Retrieval Model

The proposed approximate retrieval model consists of two parts [41]. The first part is query relaxation by using neighborhood systems. This is similar to the approach proposed by Chu and Chen [7,8]. A type abstraction hierarchy is used in Chu and Chen's approach, while neighborhood systems are used in our formulation. The proposed method formalizes, to some extent, the informal ideas discussed by Michael and Lin [22]. The second part involves the arrangement of retrieval results. In exact matching approaches, an unsorted list of objects satisfying the query is given, which can be an empty set. Alternatively, many retrieval systems provide a linear ordering by employing a similarity measure [2,25]. The use of a type abstraction hierarchy implies that the approximate results can be arranged into a tree. Although more general than systems producing a linear ordering, it is still a special case of neighborhood systems based retrieval. Results in neighborhood system based retrieval are arranged according to the relation $\prec$.

5.1 Neighborhood system based information tables

Neighborhood system based information tables provide a simple and convenient tool for describing a finite set of objects by a finite and non-empty set of attributes. Neighborhood systems on attribute values describe the semantic closeness of attribute values. Formally, a neighborhood system based information table is defined as:

$$NT = (O, AT, \{V_a \mid a \in AT\}, \{f_a \mid a \in AT\}, \{\mathrm{NS}_a \mid a \in AT\}), \quad (15)$$

where O is a finite and non-empty set of objects, AT is a finite and non-empty set of attributes, V_a is a finite and non-empty set of values for each attribute $a \in AT$, $f_a : O \longrightarrow V_a$ is an information function for each attribute $a \in AT$, and $\mathrm{NS}_a : V_a \longrightarrow \mathcal{P}(\mathcal{P}(V_a))$ defines a neighborhood system for each value $v \in V_a$ of attribute $a \in AT$. The information function f_a associates each object with a value in V_a.

Example 3 A neighborhood system based information table can be conveniently represented by an information table [44], together with a neighborhood system defined for every attribute value [6,41]. Suppose an information table, taken from Chen [6], is given by Table 1. In this table, a group of people $O = \{o_1, \ldots, o_{12}\}$ are described by three attributes $AT = \{$Age, Gender, Opinion$\}$. The domain of Age is the integers in the closed interval $[20, 30]$. The domain of Gender is $V_{\mathrm{Gender}} = \{M, F\}$. The domain of Opinion is $V_{\mathrm{Opinion}} = \{e.n, h.n, n, s.n, m, s.p, p, h.p, e.p\}$, standing for extremely-negative, highly-negative, negative, slightly-negative, medium, slightly-positive, positive, highly-positive, and extremely-positive.

For the attribute Age, each value $20 < x < 30$ has a neighborhood system consisting of two neighborhoods:

$$\mathrm{NS}_{\mathrm{Age}}(x) = \{\{x-1, x\}, \{x, x+1\}\}.$$

For example, for 24 we have $\mathrm{NS}_{\mathrm{Age}}(24) = \{\{23, 24\}, \{24, 25\}\}$. The neighborhood systems of 20 and 30 are given by $\mathrm{NS}_{\mathrm{Age}}(20) = \{\{20, 21\}\}$ and $\mathrm{NS}_{\mathrm{Age}}(30) = \{\{29, 30\}\}$. For attribute Gender, a trivial 1-neighborhood system is used such that each element has one neighborhood consisting of itself. For example, $\mathrm{NS}_{\mathrm{Gender}}(M) = \{\{M\}\}$. For attribute Opinion, the following neighborhood systems are used:

$$\begin{aligned}
\mathrm{NS}_{\mathrm{Opinion}}(e.n) &= \{\{e.n\}, \{e.n, h.n\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(h.n) &= \{\{h.n\}, \{e.n, h.n\}, \{h.n, n\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(n) &= \{\{n\}, \{h.n, n, s.n\}\},
\end{aligned}$$

	Age	Gender	Opinion
o_1	25	M	m
o_2	27	M	n
o_3	22	F	p
o_4	30	M	$e.n$
o_5	23	F	$h.p$
o_6	20	F	$e.p$
o_7	27	M	n
o_8	24	F	$s.p$
o_9	21	F	$h.p$
o_{10}	26	M	$s.n$
o_{11}	23	M	p
o_{12}	30	M	$h.p$

Table 1
An example of information table

$$\begin{aligned}
\mathrm{NS}_{\mathrm{Opinion}}(s.n) &= \{\{s.n\}, \{n, s.n\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(m) &= \{\{m\}, \{s.n, m, s.p\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(s.p) &= \{\{s.p\}, \{p, s.p\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(p) &= \{\{p\}, \{s.p, p, h.p\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(h.p) &= \{\{h.p\}, \{h.p, e.p\}, \{p, h.p, e.p\}\},\\
\mathrm{NS}_{\mathrm{Opinion}}(e.p) &= \{\{e.p\}, \{h.p, e.p\}\}.
\end{aligned}$$

Unlike the other two attributes, neighborhood systems of values in V_{Opinion} do not have the same format. Some values have more neighborhoods than others.

5.2 Approximate retrieval

For clarity and simplicity, we only consider queries formed by the equality sign and logical connectives $\wedge$ (and) and $\vee$ (or). An atomic query is of the form, $attribute\_name = attribute\_value$. If q_1 and q_2 are two queries, both $(q_1 \wedge q_2)$ and $(q_1 \vee q_2)$ are queries. For an atomic query $q : a = v$, where $a \in AT$ and $v \in V_a$, each object of the following set:

$$R(a = v) = \{o \in O \mid f_a(o) = v\}, \tag{16}$$

satisfies the query. The set $R(q)$ is called the retrieved set of objects of query q. Let $R(q_1)$ and $R(q_2)$ be the retrieved sets of objects of queries q_1 and q_2. The retrieved sets of $q_1 \wedge q_2$ and $q_1 \vee q_2$ are given by:

$$\begin{aligned} R(q_1 \wedge q_2) &= \{o \in O \mid o \text{ satisfies } q_1 \text{ and } o \text{ satisfies } q_2\} \\ &= \{o \in O \mid o \in R(q_1) \text{ and } o \in R(q_2)\} \\ &= R(q_1) \cap R(q_2), \\ R(q_1 \vee q_2) &= \{o \in O \mid o \text{ satisfies } q_1 \text{ or } o \text{ satisfies } q_2\} \\ &= \{o \in O \mid o \in R(q_1) \text{ or } o \in R(q_2)\} \\ &= R(q_1) \cup R(q_2). \end{aligned} \tag{17}$$

Queries, represented as logical expressions, are therefore interpreted in set-theoretic terms [44]. In general, one can obtain the retrieved set of any query. For example, the retrieved set of query $q_1 \wedge (q_2 \vee q_3)$ can be obtained by $R(q_1 \wedge (q_2 \vee q_3)) = R(q_1) \cap (R(q_2) \cup R(q_3))$. By the distributive properties, $q_1 \wedge (q_2 \vee q_3)$ and $(q_1 \wedge q_2) \vee (q_1 \wedge q_3)$ are equivalent queries. They produced the same set of retrieved objects. Similarly, q, $q \wedge q$ and $q \vee q$ are equivalent queries.

An important issue of approximate retrieval based on neighborhood systems is query relaxation. Consider an atomic query $q : a = v$, where $a \in AT$ and $v \in V_a$. Suppose the value v is associated with a neighborhood system $\mathrm{NS}_a(v) = \{n_a^1(v), \ldots, n_a^{K_v}(v)\}$. For a neighborhood $n_a^i(v)$, $1 \leq i \leq K_v$, we construct a query:

$$q^i : \bigvee_{v' \in n_a^i(v)} a = v', \tag{18}$$

which can be considered as a relaxed version of q by the neighborhood $n_a^i(v)$. The retrieved set of objects by q^i is given by:

$$R(q^i) = \bigcup_{v' \in n_a^i(v)} R(a = v'). \tag{19}$$

From the neighborhood system $\mathrm{NS}_a(v)$, we have the following set of queries:

$$\mathrm{NS}(q) = \{q, q^1, \ldots, q^{K_v}\}. \tag{20}$$

It may be interpreted as a neighborhood system of q, if one extends the notion of neighborhoods to the set of all queries (logical expressions). The corresponding retrieved family of sets is given by:

$$AR(q) = \{R(q), R(q^1), \ldots, R(q^{K_v})\}. \tag{21}$$

The family $AR(q)$ may be interpreted as a neighborhood system of an element that satisfies the query q. Based on the results of atomic queries, one may define the results of any query recursively. Let q_1 and q_2 be two queries with retrieved family of sets $AR(q_1)$ and $AR(q_2)$, the retrieved families of sets of $q_1 \wedge q_2$ and $q_1 \vee q_2$ are given by:

$$\begin{aligned} AR(q_1 \wedge q_2) &= AR(q_1) \sqcap AR(q_2), \\ AR(q_1 \vee q_2) &= AR(q_1) \sqcup AR(q_2). \end{aligned} \tag{22}$$

An atomic query is interpreted by a retrieved family of sets, and logical connectives are interpreted by neighborhood systems operations. Since $\sqcap$ and $\sqcup$ are not distributive over each other, two logically equivalent expressions $q_1 \wedge (q_2 \vee q_3)$ and $(q_1 \wedge q_2) \vee (q_1 \wedge q_3)$ may produce different results. Similarly, q, $q \wedge q$, and $q \vee q$ may also produce different results.

Example 4 Suppose the neighborhood system based information table given in Example 3 is used. Consider two atomic queries q_1 : Age $= 23$ and q_2 : Opinion $= h.p$. From neighborhoods $\mathrm{NS}_{\mathrm{Age}}(23) = \{\{22, 23\}, \{23, 24\}\}$ and $\mathrm{NS}_{\mathrm{Opinion}}(h.p) = \{\{h.p\}, \{h.p, e.p\}, \{p, h.p, e.p\}\}$, we have two families of relaxed queries:

$$\begin{aligned} \mathrm{NS}(q_1) &= \{\text{Age} = 23,\ \text{Age} = 22 \vee \text{Age} = 23,\ \text{Age} = 23 \vee \text{Age} = 24\}, \\ \mathrm{NS}(q_2) &= \{\text{Opinion} = h.p,\ \text{Opinion} = h.p \vee \text{Opinion} = e.p, \\ &\qquad \text{Opinion} = p \vee \text{Opinion} = h.p \vee \text{Opinion} = e.p\}. \end{aligned}$$

Their retrieved families of sets of objects are:

$$\begin{aligned} AR(q_1) &= \{\{o_5, o_{11}\}, \{o_3, o_5, o_{11}\}, \{o_5, o_8, o_{11}\}\}, \\ AR(q_2) &= \{\{o_5, o_9, o_{12}\}, \{o_5, o_6, o_9, o_{12}\}, \{o_3, o_5, o_6, o_9, o_{11}, o_{12}\}\}. \end{aligned}$$

The first subset of objects in the family exactly satisfy the query. By using neighborhood system operations, the retrieved families of sets of queries $q_1 \wedge q_2$ and $q_1 \vee q_2$ can be computed by:

$$\begin{aligned} AR(q_1 \wedge q_2) &= AR(q_1) \sqcap AR(q_2) \\ &= \{\{o_5\}, \{o_5, o_{11}\}, \{o_3, o_5, o_{11}\}\}, \\ AR(q_1 \vee q_2) &= AR(q_1) \sqcup AR(q_2) \\ &= \{\{o_5, o_9, o_{11}, o_{12}\},\ \{o_5, o_6, o_9, o_{11}, o_{12}\}, \\ &\quad \{o_3, o_5, o_6, o_9, o_{11}, o_{12}\},\ \{o_3, o_5, o_9, o_{11}, o_{12}\}, \\ &\quad \{o_3, o_5, o_6, o_8, o_9, o_{11}, o_{12}\},\ \{o_5, o_8, o_9, o_{11}, o_{12}\}, \\ &\quad \{o_5, o_6, o_8, o_9, o_{11}, o_{12}\}\}. \end{aligned}$$

The first subset in the family consists of objects exactly satisfying the query. All other subsets represent the results of various relaxations of the original queries.

The retrieval procedure using retrieved families of sets of atomic queries describes an easily implementable formulation. Alternatively, one may lift logical connectives to extended logical connectives defined on sets of expression. For example, the lifted logical connectives on $\mathrm{NS}(q_1)$ and $\mathrm{NS}(q_2)$ can be defined by:

$$\begin{aligned}\mathrm{NS}(q_1) \wedge^+ \mathrm{NS}(q_2) &= \{r \wedge s \mid r \in \mathrm{NS}(q_1), s \in \mathrm{NS}(q_2)\},\\ \mathrm{NS}(q_1) \vee^+ \mathrm{NS}(q_2) &= \{r \vee s \mid r \in \mathrm{NS}(q_1), s \in \mathrm{NS}(q_2)\}.\end{aligned} \tag{23}$$

The retrieved families of sets of objects of these families of queries are the same as the one obtained from our earlier formulation.

From the retrieved family, $AR(q)$, of sets of objects of a query q, i.e., a neighborhood system of an object exactly satisfying the query, one may present the results according to the "closer to" relation $\prec$. This involves the calculation of the $\cap$-closure of the retrieved family $AR^*(q)$, and deletion of preceding objects in the subsets of $AR^*(q)$.

Example 5 Consider the family of retrieved sets of objects $AR(q_1 \vee q_2)$ in Example 4. The $\cap$-closure is $AR^*(q_1 \vee q_2) = AR(q_1 \vee q_2) \cup \{O\}$. The relation $\prec$ produces the following ordering of objects:

$$\{o_5, o_9, o_{11}, o_{12}\} \prec \begin{matrix}\{o_3\}\\ \{o_6\}\\ \{o_8\}\end{matrix} \prec \{o_1, o_2, o_4, o_7, o_{10}\}.$$

The set of objects $\{o_5, o_9, o_{11}, o_{12}\}$ satisfy the query, the sets of objects $\{o_3\}$, $\{o_6\}$, and $\{o_8\}$ *approximately* satisfy relaxed queries derivable from neighborhood systems, and the set of objects $\{o_1, o_2, o_4, o_7, o_{10}\}$ do not satisfy the query and the relaxed queries. Objects o_3, o_6, and o_8 cannot be compared. They are retrieved by different relaxations of atomic queries Age = 23 and Opinion = *h.p.*

Consider now another query q : Age = 24 $\vee$ Opinion = *s.p.* The family of retrieved sets is given by:

$$\begin{aligned}AR(q) = \{&\{o_8\}, \{o_1, o_8\}, \{o_3, o_8, o_{11}\}, \{o_5, o_8, o_{11}\},\\ &\{o_3, o_5, o_8, o_{11}\}, \{o_1, o_3, o_8, o_{11}\}\}.\end{aligned}$$

Its $\cap$-closure is:

$$AR^*(q) = AR(q) \cup \{\{o_8, o_{11}\}, O\}.$$

Objects can be ordered as:

$$\{o_8\} \prec \begin{matrix} & \{o_1\} & \\ \{o_{11}\} \prec & \{o_3\} & \prec \{o_2, o_4, o_6, o_7, o_9, o_{10}, o_{12}\}. \\ & \{o_5\} & \end{matrix}$$

Clearly, such an ordering provides more flexibility for a user to examine objects than a linear ordering.

As shown by the example, approximate retrieval using neighborhood systems leads to a non-linear ordering of objects. A user of such a system does not need to examine objects in a sequential manner. At each branching point, a particular relaxation of the original query is used. A user may search the entire structure by choosing different paths. This offers an advantage over many existing retrieval models. A disadvantage of the proposed model is the computational costs in both retrieval and ordering of objects. Nevertheless, the advantage of having a non-linear ordering may justify further investigations of the proposed model.

6 Conclusion

In this paper, we proposed an approximate retrieval model based on the well established mathematical notion of neighborhoods. This model not only has a solid theoretical basis, but also offers practical retrieval methods. Our discussions have been focused on the justifications, formulations, and interpretations of the proposed model. Relatively, very little attention is paid to the implementation of the model. Once the conceptual investigations are complete, one needs to examine carefully the implementation issues.

Compared with other models and systems, the proposed model has many advantages. Information systems that provide approximate retrieval may play an important role with the fast growth of available information. However, in many existing systems, retrieval results are arranged into an unsorted list or a linear list, which implies that a user must examine the retrieved results in a sequential manner. To a large extent, this may stem from the fact that an exact matching method, or a partial matching method employing a similarity measure, is used. In the computation of a partial matching, one may also

use similarity measures on attribute values. The meaning of a quantitative similarity measure on attribute values is typically provided by some intuitive arguments. It may be difficult to justify the use of a particular measure for every situation. Furthermore, the combination of similarities on attribute values to produce an overall similarity measure between query and objects may not necessarily be meaningful. The proposed model does not suffer from these problems. One associates with an attribute value with a neighborhood system. Each neighborhood describes a particular type of, or degree of, similarity. Retrieval using neighborhood systems is based on a combination of different neighborhood systems. The results are therefore not necessarily a linear ordering of objects.

The idea of providing a non-linear ordering of object is very simple and intuitively appealing, which provides a user with more flexibility and may be an accurate modeling of reality. Non-linear ordering of objects may have a significant impact on the design and implementation of more useful information retrieval systems. It appears that the notion of neighborhood systems may offer a potential solution to approximate retrieval.

Acknowledgment

The author is grateful to the anonymous referees for their constructive comments.

References

[1] A. Bargiela, W. Pedrycz, Granular Computing: an Introduction, Kluwer Academic Publishers, Boston, 2002.

[2] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks 30 (1998) 107-117.

[3] C. Brink, Second-order Boolean algebras, Quaestiones Mathematicae 7 (1984) 93-100.

[4] C. Brink, Power structures, Algebra Universalis 30 (1993) 177-216.

[5] M. Chen, J. Han, P.S. Yu, Data mining, an overview from a databases perspective, IEEE Transactions on Knowledge and Data Engineering 8 (1996) 866-883.

[6] X. Chen, Neighborhood-based Information Systems, M.Sc. Thesis, Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada, P7B 5E1, 1997.

[7] W.W. Chu, Q. Chen, Neighborhood associative query answering, Journal of Intelligent Information Systems 1 (1992) 355-382.

[8] W.W. Chu, Q. Chen, A structured approach for cooperative query answering, IEEE Transactions on Knowledge and Data Engineering 6 (1994) 738-749.

[9] P.M. Cohn, Universal Algebra, Harper and Row Publishers, New York, 1965.

[10] H. Garcia-Molina, J.D. Ullman, J. Widom, Database Systems: the Complete Book, Prentice Hall, New Jersey, 2002.

[11] J.W. Han, Y. Cai, N. Cercone, Data-driven discovery of quantitative rules in relational databases, IEEE Transactions on Knowledge and Data Engineering 5 (1993) 29-40.

[12] T.B. Iwinski, Ordinal information systems I, Bulletin of the Polish Academy of Sciences, Mathematics 36 (1988) 467-475.

[13] N. Jardine, R. Sibson, Mathematical Taxonomy, Wiley, New York, 1971.

[14] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Theory and Applications, Prentice Hall, New Jersey, 1995.

[15] J. Kortelainen, Applying modifiers to knowledge acquisition, Information Sciences 134 (2001) 39-51.

[16] J.H. Lee, M.H. Kim, Y.J. Lee, Ranking document in thesaurus-based Boolean retrieval systems, Information Processing and Management 30 (1994) 79-91.

[17] T.Y. Lin, Neighborhood systems and approximation in relational databases and knowledge bases, Proceedings of the 4th International Symposium on Methodologies of Intelligent Systems, 1988.

[18] T.Y. Lin, Granular computing on binary relations I: data mining and neighborhood systems, in: L. Polkowski and A. Skowron (Eds.), Rough Sets in Knowledge Discovery 1, Methodology and Applications, Physica-Verlag, Heidelberg, 1998, pp. 286-318.

[19] T.Y. Lin, M. Hadjimichael, Non-classificatory generation in data mining, Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, 1996, pp. 404-411.

[20] T.Y. Lin, Y.Y. Yao, L.A. Zadeh (Eds.), Data Mining, Rough Sets and Granular Computing, Physica-Verlag, Heidelberg, 2002.

[21] J.S. Mi, W.X. Zhang, An axiomatic characterization of a fuzzy generalization of rough sets, Information Sciences 160 (2004) 235-249.

[22] J.B. Michael, T.Y. Lin, Neighborhoods, rough sets, and query relaxation in cooperative answering, in: T.Y. Lin and N. Cercone (Eds.), Rough Sets and Data Mining: Analysis of Imprecise Data, Kluwer Academic Publishers, Boston, 1997, pp. 229-238.

[23] W. Pedrycz, Granular computing: an introduction, Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001, pp. 1349-1354.

[24] W. Pedrycz (Ed.), Granular Computing: an Emerging Paradigm, Physica-Verlag, Heidelberg, 2001.

[25] G. Salton, M.H. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[26] W. Sierpinski (translated by C.C. Krieger), General Topology, University of Toronto, Toronto, 1961.

[27] R. Slowinski, D. Vanderpooten, Similarity relation as a basis for rough approximations, in: P.P. Wang (Ed.), Advances in Machine Intelligence & Soft-Computing, Department of Electrical Engineering, Duke University, Durham, North Carolina, 1997, 17-33.

[28] J. Stepaniuk, Approximation spaces, reducts and representatives, in: L. Polkowski and A. Skowron (Eds.), Rough Sets in Knowledge Discovery 2, Applications, Case Studies and Software Systems, Physica-Verlag, Heidelberg, 1998, pp. 109-126.

[29] N. Stojanovic, Information-need driven query refinement, Web Intelligence and Agent Systems 3 (2005) 155-169.

[30] N. Stojanovic, On the role of user's knowledge gap in an information retrieval process, Proceedings of the 3rd International Conference on Knowledge Capture, 2005, 83-90.

[31] W.Z. Wu, J.S. Mi, W.X. Zhang, Generalized fuzzy rough sets, Information Sciences 151 (2003) 263-282.

[32] W.Z. Wu, W.X. Zhang, Neighborhood operator systems and approximations, Information Sciences 144 (2002) 201-217.

[33] W.Z. Wu, W.X. Zhang, Constructive and axiomatic approaches of fuzzy approximation operators, Information Sciences 159 (2004) 233-254.

[34] C.J. Van Rijsbergen, Information Retrieval, Butterworth, London, 1979.

[35] X.P. Yang, T.J. Li, The minimization of axiom sets characterizing generalized approximation operators, Information Sciences 176 (2006) 877-899.

[36] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Information Sciences 111 (1998) 239-259.

[37] Y.Y. Yao, Granular computing using neighborhood systems, in: R. Roy, T. Furuhashi, and P.K. Chawdhry (Eds.), Advances in Soft Computing - Engineering Design and Manufacturing, Springer-Verlag, London, 1999, pp. 539-553.

[38] Y.Y. Yao, Qualitative similarity, in: Y. Suzuki, S. Ovaska, T. Furuhashi, R. Roy, and Y. Dote (Eds), Soft Computing in Industrial Applications, Springer-Verlag, London, 2000, pp. 339-348.

[39] Y.Y. Yao, Information granulation and rough set approximation, International Journal of Intelligent Systems 16 (2001) 87-104.

[40] Y.Y. Yao, Perspectives of granular computing, Proceedings of 2005 IEEE International Conference on granular computing, Vol. 1, 2005, pp. 85-90.

[41] Y.Y. Yao, X.C. Chen, Neighborhood based information systems, Proceedings of the 3rd Joint Conference on Information Sciences, Volume 3: Rough Set & Computer Science, 1999, pp. 154-157.

[42] Y.Y. Yao, T.Y. Lin, Graded rough set approximations based on nested neighborhood systems, Proceedings of 5th European Congress on Intelligent Techniques & Soft Computing, 1997, pp. 196-200.

[43] Y.Y. Yao, N. Noroozi, A unified model for set-based computations, Soft Computing: 3rd International Workshop on Rough Sets and Soft Computing, 1994, pp. 252-255.

[44] Y.Y. Yao, N. Zhong, Granular computing using information tables, in: T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (Eds.), Data Mining, Rough Sets and Granular Computing, Physica-Verlag, Heidelberg, 2002, pp. 102-124.

[45] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 19 (1997) 111-127.

[46] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) - an outline, Information Sciences 172 (2005) 1-40.