The Efficient Computation of Complete and Concise Substring Scales with Suffix Trees

Sébastien Ferré

Irisa/Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France
ferre@irisa.fr

Abstract. Strings are an important part of most real application multi-valued contexts. Their conceptual treatment requires the definition of *substring scales*, i.e., sets of relevant substrings, so as to form informative concepts. However these scales are either defined by hand, or derived in a context-unaware manner (e.g., all words occuring in string values). We present an efficient algorithm based on suffix trees that produces complete and concise substring scales. Completeness ensures that every possible concept is formed, like when considering the scale of all substrings. Conciseness ensures the number of scale attributes (substrings) is less than the cumulated size of all string values. This algorithm is integrated in Camelis, and illustrated on the set of all ICCS paper titles.

1 Introduction

In information systems, one of the most common datatype is the *string*. For instance, in a bibliographic application, most attributes are string-valued (author names, title, journal or conference name). While these strings usually bring a lot of information, they are hardly exploited in conceptual information systems based on Formal Concept Analysis (FCA) [GW99]. They are most often represented as (1) nominal values, which is right for entry types (e.g., "journal", "inproceedings") but uninteresting for titles, (2) a set of keywords given by hand [CS00], or (3) a set of keywords derived in a context-unaware manner, e.g., all title words [FR01].

An important objective of conceptual information systems is to ensure a tight combination of querying and navigation [GMA93]. In this respect, the manual or context-unaware production of keywords is unsatisfactory because they are fully part of the navigation structure, and so should be automatically derived from the context, like the concept lattice. We consider in this paper the automatic derivation of *substring scales*, whose values are full strings (like titles), and whose attributes are substrings (corresponding to keywords). For instance, in the case of the bibliographic context of all ICCS papers, one would expect to have substrings like "Formal Concept Analysis", "Conceptual Graphs". These substrings play the same role as inequalities and intervals over numeric values (*ordinal* and *interordinal* scales [GW99]), or general terms in taxonomies.

S.O. Kuznetsov and S. Schmidt (Eds.): ICFCA 2007, LNAI 4390, pp. 98–113, 2007.

A substring scale should be *complete* in the sense that every possible concept is derived from the scaled context, like when considering all substrings. A substring scale should also be *concise* enough so as not to overwhelm users during navigation, and be computed *efficiently*.

In Section 2, we present a naive conceptual scaling, and show that it does not satisfy conciseness and efficiency. In Section 3, we introduce a new solution, and show with the help of suffix trees that it has good properties w.r.t. completeness, conciseness and efficiency. Section 4 describes an algorithm for computing a complete substring scale from a set of string values. This algorithm is incremental, and so supports context updates, as required in information systems. It has been integrated into Camelis, an implementation of Logical Information Systems (LIS) [FR04], and Section 5 shows its application to a bibliographic context of ICCS paper titles, how many domain keywords are clearly identified, and how they naturally form a taxonomy. This paper ends with a discussion about other datatypes (Section 6), and a conclusion (Section 7).

2 Naive Approach

Suppose we have n objects, each object being decribed by a string over an alphabet Σ. This forms a *string context*.

Definition 1 (string context). *A* string context *is a triple $D = (\mathcal{O}, \Sigma^*, d)$, where $\mathcal{O}$ is a finite set of objects, Σ^* is the domain of strings over a finite alphabet Σ, and d is a mapping from objects to Σ-strings: for every object $o \in \mathcal{O}$, $d(o) \in \Sigma^*$ is the description of the object by a string.*

A string context can be seen as a multivalued context with only one attribute, $d(o)$ being the value of this attribute for the object o. All results in this paper also apply to contexts with several attributes, but it is not necessary to consider them explicitly here as each attribute can be treated in isolation.

Example 1. *The following table shows a basic string context that serves as an example in the following.*

o	$d(o)$
1	*abc*
2	*dab*
3	*ac*
4	*dab*

The cover of a substring is the set of objects whose description contains it in a string context. This is equivalent to the definition of extent in logical concept analysis [FR04], where formulas would be strings and substrings.

Definition 2 (cover). *Let $D = (\mathcal{O}, \Sigma^*, d)$ be a string context. The* cover *of a string $s \in \Sigma^*$ in D is defined by (where $\supseteq$ denotes the containment relation between strings and substrings)*

$$cover_D(s) = \{o \in \mathcal{O} \mid d(o) \supseteq s\}.$$