An Algebraic Approach to Information Retrieval Systems

Zbigniew W. Raś

Received January 1980; revised May 1982

This paper deals with Salton's systems of information retrieval. In his model, only the correspondence between requests and objects (documents or document descriptions) is described. The request language is treated as a partially ordered set which may have a lattice structure or even Boolean algebra structure. Those structures have influence in simplicity or retrieval methods and in mathematical pictoriality of some important properties of information systems like simplicity or selectivity. This paper investigates such problems and presents a method of designing artificial attributes and their values.

KEY WORDS: Information retrieval, artificial attributes, selective systems, homomorphism.

1. INTRODUCTION

Constructing a mathematical model of an information retrieval system we have to consider a minimum of two sets of items: a request language on the one hand and a document set or a set of document descriptions on the other. Clearly the particular model depends on many factors. Mostly, in this paper, we are going to deal with the structure of a request language. Namely, let us observe that request identifiers may be defined by some relations or on the contrary each request identifier may be independent of any other. For example, let us assume that description identifiers correspond to the values of attributes. Of course the most natural way is to take now as a set of request identifiers our previous set of description identifiers. Attributes define relations in an obvious way. Another situation arrives when key words, for example, are taken as request identifiers. In that case, it may happen that we

1 University of North Carolina, Department of Mathematics, Charlotte, North Carolina 28223 and the University of Warsaw, Institute of Math., 00901 Warsaw, Poland.
have no natural relations between them. Obviously retrieval methods depend on the structure of a request language and algorithms appropriate for one model may not be useful for another. Having more compact structures of request languages we may profit by better methods for constructing retrieval algorithms and by the same we are able to build faster algorithms.

In order to be more precise, let us take the Pawlak's model of an information retrieval which is one with attributes. Namely by Pawlak's system we understand a sequences $S = (X, A, I, U)$, where:

1. $X$ is a set called the set of objects (documents or document descriptions)
2. $A$ is an arbitrary set called the set of descriptors
3. $I$ is a set of attributes and it describes the partition of $A$, namely $A = \bigcup_{i \in I} A_i$. Elements of $A_i$ are interpreted as values of the attribute $i \in I$.
4. $U$: $A \rightarrow 2^X$ is a classification function which satisfies two conditions: $\bigcup_{a \in A_i} U(a) = X$, $U(a) \cap U(b) = \emptyset$ for any $a, b \in A_i$ and $i \in I$.

Now we are going to define a request language $L = (A, R)$ where $A$ is an alphabet and $R$ is a set of requests. We assume that the alphabet $A$ contains the following items:

1. $a$ (request identifiers, where $a \in A$)
2. $+, \cdot$ (two-argument functors)
3. $-$ (one argument functor)
4. $(, )$ (auxiliary signs)

The set $R$ of requests is defined as a least set satisfying the conditions:

1. $a \in R$
2. if $t, s \in R$ then $(t \cdot s)$, $(t + s)$, $-t \in R$.

By an elementary term of the level $n$ we mean a term of the form $a_{i_1} \cdot a_{i_2} \cdot \ldots \cdot a_{i_n}$ where $a_{i_j} \in A_{i_j}$ and $n \neq m$ for $n \neq m$.

Assume now that $n$ is fixed and for each term $a_{i_1} \cdot a_{i_2} \cdot \ldots \cdot a_{i_n}$ of the level $n$ we have a file containing all objects belonging to the set $U(a_{i_1}) \cap U(a_{i_2}) \cap \cdots \cap U(a_{i_n})$. In order to find the response for a given request $r \in R$ we have to represent $r$ by means of terms from the $n$ level. Let us note that having the restrictions assumed by Pawlak for the classification function $U$ we are able to do that because of the following properties: $U(t) \cap U(s) = U(t \cdot s)$, $U(t + s) = U(t) \cup U(s)$ for any terms $t, s \in T$ and $U(a) = \bigcup_{b \in A_i, b \neq a} \sum b$ for any $a \in A_i$, $i \in I$. 
