Efficient Lazy Algorithms for Minimal-Interval Semantics

Paolo Boldi and Sebastiano Vigna
Dipartimento di Scienze dell’Informazione, Università degli Studi di Milano

Abstract. Minimal-interval semantics associates with each query over a document a set of intervals, called witnesses, that are incomparable with respect to inclusion (i.e., they form an antichain): witnesses define the minimal regions of the document satisfying the query. Minimal-interval semantics makes it easy to define and compute several sophisticated proximity operators, provides snippets for user presentation, and can be used to rank documents: thus, computing efficiently the antichains obtained by operations such as logic conjunction and disjunction is a basic issue. In this paper we provide the first algorithms for computing such operators that are linear in the number of intervals and logarithmic in the number of input antichains. The space used is linear in the number of antichains. Moreover, the algorithms are lazy—they do not assume random access to the input antichains. These properties make the usage of our algorithms feasible in large-scale web search engines.

1 Introduction

Search engines are a popular way to retrieve information in the web. However, the classical problem studied by the theory of information retrieval, that of answering a query by returning the set of documents that match the information provided by the user, is complicated by the huge number of documents to be taken into consideration. On the web retrieving many relevant documents is usually not a problem—the documents are simply too many already. Precision, rather than recall (in particular, precision in the first 10–20 results) is the main issue.

A first possibility for extending the user capabilities is query expansion, an automatic or semi-automatic mechanism that tries to enrich a given query, by using for example some semantics extracted from the context, or by asking directly the user what is the intended meaning of his/her query.

A different approach is that of departing from the Boolean model, and provide the user with more powerful (but understandable) operators. In this paper we concentrate on minimal-interval semantics, a semantic model that uses antichains of intervals of natural numbers to represent the semantics of a query. Each interval is a witness of the satisfiability of the query, and defines a region of the document that the query satisfies (words in the document are numbered starting from 0, so regions of text are identified with intervals of integers). For instance, a query formed by the conjunction of two terms is satisfied by the minimal intervals of the document containing both terms: minimality is guaranteed...
by the fact that in an antichain every pair of elements is incomparable with respect to inclusion.

This approach has been defined and studied in full extent by Clarke, Cormack and Burkowski in their seminal paper [3]. They showed that antichains have a natural lattice structure that can be used to interpret conjunctions and disjunctions in queries. Moreover, it is possible to define several additional operators (proximity, followed-by, and so on) directly on the antichains. The authors have also described families of successful ranking schemes based on the number and length of the intervals involved.

The main feature of minimal-interval semantics is that, by its very definition, an antichain of intervals cannot contain more than \( n \) intervals, where \( n \) is the number of words in the document. Thus, it is in principle possible to compute all minimal-interval operators in time linear in the document size. This is not true, for instance, if we consider different interval-semantics approaches in which all intervals are retained and indexed (e.g., the PAT system [5] or the sgrep tool [6]), as the overall number of regions is quadratic in the document size.

In this paper, we attack the problem of providing efficient algorithms for the computation of such operators. As a subproblem, we can compute the proximity of a set of terms, and indeed we are partly inspired by previous work on proximity [11,9]. Our algorithms are linear in the number of input intervals. For conjunction and disjunction, there is also a multiplicative logarithmic factor in the number of input antichains, which however can be shown to be essentially unavoidable in the disjunctive case. The space used by all algorithms is linear in the number of input antichains (in fact, we need to store just one interval per antichain). Moreover, our algorithms are lazy, that is, while building their results they do not advance the input lists more than necessary.

Previously, the only attempt at linear lazy algorithms for minimal-interval region algebras we are aware of is the work of Young–Lai and Tompa on structure selection queries [13], a special type of expressions built on the primitives “contained-in”, “overlaps”, and so on, that can be evaluated lazily in linear time. Their motivations are similar to ours — application of region algebras to very large text collections. Similarly, Navarro and Baeza–Yates [8] propose a class of algorithms that using tree-traversals are able to compute efficiently several operations on overlapping regions. Their motivations are efficient implementation of structured query languages that permit such regions. Indeed, some of the techniques used therein (e.g., double stacks) are similar to our double indirect priority queues. Albeit similar in spirit, they do not provide algorithms for any of the operators we consider.

We believe that the existence of (almost) linear lazy algorithms for minimal-interval semantics makes it the natural candidate for advancing web search engines beyond a purely Boolean model: in particular, the possibility of limiting the interval width has a very natural interpretation for the user in terms of proximity, and ordered conjunction has obvious applications (e.g., searching for a verse in a song when some word is missing).