Succinct Data Structures for NLP-at-Scale

Matthias Petri and **Trevor Cohn**
Computing and Information Systems
The University of Melbourne, Australia
first.last@unimelb.edu.au

1 Motivation

Succinct data structures involve the use of novel data structures, compression technologies, and other mechanisms to allow data to be stored in extremely small memory or disk footprints, while still allowing for efficient access to the underlying data. They have successfully been applied in areas such as Information Retrieval and Bioinformatics to create highly compressible in-memory search indexes which provide efficient search functionality over datasets which traditionally could only be processed using external memory data structures.

Modern technologies in this space are not well known within the NLP community, but have the potential to revolutionise NLP, particularly the application to 'big data' in the form of terabyte and larger corpora. This tutorial will present a practical introduction to the most important succinct data structures, tools, and applications with the intent of providing the researchers with a jump-start into this domain. The focus of this tutorial will be efficient text processing utilising space efficient representations of suffix arrays, suffix trees and searchable integer compression schemes with specific applications of succinct data structures to common NLP tasks such as n-gram language modelling.

2 Tutorial Outline

The half-day tutorial contains several sessions, each with a practical component, on the following topics:

Introduction and Motivation (15 Minutes) In the first session we motivate the need for succinct structures. We discuss existing applications and examples in IR and NLP, and benefits of succinct structures when facing terabyte scale text processing problems.

Basic Technologies (20 Minutes) In the second session we give a brief introduction into the theoretical foundation of succinct data structures. We discuss the concept of bitvectors, data compression and basic RANK and SELECT on bitvectors which are at the core of most succinct data structures. Next we introduce several simple succinct structures such as a succinct tree representation (LOUDS) and variable size, constant access time, integer representations (DAC).

Index based Pattern Matching (20 Minutes) In the next session we start the tutorial by briefly revisiting several classical search indexes such as Suffix Arrays and Suffix Trees which can provide search functionality used in many NLP applications. We then introduce the notation of a Compressed Suffix Array (CSA), which substantially reduces the space requirement of a Suffix Array while providing the same functionality.

Compressed Suffix Arrays Internals: How does it all work (30 Minutes) This session covers the underlying technical principals of compressed suffix arrays. This includes concepts such as the Burrows-Wheeler Transform, backward search and wavelet trees.

Compressed Suffix Trees (CST): Concepts (10 Minutes) Here we extend the concept of compressed suffix arrays to compressed suffix trees (CST). This includes additional structures to represent the shape of the tree as well as additional operations supported by the CST which are important for several NLP applications.

Utilizing CSA and CST in the context of Language Models (20 Minutes) This session will cover our line of research on unlimited order language modelling with modified Kneser-Ney smoothing. Efficient methods for storing and querying language models are critical for scaling to large corpora and high Markov orders. At its core, our approach uses a compressed suffix tree of text which provides near optimal compression while supporting efficient search. We present algorithms for on-the-fly computation of probabilities under a Kneser-Ney and Modified Kneser-Ney language models. Our approach is exact and on par with leading language modelling toolkits in terms of speed while being several order of magnitudes smaller.

Other applications of Succinct Structures in NLP (10 Minutes) The last session briefly touch on advanced succinct structures consisting of Range Minimum / Maximum Queries, Succinct Tries, Treaps and 2D-range query structures.

3 Instructors

Matthias Petri is an expert in the field of practical succinct data structures and one of the main authors of the quasi-standard succinct data structure library (SDSL) used by many researchers in the field. He has several publications in top journals and conferences in algorithms, algorithm engineering and information retrieval (e.g., WWW, DCC, ALENEX, SIGIR, SPE, CPM). He specializes in algorithms engineering and applying succinct data structures to large scale text indexing problems and is currently employed as a Research Fellow working on text indexing and data compression at the University of Melbourne.

Trevor Cohn is an expert in statistical and probabilistic approaches to natural language processing, with a focus on models and algorithms for structured prediction. He publishes in the top journals and conferences in language processing and machine learning (ACL, EMNLP, AAAI). His work includes applications to parsing, translation, information extraction, summarisation and the modelling of social media. He is currently a Senior Lecturer and ARC Future Fellow at the University of Melbourne.

Acknowledgments

This tutorial was supported by the Australian Research Council (FT130101105) and a Google Faculty Research Award.

References

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, pages 390–398.

Simon Gog, Timo Beller, Alistair Moffat, and Matthias Petri. 2014. From theory to practice: Plug and play with succinct data structures. In *Proceedings of the International Symposium on Experimental Algorithms*, pages 326–337.

Enno Ohlebusch, Johannes Fischer, and Simon Gog. 2010. CST++. In *Proceedings of the International Symposium on String Processing and Information Retrieval*, pages 322–333.

Ehsan Shareghi, Matthias Petri, Gholamreza Haffari, and Trevor Cohn. 2016. Fast, small and exact: Infinite-order language modelling with compressed suffix trees. *Transactions of the Association for Computational Linguistics*, 4:477–490.

Peter Weiner. 1973. Linear pattern matching algorithms. In *Proceedings of the Annual Symposium Switching and Automata Theory*, pages 1–11.