doi:10.1093/comjnl/bxr136

Capsule Reviews

Fairouz Kamareddine

The Capsule Reviews are intended to provide a short succinct review of each paper in the issue in order to bring it to a wider readership. The Capsule Reviews were compiled by Fairouz Kamareddine. Professor Kamareddine is an Associate Editor of *The Computer Journal* and is based in the Department of Mathematical and Computer Sciences at Heriot-Watt University, Edinburgh, UK.

Fast Pattern-Matching via *k*-bit Filtering Based Text Decomposition. M. Oÿguzhan Kulekci, Jeffrey Scott Vitter and Bojian Xu
Currently, text is stored on digital media as it would be written on paper. The authors of this paper explore another way of storing text files that can boost search performance. The authors do not wish the size of the converted file to exceed the size of the original file. The format to which files are converted is called k-bit-filtered format. The main idea is to move the most informative bits of the file to the beginning, so they can be used as filters. The remaining bits are then concatenated back preserving their original order. Information theory can be used to find the indices of the most informative k-bits among the bytes of a file as can other approaches which include using compressed sizes as indicators. The authors discuss how to obtain a decent time conversion of the ordinary file into k-bit-filtered format. Then, the authors explain how to search for patterns on a k-bit-filtered format. Given a pattern P and the set R of k indices, the search process decomposes the pattern into the pattern filter (PF) and the pattern play load (PL). PF is searched on the filter part of the k-bit filtered file. Next, preprocessing for possible alignments of PL and the matching on the filter part are given where a mask matrix is constructed and the search operation algorithm is sketched and followed by a verification algorithm. Tests are conducted on a variety of texts (natural language, plain ASCII, etc.) and improvements on search speed are discussed.

Minimizing the Range for *k*-Covered Paths on Sensor Networks. Manuel Abellanas, Antonio Leslie Bajuelos and Inês Matos
Formally, the distance between a point q on the plane and a set S of points is defined as the minimum distance between q and any one point of S. The point q is said to be covered by a set S of sensors with the sensing range r if the distance between q and S is $\leq r$. Furthermore, q is said to be k-covered by S if it is covered by at least k sensors of S. One way to evaluate the quality of the coverage provided by a particular sensor network is to find minimal- (i.e. the worst covered path) and maximal-exposure paths and to maximize the minimal-exposure path (since larger sensing ranges provide better coverage). However, larger ranges result in higher costs and shorten the sensors' battery life and hence it is important to develop strategies that optimize the coverage of a given region without compromising the network's lifespan. This paper aims to extend the life of a sensor network by minimizing the sensors' range in order to assure the existence of a k-covered path between two points on a given region. The focus is on maximal-exposure paths within three types of regions: a planar graph, the whole plane and a polygonal region. Minimizing of the sensing range for k-coverage is detailed for each such region type (path on a planar graph, path on the plane, a polygonal region, and path on a polygonal region). Algorithms to calculate the minimum sensing range have been provided to allow the existence of a maximal exposure path and to output the subset of sensors needed to k-cover such a path.

Orange4WS Environment for Service-Oriented Data Mining. Vid Podpecan, Monika Zemenova and Nada Lavrac
This paper argues that data mining is difficult for non-expert users and that a formal capture of knowledge discovery tasks needs to be used to improve repeatability of experiments and to enable reasoning on the results to facilitate their reuse. The authors' goal was to develop a simple, user-friendly software platform that is able to integrate web services and local components in terms of workflow composition, originating from different communities including also a knowledge discovery ontology to support the automatization of workflow construction. To achieve this, the authors propose a novel Service-oriented Knowledge Discovery (SoKD) framework, and its implementation named Orange4WS (Orange for Web Services) which upgrades the existing data mining system Orange into a new SoKD platform. The work is aimed at supporting human experts in scientific discovery tasks. The authors claim that while each individual extension of the existing data mining technologies is not scientifically ground-breaking, the developed Orange4WS environment as a whole is a radically new data mining environment from many

perspectives. First, the authors present a motivating use case for developing and using a service-oriented knowledge discovery platform, including a user-friendly workflow editor. The goal of this use case is to produce a compact and understandable graph of terms, which could potentially give insights into relations between biological, medical and chemical terms, relevant to the subject of a user-defined query. Then, the authors describe the structure and design of the proposed ORANGE4WS software platform. To enrich the proposed knowledge discovery platform with semantics, the authors developed the Knowledge Discovery (KD) ontology which defines relationships among the declarative and algorithmic components of knowledge discovery scenarios. The three core concepts of the ontology are: knowledge, algorithm, and KD task. A planning algorithm is used to generate abstract workflows automatically where a hierarchy of algorithms based on defined classes and input/output specifications is computed, and in searching for neighbouring states the planner exploits the algorithm hierarchy. Such annotations and planning have been integrated into the Orange4WS platform. Three use cases from different domains (illustrating the availability of WEKA algorithms, relational data mining, and complex real-life systems biology) are presented to illustrate some of the capabilities of the Orange4WS implementation.

Science, Mathematics, Computer Science, Software Engineering. DICK HAMLET

This paper discusses the relationship between the four disciplines: Science, Mathematics, Computer Science (CS), and Software Engineering (SE). It concludes among other things that unlike Mathematics, both CS and SE are science-free. The author discusses the concepts of Science, Mathematics, Engineering and the traditional paradigm which states that Engineering must conform to its underlying Science and uses the Science's mathematics to establish this conformation for each particular design. The author argues that CS and SE do not fit this traditional paradigm and states that attempting to apply the mathematics/science/engineering paradigm to software, science with its theories and natural laws is missing. The author argues that there can be no software natural laws because the whole thing is an arbitrary human invention and programs have no presence in the world apart from the mathematical descriptions that come from theoretical CS. The author goes further to state that the usual experimental computer scientist has more in common with a mechanic trying to repair an engine than with experimental science and that the CS landscape is littered with dead theories that were never refuted but are not used. The author discusses the computing literature which asserts that CS is a Science and visits the opinions of Denning, Milner, Hoare, Hartmanis and others. The discussion on the science versus SE is as interesting as the earlier discussion on the science versus CS. According to the author, when an experiment fails, the software engineer sets about debugging and either the formal requirements or the program (or both) are adjusted to bring them into line and perhaps give a proof of correctness. Here, the author argues, practice can be changed to fit theory and mistaking theoretical SE for science can be detrimental to SE research. The author concludes that as long as CS/SE are seen as a science, their theories and methods will be evaluated using inappropriate standards leading to missed opportunities.

Boosting Text Compression with Word-Based Statistical Encoding. ANTONIO FARIÑA, GONZALO NAVARRO and JOSÉ R. PARAMÁ

Traditionally, classical compressors used characters as the symbols to be compressed and their compression performance was poor. An improvement to character-based compressors include the dictionary-based algorithms (in particular the LZMA algorithm) which replace text substrings by previous occurrences and improve compression ratios but at the cost of slower compression and decompression. Numerous other compressors (character-based, text-based and/or integrated with text indexing) have been introduced and are reviewed at the start of this paper. The authors then concentrate on word-based compressors and state that although the word-based detour has been justified by the interest in achieving fast compression/decompression and direct searching on the compressed text, while maintaining a reasonably competitive compression ratio, those compressors can play an even more influential role. The authors show that those compressors are compression boosters for the best classical methods, both in compression time and compression ratio and they also boost the time of sequentially searching for patterns in the compressed text and the performance of compressed self-indexes both in the compression ratio and the search performance. After a discussion of the related work including the preprocessing for compression and for self-indexing, and after introducing the necessary concepts needed for the paper, the authors analyze preprocessing using dense codes and then move to the main results which include boosting compression and boosting self-indexing. Experiments are carried out to show the advantages of the proposed method for boosting compression, self-indexing and online search.