New Indices for Text:

PAT trees and PAT arrays

Gaston H. Gonnet
Dept. of Computer Science
ETH, Zurich
Switzerland

Ricardo A. Baeza-Yates
Depto. de Ciencias de la Computación
Universidad de Chile
Casilla 2777,
Santiago, Chile

Tim Snider
Centre for the New OED and Text Research
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

Abstract

We survey new indices for text, with emphasis on PAT arrays (also called suffix arrays). A PAT array is an index based on a new model of text which does not use the concept of word and does not need to know the structure of the text.

1 Introduction

Text searching methods may be classified as lexicographical indices (indices that are sorted), clustering techniques, and indices based on hashing. In this chapter we discuss two new lexicographical indices for text, called PAT trees and PAT arrays. Our aim is to build an index for the text of size similar to or smaller than the text.

Briefly, the traditional model of text used in information retrieval is that of a *set of documents*. Each document is assigned a list of *keywords* (attributes), with optional relevance *weights* associated

to each keyword. This model is oriented to library applications, which it serves quite well. For more general applications it has some problems, namely:

- A basic structure is assumed (documents and words). This may be reasonable for many applications, but not for others.
- Keywords must be extracted from the text (this is called "indexing"). This task is not trivial and error prone, whether it is done by a person, or automatically by a computer.
- Queries are restricted to keywords.

For some indices, instead of indexing a set of keywords, all words except for those deemed to be too common (called *stopwords*) are indexed.

We prefer a different model. We see the text as one long *string*. Each position in the text corresponds to a semi-infinite string (*sistring*), the string that starts at that position and extends arbitrarily far to the right, or to the end of the text. It is not difficult to see that any two strings not at the same position are different. The main advantages of this model are:

- No structure of the text is needed, although if there is one, it can be used.
- No keywords are used. The queries are based on *prefixes* of sistrings, that is, on any substring of the text.

This model is simpler and does not restrict the query domain. Furthermore, almost any searching structure can be used to support this view of text.

In the traditional text model each document is considered a database record, and each keyword a value or a secondary-key. Because the number of keywords is variable, common database techniques are not useful in this context. Typical database queries are on equality or on ranges. They seldom consider "approximate text searching".

This paper describes PAT trees and PAT arrays. PAT arrays are an efficient implementation of PAT trees, and support a query language more powerful than do traditional structures based on keywords and boolean operations. PAT arrays were independently discovered by Gonnet (1987) and Manber and Myers (1990). Gonnet used them for the implementation of a fast text searching system, PAT™ (see Gonnet (1987) and Fawcett (1989)), used with the *Oxford English Dictionary*