Succinct Data Structures in Information Retrieval: Theory and Practice

Simon Gog
Institute of Theoretical Informatics, Karlsruhe Institute of Technology
Am Fasanengarten 5
76131 Karlsruhe, Germany
gog@kit.edu

Rossano Venturini
Department of Computer Science, University of Pisa
Largo Bruno Pontecorvo 3
I-56127 Pisa, Italy
rossano.venturini@uniipi.it

ABSTRACT

Succinct data structures are used today in many information retrieval applications, e.g., posting lists representation, language model representation, indexing (social) graphs, query auto-completion, document retrieval and indexing dictionary of strings, just to mention the most recent ones. These new kind of data structures mimic the operations of their classical counterparts within a comparable time complexity but require much less space. With the availability of several libraries for basic succinct structures – like SDSL, Succinct, Facebook’s Folly, and Sux – it is relatively easy to directly profit from advances in this field.

In this tutorial we will introduce this field of research by presenting the most important succinct data structures to represent set of integers, set of points, trees, graphs and strings together with their most important applications to Information Retrieval problems. The introduction of the succinct data structures will be sustained with a practical session with programming handouts to solve. This will allow the attendees to directly experiment with implementations of these solutions on real datasets and understand the potential benefits they can bring on their own projects.

1. MOTIVATION

The current growth and availability of massive amounts of data gathered and processed by applications – Web search engines, textual and biological databases, just to cite a few – has changed the algorithmic requirements of basic processing and mining tools and provide ample motivation for a great deal of new theoretical and practical research on algorithms and data structures.

Not surprisingly, the last decades have seen a massive research in the field of the so-called succinct and compressed data structures. These new kind of data structures mimic the operations of their classical counterparts within a comparable time complexity but requiring much less space. These solutions usually resort to a careful combination of ideas born both in algorithmic and data compression fields. As a concrete example consider trees which are probably among the most used data structure in practice. Everybody is familiar with the classical way to represent the structure of a tree in memory: each node is connected to its children by means of pointers. This representation immediately allows simple navigational operations on the tree. One can also augment the structure with auxiliary information in order to answer queries such as: Who is the parent of a given node? How many nodes are in a given subtree? Who is the lowest common ancestor of a given pair of nodes? and so on. The main drawback of this solution is its space occupancy which can be very optimistically estimated in 32 bits per node. Not surprisingly, the problem of representing trees space efficiently has been one of the first problems to be addressed. The current best succinct tree representation requires provably no more than 2 bits per node and is able to answer in constant time a large set of queries. The practical consequence of the use of this succinct data structure is immediate: applications can store in main memory trees whose sizes would be prohibitive with the classical representation.

These efficient tree representations are just an example of succinct data structures and, even if the research field is relatively new, a large fraction of classical data structures have their succinct counterparts. Most of these results are of practical interest and several implementations exist, for example we mention SDSL\(^1\), Succinct\(^2\), Facebook’s Folly\(^3\) and Sux\(^4\).

Several authors are successfully applying the ideas at the basis of the succinct data structures to solve problems in other fields of research. Information retrieval community profits a lot from these data structures as there exist several applications in which they play a central role, e.g., posting lists representation, language model representation, indexing (social) graphs, query auto-completion, document retrieval and indexing dictionary of strings, just to mention the most recent ones.

The tutorial will introduce this field of research by presenting the most important succinct data structures to represent set of integers, set of points, trees, graphs and strings together with their most important applications to Information Retrieval problems. The attendees to directly experiment with implementations exist. For example we mention SDSL\(^1\), Succinct\(^2\), Facebook’s Folly\(^3\) and Sux\(^4\).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR ’16, July 17-21, 2016, Pisa, Italy
© 2016 ACM. ISBN 978-1-4503-4069-4/16/07...$15.00
DOI http://dx.doi.org/10.1145/2911451.2914802

\(^1\)https://github.com/simongor/sdsl-lite
\(^2\)https://github.com/ot/succinct
\(^3\)https://github.com/facebook/folly
\(^4\)http://sux.di.unimi.it
2. OBJECTIVES AND RELEVANCE TO THE IR COMMUNITY

Indexing is the key to efficient query processing in the vast majority of IR applications. The use of inverted indexes in web search engines is probably one of the most prominent examples in Information Retrieval.

While index compression is a relatively old and established method in IR and covered by all relevant text books the new data structures which combine compression and efficient execution of operations – like random access – are a relatively new field of which many IR researchers are not yet aware of. However, this field of research is highly relevant for IR as, for example, we have already seen that techniques from the succinct data structure field are able to improve inverted indexes systematically [20, 35]. Succinct data structures often extend the functionality of traditional structures or reduce query time. For example, while inverted indexes are restricted in handling phrase queries, new document retrieval frameworks based on succinct structures allow to efficiently answer top-\(k\) queries for phrases of arbitrary length [19], and it is also possible to determine the document frequency of any phrase in constant time by just spending at most 2 bits per element in the collection [32]. Unfortunately, succinct structures per se are not easy to implement from scratch and especially the implementation of the latter two results is very complex. However, as more and more implementations of basic succinct structures become available via libraries, it is possible to combine these pieces to obtain more advanced solutions. While other fields like Bioinformatics already actively profit from using succinct structures after engineering the theoretical proposals to their applications, e.g., the FM-Index is used in DNA read mapping, the IR community does not yet fully exploit the new possibilities of succinct structures. With this tutorial we do not only inform the audience about succinct data structures but also point out engineering challenges and encourage the participants to use the potential of these solutions in IR. Therefore the objectives of this tutorials are (1) to introduce the audience to the world of succinct data structures, (2) to teach the basic techniques of succinct structures, (3) to present the practical impact of the structures to selected applications in IR, and (4) to show the participants how they can implement their own succinct structures on top of existing basic structures provided by the succinct data structure library.

3. DETAILED SCHEDULE

The tutorial is subdivided in two parts. The goal of the first part is to describe the most important succinct data structures. We will present the theoretical achievements in this field of research by focusing our attention to the ones that have demonstrated their relevance in practice. The introduction of each of these solutions will be sustained by presenting a selection of Information Retrieval problems in which they have an immediate application.

The goal of the second part is to present ready-to-use implementations of these data structures. This is done by introducing the Succinct Data Structure Library (SDSL). This library covers all the data structures presented in the previous part and has one of the presenters as its creator and one of its main developers. The speakers will also prepare a set of programming handouts which will allow the attendees to experiment with the learned concepts. The ultimate objective of the tutorial is to demonstrate to the attendees the potential benefit that succinct data structures can bring in their applications at a tiny implementation cost.

Schedule of the tutorial.

- **Part I**
  - Introduction to succinct data structures;
  - Basic operations on binary vectors [25, 31];
  - Elias-Fano representation [1, 5, 8, 27, 28, 29, 35];
  - Succinct representations of trees and range minimum/maximum queries [8, 10, 12, 33];
  - Wavelet trees [2, 13, 17, 24];
  - Compressed Full-Text indexes [7, 23];
  - Structures for IR [9, 11, 11, 20, 21, 24].

- **Part II**
  - Overview about SDSL components and concepts;
  - Detailed presentation of components via small code examples;
  - Exploration of practical performance of structures;
  - Presentation of benchmarking facilities;
  - Programming handouts.

4. ACKNOWLEDGEMENT

This tutorial was partially supported by the EU H2020 Program under the scheme INFRAIA-1-2014-2015: Research Infrastructures grant agreement #654024 SoBigData: Social Mining & Big Data Ecosystem.

References


