2015 Imperial College Computing Student Workshop

ICCSW'15, September 24–25, 2015, London, United Kingdom

Edited by

Claudia Schulz

Daniel Liew

Editors

Claudia Schulz
Department of Computing
180 Queen's Gate, London, SW7 2AZ
United Kingdom
claudia.schulz@imperial.ac.uk

Daniel Liew
Department of Computing
180 Queen's Gate, London SW7 2AZ
United Kingdom
daniel.liew@imperial.ac.uk

ACM Classification 1998
A.1 Introductory and Survey, C.1 Processor Architectures, C.1.2 Multiple Data Stream Architectures (Multiprocessors), D.1 Programming Techniques, D.1.3 Concurrent Programming, D.2 Software Engineering, D.2.4 Software/Program Verification, D.3 Programming Languages, D.3.4 Run-Time Environments, F.2 Analysis of Algorithms and Problem Complexity, F.2.2 Computations on discrete structures, F.4 Mathematical Logic and Formal Languages, F.4.1 Mathematical Logic, H.3 Information Storage and Retrieval, H.3.3 Information Search and Retrieval, I.2 Artificial Intelligence, I.2.1 Applications and Expert Systems, I.2.3 Answer/reason extraction, I.2.4 Knowledge Representation Formalisms and Methods, I.2.6 Learning, I.2.7 Natural Language Processing, I.3 Computer Graphics, I.3.1 Hardware Architecture, I.4 Image Processing and Computer Vision, I.4.5 Reconstruction, J.3 Life and Medical Sciences, K.3 Computers and Education, K.3.1 Computer Uses in Education

ISBN 978-3-95977-000-2

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-95977-000-2.

Publication date
September, 2015

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/OASIcs.ICCSW.2015.i

ISBN 978-3-95977-000-2 **ISSN 2190-6807** **http://www.dagstuhl.de/oasics**

OASIcs – OpenAccess Series in Informatics

OASIcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 2190-6807

www.dagstuhl.de/oasics

Contents

Keynotes

Regular Papers

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

Welcome to the 2015 Imperial College Computing Student Workshop (ICCSW'15), the fifth workshop in its series. ICCSW was initiated with a "by students, for students" spirit: a workshop organised solely by students to give student speakers the opportunity to present their work. The organising students gain the valuable experience of what is involved in conference organisation, including writing a call for sponsors, taking part in the reviewing process, and chairing a session. On the other hand, the participating students benefit from the interaction with international researchers who are at a similar stage in their career and developing skills in presenting their research to a non-specialist computer science audience. This year's Imperial College Computing Student Workshop was organised by the Imperial College ACM Student Chapter closely following the "by students, for students" spirit. As in previous years, a unique reviewing model was adopted, where in addition to members of the Imperial College ACM Student Chapter all authors were involved in the reviewing process. Furthermore, we reached out to students at a variety of European universities to act as "ambassadors" for ICCSW'15 by spreading the word about the workshop.

This volume contains the papers accepted for presentation at the 2015 Imperial College Computing Student Workshop. ICCSW'15 received 18 submissions from 10 different countries across two different tracks: the traditional paper track including technical and tool papers, and a survey track. After the thorough reviewing process and discussion by members of the Imperial College ACM Student Chapter 11 papers were accepted, representing a 61% acceptance rate which is in line with previous years.

In addition to the regular presentations of accepted papers, we are pleased to welcome two prominent keynote speakers to ICCSW'15:

- "Google and Open Source" by Chris DiBona, Director of Open Source at Google
- "Going Beyond Fact-based Question Answering" by Erik Mueller, researcher at MIT and member of the IBM team that developed the Watson Jeopardy! system.

On behalf of the organising committee, we wish to thank all authors, accepted or not, as well as our ambassadors, who acted as reviewers in our unique peer review process and promoted ICCSW'15 respectively. Furthermore, we also wish to thank our sponsors: Google, for their platinum-level sponsorship, who have supported ICCSW since its inception in 2011; and ARM, for their bronze-level sponsorship. Without the support of our sponsors, ICCSW'15 would not have been possible. Our thanks also goes to Imperial College London, and in particular the Department of Computing, who provided us with invaluable support in organising ICCSW'15.

Claudia Schulz and Daniel Liew
ICCSW'15 Editors

Conference Organisation

Organising Committee

Kristijonas Čyras	Imperial College London
Casper da Costa-Luis	Imperial College London
Martin Ingram	Imperial College London
Bertan Kavuncu	Imperial College London
Daniel Liew	Imperial College London
Luo Mai	Imperial College London
Pedro Martinez Mediano	Imperial College London
Feryal Mehraban Pour Behbahani	Imperial College London
Kyriacos Nikiforou	Imperial College London
Claudia Schulz	Imperial College London
Doniyor Ulmasov	Imperial College London
Silvia Vinyes Mora	Imperial College London
Jacek Zienkiewicz	Imperial College London
	Imperial College London ACM Student Chapter `http://acm.doc.ic.ac.uk/`

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Ambassadors

Sami Alajrami	Newcastle University
Abdallah Arioua	University of Montpellier
Shaswar Baban	Kings College London
Gabriel Barata	University of Lisbon
Evgenij Belikov	Heriot-Watt University
Tarek Besold	University of Osnabrück
Guido Boella	University of Turin
Benoit Desouter	Ghent University
Emmanuelle Dietz	Dresden University of Technology
Stefan Ellmauthaler	Leipzig University
Jorge Fandiño	University of A Coruña
Manoel Franca	City University London
Virginia Grande	Uppsala University
Carlos A.S.J. Gulo	University of the State of Mato Grosso
Evgenios Hadjisoteriou	University of Cyprus
Haruna Isah	University of Bradford
Mathilde Janier	University of Dundee
Amin Mobasheri	Heidelberg University
Tiago Oliveira	University of Minho
Sandro Pinto	University of Minho
Thomas Schön	Uppsala University
Zohreh Shams	University of Bath
Christof Spanring	University of Liverpool & Vienna University of Technology
Bas van Gijzel	University of Nottingham
Agnieszka Zbrzezny	Częstochowa University of Technology

External Reviewers

Evgenij Belikov	Heriot-Watt University
Álvaro Câmara	University of Porto
Khaled El-Abbasy	University of Westminster
Jorge Fandiño	University of A Coruña
Virginia Grande	Uppsala University
Evgenios Hadjisoteriou	University of Cyprus
Haruna Isah	University of Bradford
Carlos A.S.J. Gulo	University of the State of Mato Grosso
Olfat Mirza	University of Warwick
Tiago Oliveira	University of Minho
Hasmik Osipyan	National Polytechnic University of Armenia
Simone Prado	São Paulo State University
Thiago Rúbio	Federal Center for Technological Education of Minas Gerais
F.Sharmila Satthar	University of Brighton
Julius Schöning	University of Osnabrück
António Silva	University of Minho
Maciej Skorski	University of Warsaw
Christof Spanring	University of Liverpool
Shazia Tabassum	University of Porto
Agnieszka Zbrzezny	Częstochowa University of Technology
Jian Zhang	University of Dundee

Supporters and Sponsors

Supporting Scientific Institutions

Imperial College London
http://www.imperial.ac.uk/

Platinum Sponsors

Google Inc.
http://www.google.com/

Bronze Sponsors

ARM Holdings, plc.
http://www.arm.com/

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Google and Open Source

Chris DiBona

Google Inc., Mountain View, CA, US

Abstract

In this talk Chris DiBona will review Google's use of open source projects and the history of prominent releases like Android, Chromium, Angular.js and some 3500 other projects (though not all of them will be surveyed!). Keeping such releases on track and efficient and, in some cases, retiring them has been his focus since he started at the company. He'll review the various ways Google supports the worldwide community of software developers that it has derived so much value from. Specifically for the students assembled, Mr. DiBona will also talk about the university oriented program "The Summer of Code" which is designed to lure students into open source projects and provide for them the real world mentorship they need to become open source committers themselves. A paid internship that lasts approximately 3 months during the summer months, The Summer of Code has introduced over 10,000 developers in 123 countries to open source software development and added over 30 million lines of code to open source projects that Google and the students use every day of their lives.

1998 ACM Subject Classification D.2 Software Engineering

Keywords and phrases Open Source

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.1

Category Keynote Talk

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 1–1

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Going Beyond Fact-based Question Answering

Erik T. Mueller

MIT, Cambridge, MA, US

Abstract

To solve the AI problem, we need to develop systems that go beyond answering fact-based questions. Watson has been hugely successful at answering fact-based questions, but to solve hard AI tasks like passing science tests and understanding narratives, we need to go beyond simple facts. In this talk, I discuss how the systems I have most recently worked on have approached this problem. Watson for Healthcare answers Doctor's Dilemma medical competition questions, and WatsonPaths answers medical test preparation questions. These systems have achieved some success, but there is still a lot more to be done. Based on my experiences working on these systems, I discuss what I think the priorities should be going forward. First, to deal with the richness of human knowledge, we need to move beyond propositional logic to predicate logic. Second, to deal with the real world, we need to represent and reason about events and time. Third, to find multiple solutions and keep them distinct from one another, we need to use declarative problem solving methods like answer set programming. As one example of a formalism that embodies these three things, I review the event calculus described in my book Commonsense Reasoning. This formalism is especially useful for the narrative understanding task.

1998 ACM Subject Classification I.2.1 Applications and Expert Systems, I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases Commonsense Reasoning

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.2

Category Keynote Talk

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 2–2

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Language Run-time Systems: an Overview

Evgenij Belikov

Heriot-Watt University, School of Mathematical and Computer Sciences
Riccarton, EH14 4AS, Edinburgh, Scotland, UK
eb120@hw.ac.uk

Abstract

The proliferation of high-level programming languages with advanced language features and the need for portability across increasingly heterogeneous and hierarchical architectures require a sophisticated run-time system to manage program execution and available resources. Additional benefits include isolated execution of untrusted code and the potential for dynamic optimisation, among others. This paper provides a high-level overview of language run-time systems with a focus on execution models, support for concurrency and parallelism, memory management, and communication, whilst briefly mentioning synchronisation, monitoring, and adaptive policy control. Two alternative approaches to run-time system design are presented and several challenges for future research are outlined. References to both seminal and recent work are provided.

1998 ACM Subject Classification A.1 Introductory and Survey, D.3.4 Run-time Environments

Keywords and phrases Run-time Systems, Virtual Machines, Adaptive Policy Control

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.3

1 Introduction

Programming languages have evolved from assembly languages, where programmers have to specify a sequence of low-level instructions for a specific target platform, towards higher levels of expressiveness and abstraction. Imperative languages enabled *structured programming* by simplifying the expression of nested conditionals, iteration, and by allowing the grouping of instructions to named procedures. More recently, object-oriented and declarative languages with sophisticated type systems and support for generics, polymorphism, concurrency, and parallelism, among other advanced features, further improved *portability* and *productivity*.

It is widely known that given a set of M programming languages and N target platforms, using a *platform-independent intermediate representation*, which abstracts over the instruction set of the target architecture, reduces the number of necessary translation components from $M*N$ to $M+N$ facilitating the implementation of and interoperability between programming languages across a wide range of diverse hardware architectures [37, 34].

As the advanced language features, such as automatic memory management, generics and reflection, require support beyond the static capabilities of a compiler, a *run-time system* (RTS, also referred to as *high-level language virtual machine* – a process-oriented sub-class of virtual machines [31, 11, 20, 41]) was introduced to dynamically manage program execution whilst maintaining high performance, which program interpreters sacrifice in most cases [38]. The added flexibility of delayed specialisation and additional context information available at run time allow for dynamic optimisation for many applications with no hard real-time requirements, e.g. by detecting program hot spots and recompiling relevant program fragments just-in-time (JIT) [4, 2]. Moreover, recent trends towards increasingly hierarchical and heterogeneous parallel architectures [35, 9] and the promise of Big Data Analysis [30] have

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 3–12

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

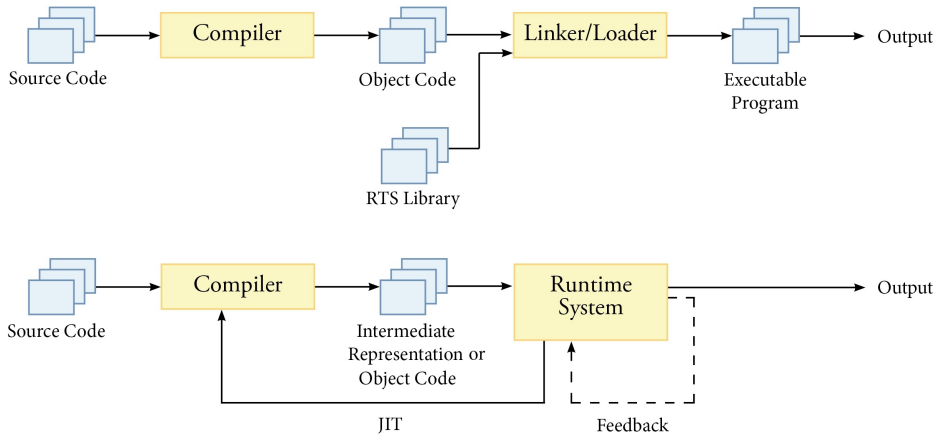


Figure 1 RTS in the Software Life Cycle.

reinvigorated the interest in language support for concurrency, parallelism, and Distributed Computing, based on advanced compiler and RTS-level functionalities [7].

This paper provides an overview of a set of representative language run-time systems including both established systems and research prototypes, presents the key components of a generic RTS and discusses associated policies as well as two more radical alternative RTS designs. Finally, challenges and opportunities for future research are outlined. While the discussion remains rather brief and high-level, references to technical literature are provided.

2 The Architecture of a Generic Language Run-time System

Here, a *run-time system* is defined as *a software component responsible for dynamic management of program execution and of the resources granted by the OS*. A RTS is distinct from and embedded into a more broad *run-time environment*, which is defined to include the OS that is responsible for overall resource management, and RTS instances managing other running applications unaware of each other's existence and resource demands. In other words, an RTS mediates interaction between the application and the OS whilst managing language-specific components in user space to avoid some context switching overhead [39].

2.1 Run-time System in the Software Life Cycle

The term *software life cycle* describes the stages of the software development process and their relations. Here the focus is on the compilation and execution stages. The RTS is primarily used during the application run time, consisting of the startup, initialisation, execution and termination phases, in addition to potential dynamic linking, loading and JIT-compilation as well as ahead-of-time (AOT) compilation.

Figure 1 illustrates the possible roles of the RTS in the software life cycle (excluding design, construction, and maintenance). For instance, C's minimalistic RTS (`crts0`) is added by the linker/loader and initialises the execution environment (e.g. the stack, stack/heap pointers, registers, signal handlers), transfers control to the program by calling the `main` function, and eventually passes on `main`'s return value to the run-time environment (e.g. the shell). However, it is platform-specific and implemented in assembler so that recompilation is required on different architectures, thus the *abstract machine* functionality is not provided.

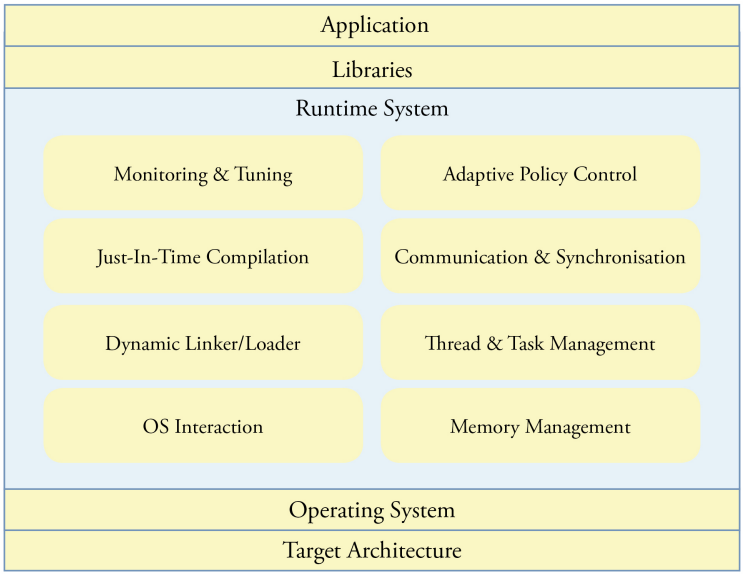


Figure 2 RTS Components in the Software Stack.

More sophisticated RTSes are either implemented as a separate library and linked to the application object code to create an executable program (e.g. GUM RTS for Glasgow parallel Haskell [42]) or are themselves applications which receive program code as an input (e.g. Java Virtual Machine (JVM) [26], and .NET Common Language Runtime (CLR) [33]). The input is usually in an *intermediate representation form* (IR, virtual instruction set) that decouples symbolic programming language representation from the native instruction set of the target architecture, which is then either interpreted or JIT-compiled to native code. Ahead-of-time compilation can be used to compile IR to native code to avoid initially running the interpreted IR code until the JIT phase is complete to reduce start-up time.

2.2 Run-time System Structure and Function

We focus on the general components corresponding to key RTS functions and associated *policies*, i.e. *sets of rules that define choices in the behaviour of the system*. We omit the discussion of RTSes for embedded platforms as strict real-time requirements, often associated with embedded applications, preclude the use of an RTS. The main function of the RTS is the management of program execution on the implemented abstract machine based on a specific execution model. The policies include interaction with the OS, memory management, thread and task management, communication, and monitoring, among others.

Execution Models. A common model is that of a *stack machine* corresponding closely to the Von-Neumann architecture [18], where a stack is used to store function arguments and intermediate results returned from function calls. This sequential model is often employed in RTSes for imperative and object-oriented languages along with *register machines*, where a stack is replaced by a set of virtual registers, which are mapped to the machine's registers.

Another prominent model is *graph reduction* commonly used with functional languages based on the lambda calculus [22, 5]. This model is suitable for parallelism as parts of the graph can be reduced independently [17]. Another notable model is *term-rewriting* that is used

with languages based on predicate logic [23], such as Prolog [43], where facts are represented by terms and new facts can be deduced from available facts. This model implements a variant of tree search and has proven particularly suitable for Artificial Intelligence applications.

OS Interaction. Commonly, an RTS relies on the OS for resource allocation, I/O and networking and provides a layer hiding OS-specific details and offering an *architecture-independent* API to systems programmers. RTS often directly implements higher-level language features and lower-level language primitives that are difficult to implement efficiently in the language itself. Hiding low-level details from the programmers is beneficial for productivity by providing abstractions and by transparently handling boilerplate initialisation, clean-up and error checking. For instance, channels in Erlang [1] offer a higher level communication mechanism than sockets [39] and in some cases communication and synchronisation are *fully implicit*, as in the distributed GUM RTS for parallel Haskell.

Error and Exception Handling. Error handling is often fairly crude such as a global `errno` variable that can be explicitly checked along with the error code returned by the function. More advanced language features include custom signal handlers and exception handling using the `try/catch` mechanism which is explicit and often results in complex control flow. Sophisticated RTSes can transparently handle errors at run time as demonstrated by Erlang's supervision mechanism, where failed processes can be restarted and resume execution.

Memory Management. Most high-level languages require the RTS to support *garbage collection* (GC) to automatically manage allocation and reclamation of heap objects. There are several common GC schemes [24]. One mechanism is *reference-counting* where a counter indicates the number of references to an object which can be recovered once this count drops to zero. By contrast, *mark-and-sweep* GC walks through the heap and marks all objects to be collected which is more disruptive but is able to collect reference cycles. Moreover, GC can be *compacting* to avoid fragmentation by periodically grouping the used objects together, and *generational* where objects reachable for a long time are promoted to a less often GC'd heap region, based on the insight that long surviving objects are likely to survive longer and most objects expire after a short period of time. A potential scalability issue on multicores is the use of a *stop-the-world* GC which suspends all execution when GC is performed, suggesting the use of private heaps across cores to enable *concurrent* or *distributed* GC.

Thread and Task Management. OS-level threads and RTS-level tasks (or *lightweight threads*) are at the heart of support for concurrency and parallelism, which is a key source of application performance on modern hardware. Often the RTS manages a thread pool and a task pool multiplexing tasks onto a set of OS threads for scalability. Many languages provide explicit threaded programming model, however, this model is deemed a poor choice due to observable non-determinism and notoriously difficult to detect and correct *race conditions* and *deadlocks* [25]. Thus higher-level deterministic programming models such as Transactional Memory [19] and semi-explicit or skeleton-based models have gained increasing attention [7].

Furthermore, explicit *synchronisation* involves a granularity trade-off that is detrimental to portable performance: a global lock would be safe but sacrifices parallelism, whereas fine-grained locking may result in prohibitive overheads. Coordination of execution, including scheduling and work distribution across heterogeneous architectures substantially increases the complexity of RTS decisions. This is the reason why *optional* tasks or threads are preferable to *mandatory* ones: the RTS only executes a task in parallel if it appears worthwhile [42, 13].

Two main families of work distribution mechanisms are *work pushing* (or work sharing) where work is offloaded eagerly and *work stealing* [8], a decentralised and adaptive mechanism where work distribution is demand-driven.

Communication. Large-scale applications require scalability beyond a single node mandating distribution of work across a cluster, a Grid, or in the Cloud. Two common schemes include *shared memory* (which can be distributed, requiring an additional abstraction layer) and *message passing*. Serialisation of computation and data is required if communication occurs across a network. A low-level library such as MPI [16] can be used to implement communication functionality. More sophisticated protocols such as publish/subscribe [12] may be beneficial for distributed cooperative RTSes which share contextual information.

Monitoring. Profiling, debugging, and execution replay rely on tracing (logging of relevant events), whereas dynamic optimisations rely on system information obtained through monitoring at run time. Common profiling strategies are, in order of increasing overhead, cumulative summary statistics, census-based statistics (collected at times of disruption, e.g. GC), fine-grained event-based statistics (using a sampling frequency or recording all events).

3 A Qualitative Run-time System Comparison

Table 1 provides an overview of several mainstream and research RTSes regarding their support for the features introduced above. We observe that a stack machine implementation is most common for mainstream languages whilst graph reduction is popular among declarative implementations. CLR provides support for tail call optimisation, so that declarative languages can be supported (e.g. F#), whereas JVM lacks direct support. JVM and CLR are compared in more detail elsewhere [15, 40], and although superficially similar they have many distinctive features (e.g. unlike JVM, CLR was designed for multi-language support).

As concurrency and parallelism support grow more important we observe that such support was added as an afterthought and often as a non-composable library to the mainstream languages. By contrast, Erlang was designed for concurrency and natively supports channels which facilitate communication among lightweight processes. Additionally, declarative languages often are a better match for parallelism due to immutability and graph reduction execution model [17]. For example, Haskell provides support for lightweight threads and data parallelism, whereas GUM supports adaptive distributed execution of GpH programs [29, 42].

Most of the RTSes support some kind of GC but only in few cases a more scalable design is used (e.g. Erlang uses per-process GC, X10 [10] and GUM use distributed GC). However, an empirical study is necessary to judge on the relative merits of different schemes. In mainstream languages communication and synchronisation are often explicit, increasing the complexity of concurrent and parallel programming. From this angle GpH appears as a very-high-level language in which all such aspects are implicit, whereas communication in X10 is implicit using the *Partitioned Global Address Space* (PGAS) abstraction whereas locality settings for the distributed data structures are explicitly provided by the programmer.

Many of the RTSes also employ adaptive optimisations of some kind such as JIT, work-stealing, or self-adjusting granularity control to throttle parallelism depending on available resources (e.g. in GUM). Like memory management, this is an area of ongoing research [2].

Table 1 A High-Level Overview of Ten Representative Run-time Systems.

RTS/ Language	Exec. Model	Concurrency/ Parallelism	Memory Management	Commun./ Synchron.	Adaptive Policies
crt0 C	stack machine	libs: fork, Pthreads	explicit malloc/free	sockets, MPI/ mutex, locks	user-defined
JVM Java/Scala/..	stack machine	libraries Threads, ..	impl. gen. mark&sweep	sockets, RPC/ synchronized	JIT
CLR C#/F#/..	stack machine	async/futures PLINQ, ..	implicit gen. m&s	sockets/ locks, wait	JIT
GHC-SMP Haskell	graph reduction	forkIO, STM, Par monad, ..	implicit gen. m&s	libraries/ MVars,TVars	work stealing
GUM Haskell ext.	graph reduction	par, Eval Strategies	impl. distr. ref. count.	both implicit (graph nodes)	work stealing
Manticore CML ext.	graph reduction	expl. tasks/ impl. data	impl. gen. local/global	expl. mesg./ sync	lazy tree splitting
X10 RTS X10	stack machine	async, futur./ foreach, ..	impl. distr. ref. count.	PGAS/ atomic	work stealing
BEAM Erlang	register machine	spawn primitive	per process compact./gen.	expl. mesg./ mesg. boxes	hot code swapping
Cilk C extension	stack machine	spawn, cilk func.	impl. cactus stack	explicit/ sync	work stealing
SWI Prolog ISO Prolog	term rewriting	threads/ concurrent	impl. low prio. thread	sockets/ mutex, join	assert/ retract

4 Alternative Views on the Role of the Run-time System

The proliferation of Cloud Computing as an economical approach to elastically scaling IT infrastructure based on actual demand and of Big Data Analysis in addition to Scientific High-Performance Computing applications have lead to an increased interest in parallel and distributed RTSes capable of managing execution across distributed platforms at large scale. Additionally, cache coherence protocols exhibit limited scalability and many novel architectures (e.g. custom System-on-Chip architectures) resemble distributed systems to some extent and may be non-cache-coherent [36].

Moreover, as many programming languages were not designed with parallelism and distribution in mind, their RTSes provide only minimal support for rather low-level coordination. This is an issue since most programmers are not parallelism and distributed systems experts and explicitly specifying coordination is deemed prohibitively unproductive, favouring flexible and safe RTS approaches. Convergence in RTS and OS functionality can be observed in addition to the aforementioned architectural trends that lead to promising alternative views.

4.1 Holistic Run-time System

A Holistic RTS [28] would move from separate RTS instances running on top of a host OS to a distributed RTS which takes over most of the OS functionalities and provides support for multiple languages simultaneously. The RTS would manage multiple applications at the same time and in a distributed fashion to avoid interference and coordinate distributed GC across isolated heaps. Additional benefits arise from the ability to share libraries and RTS components as well as to better utilise higher-level information for holistic optimisation.

4.2 Application-Specific Run-time System

At the other end of the spectrum, an alternative approach is based on a generic distributed OS [36] (e.g. based on the *Multikernel* approach [6]) offering a standardised interface for RTSes to communicate their demands, facilitating cooperative execution and helping avoid pathological interference. This way RTSes could be more lightweight and tailored for specific applications taking additional context information about application characteristics into account. To achieve this, a novel standard OS-RTS-interaction interface is required that would allow the OS to schedule RTSes and mediate the negotiations among them.

5 Challenges and Opportunities for Future Research

Recent trends towards heterogeneous and hierarchical parallel architectures and increased demand for high-level language features pose substantial challenges for the design and implementation of efficient language run-time systems offering opportunities for research.

5.1 Adaptation and Auto-Tuning for Performance Portability

As argued above, manual program tuning is deemed infeasible for most application domains, thus rendering automatic compiler and RTS optimisations critical for performance portability and scalability. Alas, many decisions have been shown to be NP-hard in general, thus requiring heuristics and meta-heuristics to facilitate heuristics choice based on available context information. Machine Learning has been shown to improve compiler flag selection [14] and it appears that similar techniques may be applicable for RTS-level adaptation. To achieve performance portability, the RTS must provide some performance guarantees based on applications' requirements and available architectural and system-level information.

5.2 Fault Tolerance

Traditionally, language RTS designs have focused on language features and performance, yet mechanisms to transparently detect, avoid, and recover from failures are becoming more important, as hardware failure is rather the norm than exception in modern large-scale systems [3]. In particular, the mean time between failures is rapidly decreasing following the exponential growth of the transistor budget in accordance with Moore's Law.

5.3 Safety and Security

One of the key benefits of using RTSes is that untrusted code can be executed safely in a sandboxed environment so that other applications are safeguarded from many negative effects that could be caused by malicious code. Additionally, intermediate code can be verified and type-checked by the RTS (e.g. as done by JVM's bytecode verifier) to eliminate the possibility of whole classes of common errors. Furthermore, automatic GC eliminates the source of memory corruption errors and avoids the buffer overflow vulnerability. It is an open issue to ensure safe and secure execution of sensitive workloads on Cloud-based infrastructures, which is the reason why many companies are reluctant to use new technologies or prefer investing into private data centres. Support for dependent and session types is another research direction focused on improving the safety of high-level languages [32, 21].

5.4 Unified Distributed Run-time Environment

As described above, both alternative RTS approaches require tighter integration of RTS and OS functions favouring cross-layer information flow. Moreover, a unified and extensible RTS needs to support diverse execution models for multi-language interoperability, as has been demonstrated by the CLR. The holistic view would facilitate avoidance of interference and enable cooperative solutions for resource management. Ultimately, sophisticated RTSes would hide coordination and management complexities from programmers increasing productivity and avoiding explicit overspecification of coordination to achieve high performance portability across diverse target platforms. On the other hand, it may be worthwhile to spatially and temporally separate applications and grant exclusive access to some resources on massively parallel architectures [27].

6 Conclusion

Based on the current trends, modern programming languages are likely to increasingly rely on a sophisticated RTS to manage efficient program execution across diverse target platforms. We have reviewed the role of the RTS in the software life cycle and the key RTS components and associated policies. Additionally, a qualitative comparison of mainstream and research RTSes illustrated the breadth of the design choices across the dimensions of execution models, memory management, support for concurrency and parallelism, as well as communication and adaptive policy control. The overview is complemented by references to influential historical results and current research outcomes alongside a discussion of alternative RTS designs followed by an outline of several promising areas for future research.

Acknowledgements. The author is grateful to Greg Michaelson and Hans-Wolfgang Loidl for thought-provoking discussions, to Julian Godesa, Konstantina Panagiotopoulou and Prabhat Totoo for useful comments, and to Siân Robinson Davies for designing the figures, as well as to the three anonymous reviewers for valuable feedback that improved this paper.

References

1 J. Armstrong. Erlang – a survey of the language and its industrial applications. In *Proceedings of INAP*, volume 96, 1996.
2 M. Arnold, S. Fink, D. Grove, M. Hind, and P. Sweeney. A survey of adaptive optimization in virtual machines. *Proc. of the IEEE*, 93(2):449–466, 2005.
3 A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.
4 J. Aycock. A brief history of just-in-time. *ACM Computing Surveys*, 35(2):97–113, 2003.
5 H. Barendregt. *The lambda calculus*, volume 3. North-Holland Amsterdam, 1984.
6 A. Baumann, P. Barham, P.-E. Dagand, T. Harris, R. Isaacs, S. Peter, T. Roscoe, A. Schüpbach, and A. Singhania. The Multikernel: a new OS architecture for scalable multicore systems. In *Proc. of 22nd ACM Symp. on Operating systems principles*, pages 29–44, 2009.
7 E. Belikov, P. Deligiannis, P. Totoo, M. Aljabri, and H.-W. Loidl. A survey of high-level parallel programming models. Technical Report MACS-0103, Heriot-Watt University, 2013.
8 R. Blumofe and C. Leiserson. Scheduling multithreaded computations by work stealing. *Journal of the ACM*, 46(5):720–748, September 1999.
9 A. Brodtkorb, C. Dyken, T. Hagen, J. Hjelmervik, and O. Storaasli. State-of-the-art in heterogeneous computing. *Scientific Programming*, 18(1):1–33, May 2010.

10 P. Charles, C. Grothoff, V. Saraswat, C. Donawa, A. Kielstra, K. Ebcioglu, C. von Praun, and V. Sarkar. X10: an object-oriented approach to non-uniform cluster computing. *ACM SIGPLAN Notices*, 40(10):519–538, 2005.

11 P. Deutsch and A. Schiffman. Efficient implementation of the Smalltalk-80 system. In *Proc. of the 11th ACM Symp. on Principles of Programming Languages*, pages 297–302, 1984.

12 P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114–131, 2003.

13 M. Fluet, M. Rainey, J. Reppy, and A. Shaw. Implicitly threaded parallelism in Manticore. *Journal of Functional Programming*, 20(5-6):537–576, 2010.

14 G. Fursin, Y. Kashnikov, A. Memon, et al. Milepost GCC: Machine learning enabled self-tuning compiler. *International Journal of Parallel Programming*, 39(3):296–327, 2011.

15 J. Gough. Stacking them up: a comparison of virtual machines. *Australian Computer Science Communications*, 23(4):55–61, 2001.

16 W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT Press, 1999.

17 K. Hammond. Why parallel functional programming matters: Panel statement. In *Proceedings of Ada-Europe*, volume 6652 of *LNCS*, pages 201–205, 2011.

18 J. Hennessy and D. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 5th edition, 2011.

19 M. Herlihy and J. E. B. Moss. Transactional memory: Architectural support for lock-free data structures. In *Proc. of the 20th Annual International Symposium on Computer Architecture (ISCA'93)*, pages 289–300. ACM, 1993.

20 U. Hölzle and D. Ungar. A third-generation SELF implementation: reconciling responsiveness with performance. *ACM SIGPLAN Notices*, 29(10):229–243, 1994.

21 K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. *ACM SIGPLAN Notices*, 43(1):273–284, 2008.

22 J. Hughes. Why functional programming matters. *Computer Journal*, 32(2):98–107, 1989.

23 J. Jaffar and J.-L. Lassez. Constraint logic programming. In *Proc. of the 14th ACM SIGPLAN Symposium on Principles of programming languages*, pages 111–119. ACM, 1987.

24 R. Jones, A. Hosking, and E. Moss. *The garbage collection handbook: the art of automatic memory management*. Chapman & Hall/CRC, 2011.

25 E. Lee. The problem with threads. *IEEE Computer*, 39(5):33–42, 2006.

26 T. Lindholm, F. Yellin, G. Bracha, and A. Buckley. *The Java virtual machine specification*. Pearson Education, 2014. First published in 1996.

27 R. Liu, K. Klues, S. Bird, S. Hofmeyr, K. Asanovic, and J. Kubiatowicz. Tessellation: Space-time partitioning in a manycore client OS. In *Proceedings of the 1st USENIX Conference on Hot Topics in Parallelism*, pages 10–10, 2009.

28 M. Maas, K. Asanovic, T. Harris, and J. Kubiatowicz. The case for the holistic language runtime system. In *First International Workshop on Rackscale Computing*, 2014.

29 S. Marlow, S. Peyton Jones, and S. Singh. Runtime support for multicore Haskell. *ACM SIGPLAN Notices*, 44(9):65–78, 2009.

30 V. Mayer-Schönberger and K. Cukier. *Big Data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

31 J. McCarthy. History of LISP. In *History of programming languages*, pages 173–185, 1978.

32 J. McKinna. Why dependent types matter. *ACM SIGPLAN Notices*, 41(1), 2006.

33 E. Meijer and J. Gough. Technical overview of the Common Language Runtime. *Language*, 29:7, 2001.

34 K. Nori, U. Amman, K. Jensen, and H.-H. Nägeli. *The Pascal-P Compiler: Implementation Notes*. ETH Zurich TR 10, 1974.

35 J. Owens, D. Luebke, N. Govindaraju, et al. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

36 S. Peter, A. Schüpbach, D. Menzi, and T. Roscoe. Early experience with the Barrelfish OS and the Single-Chip Cloud Computer. In *MARC Symposium*, pages 35–39, 2011.

37 M. Richards. The portability of the BCPL compiler. *Software: Practice and Experience*, 1(2):135–146, 1971.

38 T. Romer, D. Lee, G. Voelker, A. Wolman, W. Wong, J.-L. Baer, B. Bershad, and H. Levy. The structure and performance of interpreters. *ACM SIGPLAN Not.*, 31(9):150–159, 1996.

39 A. Silberschatz, P. Galvin, and G. Gagne. *Operating system concepts*. AddisonWesley, 1998.

40 J. Singer. JVM versus CLR: a comparative study. In *Proc. of 2nd Intl Conf. on Principles and Practice of Programming in Java*, pages 167–169. Computer Science Press, 2003.

41 J. Smith and R. Nair. The architecture of virtual machines. *Computer*, 38(5):32–38, 2005.

42 P. Trinder, K. Hammond, J. Mattson Jr., A. Partridge, and S. Peyton Jones. GUM: a portable parallel implementation of Haskell. In *Proc. of PLDI'96*, 1996.

43 J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012.

Affective Computing to Enhance E-Learning in Segregated Societies

Khaled El-Abbasy[1], Anastassia Angelopoulou[1], and Tony Towell[2]

1 **University of Westminster, Department of Computer Science**
115 New Cavendish Street, London, UK
{k.el-abbasy,A.Agelopoulou01}@my.westminster.ac.uk

2 **University of Westminster, Department of Psychology**
115 New Cavendish Street, London, UK
A.Towell@westminster.ac.uk

Abstract

According to UN Women, to build stronger economies, it is essential to empower women to participate fully in economic life across all sectors. Increasing women and girls' education enhances their chances to participate in the labor market. In certain cultures, like in Saudi Arabia, women contribution to the public economy growth is very limited. According to the World Bank [1], less than 20 percent of the female population participate in the labor force. This low participation rate has many reasons. One of them, is the educational level and educational quality for females. Although Saudi Arabia has about thirty three universities, opportunities are still limited for women because of the restrictions of access put upon them. A mixture of local norms, traditions, social beliefs, and principles preventing women from receiving full benefits from the educational system. Gender segregation is one of the challenges that limits the women access for education. It causes a problem due to the shortage of female faculty throughout the country. To overcome this problem, male faculty are allowed to teach female students under certain regulations and following a certain method of education delivery and interaction. However, most of these methods lack face-to-face communication between the teacher and students, which lowers the interactivity level and, accordingly, the students' engagement, and increases the need for other alternatives. The e-learning model is one of high benefit for female students in such societies. Recognizing the students' engagement is not straightforward in the e-learning model. To measure the level of engagement, the learner's mood or emotions should be taken into consideration to help understanding and judging the level of engagement. This paper is to investigate the relationship between emotions and engagement in the e-learning environment, and how recognizing the learner's emotions and change the content delivery accordingly can affect the efficiency of the e-learning process. The proposed experiment alluded to herein should help to find ways to increase the engagement of the learners, hence, enhance the efficiency of the learning process and the quality of learning, which will increase the chances and opportunities for women in such societies to participate more effectively in the labor market.

1998 ACM Subject Classification K.3.1 Computer Uses in Education

Keywords and phrases Affective computing, E-learning, Women education

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.13

1 Introduction

Education is a form of learning in which the knowledge, skills and habits of a group of people are transferred from one generation to the next through teaching, training, or research [2]. The need for education serves different purposes. Creating a platform of skilled and qualified

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 13–20

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

manpower able to communicate and compete with peers in the global market place, be it the educational, economic, or research industry has always been one essential need for education [3]. According to Rand, the developer of the philosophical system called 'Objectivism', "The only purpose of education is to teach a student how to live his life by developing his mind and equipping him to deal with reality" [4]. Similarly, Noam Chomsky says, "Education is really aimed at helping students get to the point where they can learn on their own" [5]. However, education cannot achieve its goals unless it is efficient.

Efficiency is a way of measuring the level of performance in producing the best amount of outputs from a process that uses a limited amount of inputs. Thus, the efficiency of education is highly dependent on the delivery method. Students learn best when they actively participate in the learning process, when they are engaged and motivated to learn, and when they can build on their existing knowledge and understanding [6].

For as long as the concept of education has existed, there was only one form of education called the traditional education where the teacher has contact with students in a learning environment often called "classroom". The basic technique in this kind of educational setting was oral recitation where students listen quietly to their mates one by one reciting the lesson. The role of the teacher was to assign and listen to these recitations and assess the performance of students usually by oral examination. This kind of education was inefficient to a high extent as it was highly dependent on rote memorization, which is memorization with no effort at understanding the meaning [7].

Progressive education was later introduced as a new approach and form of education that emphasis learning by doing. Progressive learning is more learner oriented. In progressive education, the learner has more freedom to choose what to learn and how to learn by getting involved in hands-on projects, which is a form of what is called Experiential Learning (learning from experience). This form of education is highly engaging for students as it includes a high level of interaction. The learner is more independent, self-motivated, and more involved in collaborative and cooperative learning projects. The increased level of participation of learners in progressive education makes this type of setting much more efficient than the traditional one, and closer to achieving the learning objectives.

Although progressive learning is more learner oriented, the teacher still has an important role. The teacher is expected to be a facilitator of learning – encourage students to use a wide variety of activities, manage their discussions, guide them through the wide variety of materials and asses their performance. In other words, teachers adopt and adapt different tools to help students absorb the materials. Technology is one of these tools. It is used to overcome one or more of the shortcomings of traditional instruction. It combats boredom by adding motivational features that excite learners' interest. Technology in teaching in its early phases denoted the use of traditional media like visual and audio media. However, with the advancement of technology, the traditional media turned digital where the use of computers, interactive multimedia and e-learning turned out to be the main characteristic of the modern educational format.

E-learning refers to the use of electronic media and information and communication technologies in education. Bernard Luskin, a pioneer of e-learning, advocates that the "e" should mean "exciting, energetic, enthusiastic, emotional, extended, excellent, and educational" in addition to "electronic" [8]. E-learning can be used for distance learning, which is a way for delivering education and instruction to students who are not physically present in a traditional setting such as classroom. It has two modes of delivery, either synchronous or asynchronous.

In synchronous learning, all participants should be present at the same time. Different technologies can be used for delivery such as web conferencing, videoconferencing, instructional

television, direct-broadcast television, internet radio, live streaming, telephone and VoIP. On the other hand, asynchronous learning does not require the presence of the students at the same time. They can access the course materials on their own schedule. Mail correspondence, message board forums, email, interactive media, video and audio recording are different examples for technologies used for this mode. Moreover, a recent distance learning technology called MOOCs is used to teach online a massive number of students with minimal involvement by professors. MOOC, which stands for "massive open online courses" can be either synchronous or asynchronous. E-learning usually does not require face-to-face interaction between the students and the instructor. However, it can be mixed with campus based education, and this is known by blended learning or hybrid learning.

According to Gardner [9], not all people have the same abilities, thus not all of us learn the same way. Students learn in different ways and usually have their own styles and preferences for learning. This theory is supported by research on what is called Learning Styles [10].

Learning styles refers to a cluster of psychological behaviors to determine how an individual perceives, interacts with, and responds emotionally to learning environments. Most learning style theorists have settled on four basic styles:

- *The Mastery Style Learner.* Absorbs information concretely; processes information sequentially, in a step-by-step manner; and judges the value of learning in terms of its clarity and practicality.
- *The Understanding Style Learner.* Focuses more on ideas and abstractions; learns through a process of questioning, reasoning, and testing; and evaluates learning by standards of logic and the use of evidence.
- *The Self-Expressive Style Learner.* Looks for images implied in learning; uses feelings and emotions to construct new ideas and products; and judges the learning process according to its originality, aesthetics, and capacity to surprise or delight.
- *The Interpersonal Style Learner.* Like the Mastery learner, focuses on concrete, palpable information; prefers to learn socially; and judges learning in terms of its potential use in helping others.

Learners develop and practice a mixture of styles as they live and learn. Teachers should be able to analyze students and recognize their different abilities and learning styles in order to design and plan lessons in a way that helps all students absorb and process information efficiently.

E-learning may be a luxurious choice for some, but for others it may be the preferable or the only possible route to education. This may be the case in segregated societies, like the one in Saudi Arabia, where female students have limited face-to-face interaction with their instructors due to gender segregation. E-learning is indeed the future of education in today's global world; however, its major drawback is the absence of a physically present instructor capable of observing learners and interacting with them according to their needs. This research aims at developing an affective E-learning framework for all-female settings in gender segregated-societies such as Saudi Arabia, by applying a multimodal system (e.g. facial expressions) that measures and senses the learners' reaction to the materials used during the learning process. By doing so, women and girls' education will enhance their chances to participate in the labor market, hence, building stronger economy.

2 Learning engagement

For all kinds of education: traditional, progressive, e-learning or blended learning, the major challenge is how to achieve maximum efficiency of the education process and keep learners

engaged during the learning process. According to Bangert-Drowns and Pyke, truly engaged learners are behaviorally, intellectually, and emotionally involved in their learning tasks [11].

The first step in engaging learners is determining their interest level. According to Schmoker, there are evidences to detect [12]

- paying attention (alert, tracking with their eyes),
- taking notes,
- asking / responding to questions, and
- reacting (laughing, crying, shouting, etc.).

In face-to-face teaching, experienced teachers recognize the engagement level of the students and react accordingly. They change their teaching method during the class to grab the students' attention. For example, the teacher can change from a presentation method to a discussion one, ask a challenging question, tell a story, or play a game. Mixing different teaching methods and strategies in the teaching process engages students and efficiently achieves the set educational goals. This strategy can be adoptable in the traditional and progressive education forms, where the teacher has direct contact with students and can recognize their engagement level. However, in some cultures, where education is segregated based on gender as a result of social regulations, face-to-face teaching is not applicable. For example, in Saudi Arabia, this segregation causes a problem at the level of higher education due to the shortage of female faculty throughout the country. To overcome this problem, male faculty are allowed to teach female students under certain regulations and following a certain method of education delivery and interaction. One of these methods involves the use of closed-circuit television (CCTV) for transmitting a male faculty's lecture delivered to female students [13]. Another method involves teaching in a special lecture room divided by a dark one-way see – through glass barrier, where the students may view the professor, but not the other way around [14].In very rare cases, male professors can teach female students directly and without barriers. However, in many of these cultures female students cover their faces in the presence of male instructors. The absence of face-to-face communication in such cases, lowers the interactivity level and, accordingly, the students' engagement, and increases the need for other alternatives. The e-learning model is one of high benefit for female students in such societies. However, recognizing the students' engagement is not straightforward in the e-learning model, where there is no direct contact between the instructor and the learner. In e-learning, the physical activities that demonstrate engagement like taking notes or tracking teacher with eyes are not applicable; instead, activities related to the usage of computers like focus on the display, strike on the keyboard, click on the mouse, and stay on the web page are good indicators of engagement. However, more indicators are needed to measure the level of engagement more accurately. Therefore, the learner's mood or emotions should be taken into consideration to help understanding and judging the level of engagement.

3 Emotions

Emotion refers to a shaking of the organism as a response to a particular stimulus (person, situation or event), which is generalized and occupies the person as a whole [15]. Usually, it is very brief in duration, which makes it different than "mood". Mood is a feeling that tends to be less intense than emotion and often lacks a contextual stimulus [16]. Both emotions and moods are encompassed under the umbrella of "affect", which is a generic term that covers a broad range of feelings that people experience [17].

Emotions should be characterized and distinguished from other states. Through time, different scholars proposed different classifications of emotions. For example, Rene Descartes

classified emotions as "love, hate, desire, joy, sadness, and admiration" [18]. Other scholars (Spinoza, Hobbes, Jorgensen, and Tomkins) identified other categories of emotions (ibid.)

In modern research, psychologists tried to categorize basic emotions according to facial expressions. Ekman developed a list of basic emotions that are not culturally determined, but universal. These basic emotions are "anger, disgust, fear, happiness, sadness, and surprise" [19]. Emotions are best treated as multifaceted phenomena consisting of the following components: behavioral reactions (e.g. retreating), expressive reactions (e.g. smiling), physiological reactions (e.g. heart pounding), and subjective feelings (e.g. feeling amused) [20]. From an educational point of view, emotions can be classified into positive and negative emotions. Positive emotions encourage students to engage and achieve, such as joy (enjoyment of learning), hope and pride. In this case, Csikszentmihalyi's model of "flow" can be applied; in which there is a zone where people can concentrate their attention so intensely on solving a problem or doing things that they lose track of time [21].]. Such flow is optimal experience that leads to happiness and creativity. If the task is not challenging enough or too challenging, negative emotions such as anger, anxiety, shame or boredom affect the efficiency of learning.

Research shows that a slight positive mood does not just make the student feel a little better but also stimulates a different kind of thinking, characterized by a tendency towards greater creativity and flexibility in problem solving, as well as more efficiency and thoroughness in decision making [22].

4 Emotion recognition

Different techniques and measurement tools are applied to recognize emotions. These tools can be grouped into three areas: Psychological, Physiological and Behavioral [23]

4.1 Psychological tools

These tools are derived from clinical psychology. They apply verbal and non-verbal descriptions of emotions and can be classified into: Verbal self-reporting (e.g. questionnaires, interviews) and Non-verbal self-reporting (e.g. photographs). These tools are inexpensive and non-invasive; however, different factors affect their reliability and should be taken into consideration such as consciousness. Users are often reluctant to disclose their inner feelings to researchers out of embarrassment [24].

4.2 Physiological tools

These tools depend mainly on the use of sensors to measure subject's physiological reactions. They are based on recording electrical signals produced by the brain, heart, muscle, and skin. Examples of these tools are [25]:

- Electromyogram EGM: measures muscle activity
- Electroencephalography EEG: measures brain activity
- Electrodermal Activity EDA or Skin Conductance SC: measures hydration in the epidermis and dermis of the skin
- Electrocardiogram ECG: measures heart activity
- Electrooculogram EOG: measures eye pupil's size and movement
- Blood Volume Pulse BVP: measures blood pressure

4.3 Motor-behavioral tools

These tools measure behavioral expressions and changes in physical body states that reflect people's emotions. Traditional devices can be used within these tools like a PC camera,

microphone, mouse, and keyboard, in addition to sensors and special software. Examples of these tools are: Speech recognition, Hand tracking–body posture, Facial expressions, etc.

5 Proposed affective e-learning platform

The intended research aims to investigate an affective e-learning model in a specific culture. The research will be conducted in Saudi Arabia, in particular Effat University. Effat University is a private non-profit institution of higher education for women in Saudi Arabia, established in 1999. The University has three colleges: College of Engineering, College of Humanities and Social Science, and College of Business. A total of 2346 students are currently enrolled in Effat University – 53 % in the College of Engineering. The University employs 149 faculty members, 26 % of whom are male faculty. This research will be conducted on students attending the College of Engineering where a high level of interaction is needed to teach the practical components of courses. A pilot study will be carried out on a small group of students before the main study to identify the feasibility of the investigation and detect and solve any possible difficulties. In the pilot study we will also be testing the validity of the affective recognition software by using sensors to measure arousal, and visual engagement with the material. The pilot study will be conducted at the University of Westminster with its brand new state-of-the-art Cognitive Development Laboratory, which hosts a Tobii TX300 eye- tracker, a 64-channel Neuroscan and Biosemi Active2 EEG system.

5.1 Participants

Six groups of students in their first year of study will be recruited to participate in the study. Three groups from the University of Westminster will participate in the pilot study (n=15: 5 students for each group). The other three groups will be recruited from Effat University in Saudi Arabia where the main study will be conducted (n=75: 25 students for each group). Students will be randomly selected, the age is approximately matched between the groups (age range will be 18–22). Each group in the pilot study and the main investigation will be taught using one of three approached: face-to-face, e-learning, and affective e-learning.

5.2 Methods

The three groups in the pilot study and the main study will be taught according to the following plan: Group A will be taught by a male faculty using the traditional face-to-face approach, Group B will use an e-learning approach where they have complete control of materials delivery, and Group C will use the proposed affective e- learning approach, where the system will control materials delivery following the student's emotions. All groups will be assessed in the same way at the end of course and the results of their assessment will be evaluated and compared taking into consideration the approach adopted in teaching.

5.2.1 Phase 1 – Teaching

The selected topic will be taught to the three groups using the three approaches as follows: Group A: Will use the traditional face-to-face approach. An instructor will be involved. Group B: Will use only the e- learning approach. No direct contact with the instructor. Group C: Will use the affective e-learning approach. Although, there will be no direct contact with the instructor, but the process will be monitored remotely to ensure smooth running of the process. The same materials used with group B will be used with group C; however,

the system will control the delivery of the materials according to the user's emotions and attention, which will be recognized using the proposed facial expression recognition system.

5.2.2 Phase 2 – Assessment

An assessment will be conducted to examine and assess the performance of the three groups. Assessment will be divided into three stages: Baseline assessment, Formative assessment during the learning period, and Summative post learning assessment.

5.3 Analyses

SPSS software will be used to compare outcome measures between the 3 groups using two way ANOVA (time of assessment x group) with appropriate tests of pairwise comparisons. Correlational analysis will also be used to determine which factors at baseline correlate with final module performance including demographic variables and learning style classification. If time permits, a further assessment test will be carried out a few months following the end of the module to establish the efficacy of affective e learning on longer term retention.

6 Conclusion

Learners have different emotions and different learning styles. In traditional education, they respond to the teaching methods in visible ways. Teachers thus often alternate delivery methods in an effort to meet all learning styles. E-learning, on the other hand, provides individual learners with the same content and instructions, regardless of their learning styles. Designing an e-learning system that will recognize the emotion(s) of the learner, which can be a result of his/her learning style, and apply different content or different teaching methods for each learner may engage the student in a better way hence enhance the efficiency of the education process. Such a model will highly benefit e-learning systems.

References

1 Labor force participation rate, female (percentage of female population ages 15+) (modeled ILO estimate), http://data.worldbank.org/indicator/SL.TLF.CACT.FE.ZS, 28 05 2015.

2 Education, http://En.wikipedia.org/wiki/education, 12 04 2015.

3 What is the purpose of Education, http://www.forbes.com/sites/sap/2012/08/15/what-is-the-purpose-of-education/, 18 04 2015.

4 Objective Education, http://objectiveeducation.com/, 22 04 2015.

5 Parenting, http://www.parenting.com/blogs/mom-congress/melissa-taylor/what-purpose-education, 22 04 2015.

6 John D. Bransford, Ann L. Brown, and Rodney R. Cocking, *How People Learn: Brain, Mind, Experience, and School*,National Academy Press, 2000.

7 Traditional Education, http://en.wikipedia.org/wiki/Traditional_education, 12 05 2015.

8 Think "Exciting": E-Learning and the Big "E", http://www.educause.edu/ero/article/think-exciting-e-learning-and-big-e, 12 05 2015.

9 Gardner, H., *Frames of Mind: The Theory of Multiple Intelligences.* New York: Basic Books, 1983.

10 Integrating Learning Styles and Multiple Intelligences, http://www.ascd.org/publications/educational-leadership/sept97/vol55/num01/Integrating-Learning-Styles-and-Multiple-Intelligences.aspx, 22 04 2015.

11 Bangert-Drowns, Robert L., Pyke, *A Taxonomy of Student Engagement with Educational Software: An Exploration of Literate Thinking with Electronic Text.* Journal of Educational Computing Research, **24** (2001), pp. 213–34.

12 Schmoker, M. J., *Results Now: How We Can Achieve Unprecedented Improvements in Teaching and Learning.* New York: Association for Supervision and Curriculum Development, 2006.

13 Del Castillo, D., *Teaching Through an Electronic Veil.* The Chronicle of Higher Education, **29** (2003), pp. A48.

14 Abdulrahman, M., *Students' Perceived Barriers to In-Class Participation in a Distributed and Gender Segregated Education Environment.* Electronic Journal of Information Systems in Developing Countries, **35** (2008), pp. 1–17.

15 Michalis Feidakis, Thanasis Daradoumis, Santi Caballé, *Emotion Measurement in Intelligent Tutoring Systems: What, When and How to Measure.* Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2011,Fukuoka, Japan, pp. 807–812.

16 H. M. Weiss and R. Cropanzano, *Affective Events Theory: A Theoretical Discussion of the Structure, Causes and Consequences of Affective Experiences at Work.* in B. M. Staw and L. L. Cummings (eds.), Research in Organizational Behavior, **18** (1996), pp. 17–19.

17 George, J. M. , *Trait and State Affect.* In Individual Differences and Behavior in Organizations (1996), pp. 145.

18 Guey-Shya Chen, Min-Feng Lee, *Detecting Emotion Model in e-Learning System.* International Conference on Machine Learning and Cybernetics, 2012, Xian, pp. 1686–1691.

19 Ekman, P, *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life.* New York: Times Books/Henry Holt and Co., 2003.

20 Desmet, P. M., *Measuring Emotions: Development and application of an instrument to measure emotional responses to products.* In A. M. M.A. Blythe, Funology: From Usability to Enjoyment, Dordrecht: Kluwer Academic Publishers., 2003.

21 Csikszentimihalyi, M., *Flow: The psychology of optimal experience.* New York: Harper and Row, 1990.

22 Isen, A. M., *Positive Affect and Decision Making.* In M. L. Haviland, Handbook of Emotions,New York: The Guildford Press (1990), pp. 720.

23 Zimmermann, P. G., *Beyond Usability-Measuring Aspects of user Experience.* Doctoral dissertation, Zürich, Swiss Federal Institute of Technology, 2008.

24 Wong, M., *Emotion assessment in evaluation of affective interfaces.* Master Thesis, Ontario: University of Waterloo, 2006.

25 Andreas Haag et al., *Emotion Recognition from Physiological Signals using Bio Sensors: First Steps towards an Automatic System*, Lecture Notes in Computer Science (2004), pp. 36–48.

Mining Scientific Articles Powered by Machine Learning Techniques

Carlos A. S. J. Gulo[1,2], Thiago R. P. M. Rúbio[1,3], Shazia Tabassum[1,4], and Simone G. D. Prado[5]

1 **Departamento de Engenharia Informática, Faculdade of Engenharia, Universidade do Porto, Porto, Portugal**

2 **PIXEL Research Group, UNEMAT, Brazil**
sander@unemat.br

3 **LIACC – Artificial Intelligence and Computing Science Laboratory, Universidade do Porto, Porto, Portugal**
reis.thiago@fe.up.pt

4 **LIAAD, Inesctec, Porto, Portugal**
shazia.tabassum@inesctec.pt

5 **Departamento de Computação , Faculdade de Ciências, Universidade Estadual Paulista, Bauru, Brazil**
simonedp@fc.unesp.br

Abstract

Literature review is one of the most important phases of research. Scientists must identify the gaps and challenges about certain area and the scientific literature, as a result of the accumulation of knowledge, should provide enough information. The problem is where to find the best and most important articles that guarantees to ascertain the state of the art on that specific domain. A feasible literature review consists on locating, appraising, and synthesising the best empirical evidences in the pool of available publications, guided by one or more research questions. Nevertheless, it is not assured that searching interesting articles in electronic databases will retrieve the most relevant content. Indeed, the existent search engines try to recommend articles by only looking for the occurrences of given keywords. In fact, the relevance of a paper should depend on many other factors as adequacy to the theme, specific tools used or even the test strategy, making automatic recommendation of articles a challenging problem. Our approach allows researchers to browse huge article collections and quickly find the appropriate publications of particular interest by using machine learning techniques. The proposed solution automatically classifies and prioritises the relevance of scientific papers. Using previous samples manually classified by domain experts, we apply a Naive Bayes Classifier to get predicted articles from real world journal repositories such as IEEE Xplore or ACM Digital. Results suggest that our model can substantially recommend, classify and rank the most relevant articles of a particular scientific field of interest. In our experiments, we achieved 98.22% of accuracy in recommending articles that are present in an expert classification list, indicating a good prediction of relevance. The recommended papers worth, at least, the reading. We envisage to expand our model in order to accept user's filters and other inputs to improve predictions.

1998 ACM Subject Classification H.3.3 Information Search and Retrieval

Keywords and phrases Machine Learning, Text Categorisation, Text Classification, Ranking, Systematic Literature Review

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.21

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 21–28

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Literature review is one of the most important phase of research to ascertain the state of the art of any specific domain area [4, 17, 6]. However, is possible to perform a literature review guided by different methodologies [17], for instance using traditional literature review or systematic literature review. In summary, a traditional literature review aims to perform a collection of information about researchers, theories and hypothesis, as such as how to solve a research problem using novel methodologies [4]. The outcome of this type of literature review, in general, is in reports or in thesis' chapter. Otherwise, the Systematic Literature Review (SLR) can be defined as a set of procedures which allows analysing systematically an interesting literature locating, appraising, and synthesising the most relevant researches in a domain area. The description of procedures should allow a reproducible literature review. Besides it, a literature review phase also helps to identify the gaps and challenges in that area. However, searching for interested articles in electronic databases do not retrieve the most relevant content indeed, although, the search engines recommend articles in which the specific keywords occur.

Automatic recommendation of articles is a challenging problem [20], mainly because the most scientific article contents are represented by text [15]. Text representation is critical in some text processing applications such as text categorisation [1], information retrieval[10], and topic modelling[3, 9]. Moreover, it's not a trivial process generating useful features from text representation to be used in many machine learning algorithms to support natural language processing [15].

The motivation for the development of the proposed model is providing automatically an efficient way of recommend, classify and rank important scientific literature. The manual process of finding and reviewing the most relevant literature that supports a research hypothesis is time consuming and error-prone, although researches [20, 3, 9, 1] for recommending scientific articles to users based on other users' ratings have showed good results. Our contribution in this new stage of our work is an automatic recommender system design focused on systematic literature review methods. The proposed solution is based on machine learning techniques and the process automatically classifies and prioritise the relevance of scientific papers.

In this paper we combine text mining and machine learning techniques as support to identify the most relevant literature using a data set collection searched in many journal repositories: ACM Portal [1], Engineering Village [2], IEEE Xplore [3], ScienceDirect [4], Web of Science [5]. Data set is analysed quantitatively in order to reduce the time used to review papers to write the literature review about our research domain: *high performance computing as support to computer aid diagnostic systems using medical images.*

Text Mining is a common process of extracting relevant information using a set of documents [8]. It provides basic preprocessing methods, such as identification, extraction of representative characteristics, and advanced operations as identifying complex patterns [9, 8]. Document classification is a task that consists of assigning a text to one or more categories: the name of its class of subject, and main topics.

The rest of the paper is organised as follows. In section 2 reviews the related works. Section 3.1 presents articles recommendation module, in Section 3 the experiments performed

[1] http://dl.acm.org/
[2] http://www.engineeringvillage.com/
[3] http://ieeexplore.ieee.org
[4] http://www.sciencedirect.com
[5] http://apps.webofknowledge.com

using Naive Bayes and the results obtained with the sets of scientific articles considered in the automatic text recommendation, are discussed in Section 3.2, which is followed by the concluding remarks in Section 4.

2 Background

Automatic recommendation of scientific articles consists on many sub-tasks, namely: data collection, text processing, data division, features extraction, feature selection, data representation, classifier training, applying the classification model, and performance evaluation [9].

Starting with data collection, we have to manage gathering the relevant references from known databases, such as literature repositories or other specific way to get documents. With this data, text preprocessing should remove undesirable information that represent noise. Stop words are removed (prepositions, pronouns, articles, adverbs and other auxiliary words) and the resulting words are steamed[21, 9].

Feature extraction reflects the terms we want to extract from the text. It may be related to the content (keywords) or not (author name, publication date, etc.), depending on data mining goals. At this step, the data is stored as a matrix that match the selected features with their weighting in the text. The calculation of the weighting can be obtained using statistical methods, such as the frequency on the documents (absolute or relative) [22, 7].

At this point, the data is divided into two main sets: training and test. We apply classifier algorithms to the training set in order to obtain a model that can predict a class or label to unseen data (test). These models usually recur to statistical approaches or machine learning paradigms. There is no ideal ratio of training data to testing data. The classification performance is the average performance of implemented classification models[24, 12]

Machine learning algorithms consist on recognising patterns from a data set and we aim to evaluate the extrapolation with unknown data. Many statistical algorithms can be used to create a model for classifying or labelling, such as [2, 5, 16, 18, 19, 23]: Latent Semantic Analysis (LSA) Language Model, Gaussian Model, Bayesian Model, among others. Various techniques are used: Support Vector Machine (SVM), Naive Bayes classifier, K-Nearest Neighbour (K-NN), Rocchio Algorithm, Decision Trees, Ensemble Classifiers, Inductive Logic Programming (ILP).

The last steps are the performance evaluation and the classification itself. As the training data have already a target value (previous) classification, if we present this same data to the trained model, the resulting performance is then obtained. The most common metrics for evaluating performance are accuracy, recall and precision. Recall correspond to the ability of the algorithm in retrieving the most relevant documents, meanwhile precision shows the capacity of the model in excluding not interesting documents. Once the model's predictive performance is adjusted, the final step consists on presenting new and unseen data (the test set from data division) and get the final result of the classification model [21].

3 Experiments And Model Design

The infrastructure used to perform the experiments and also illustrates the obtained results was composed of the Rapidminer Predictive Analytics Platform available to download through the Rapidminer website [6]. We have used the Rapidminer to construct the models and analyse

[6] `https://rapidminer.com/` – Rapidminer is a visual environment for predictive analytics, and it's considered easy-to-use just following the simple and intuitive instructions, and it's not required programming any code line to build models and make predictions

Table 1 Total of articles searched in journal repositories.

Repositories	Publication	
	Searched Queries	Papers
ACM Portal	("medical image") and ("high performance computing" or "parallel computing" or "parallel programming") and (PublishedAs:journal) and (FtFlag:yes) and (Abstract-Flag:yes)	28
Engineering Village	(((((medical imag*) WN KY) AND ((high NEAR/0 performance NEAR/0 comput*) WN KY)) AND ((parallel NEAR/0 comput*) WN KY)) AND ((parallel NEAR/0 programm*) WN KY)), Journal article only, English only	22
IEEE Xplore	(((medical imag*) AND (("high performance comput*" OR "parallel programm*") OR "parallel comput*")))	68
ScienceDirect	"medical image" AND ("high performance computing" OR "parallel computing" OR "parallel programming")[Journals(Computer Science,Engineering)]	409
Web of Science	((("high performance comput*") OR ("parallel comput*") OR ("parallel programm*")) AND ("medical imag*"))	72
Total		599

the application of the algorithm. A portable computer equipped with an Intel(R) Core(TM) i7-2630QM 2.0 GHz, 8GB of RAM (DDR3 1333 MHz), Linux Debian Stretch (64 bits) operating system.

Data set used in experiments was built using the searching queries in each repository included in the previous Table 1, and composed by 575 observations (after removing 24 duplicated references), and 4 variables (*id, Title, Abstract, and Priority)*. The analysed variable is text data, the *Abstract*, and its unstructured data. Unstructured data has variable length, one observation contains a scientific text, it has variable spelling using singular and plural forms of words, punctuation and other non alphanumeric characters, and the contents are not predefined to adhere to a set of values, it requires converting it to structured data for further processing. The preprocessing steps, provided by Text Mining methods, are responsible to make everything lowercase, remove punctuation and spaces, extract words from the data, replace synonyms, plural and other variants of words with a single term, reduce words to their stem, and remove common English stop-words, finally, create the structured data in table format where each word becomes a variable with a numeric value for each record [8].

3.1 Ranked-recommendation Based On Machine Learning Classification Model

This research work is new in terms of the methodology used to rank and prioritize papers. As a general classification model, we classify the scientific papers using a Naive Bayes classification algorithm. As a novel method we improve over it by extending the model to build a ranking model over the classification model as shown in Figure 1. This model uses the word list from the trained model and the already classified model using Naive Bayes Classifier. Then generates a ranking model which can be used as a recommendation system for future searches.

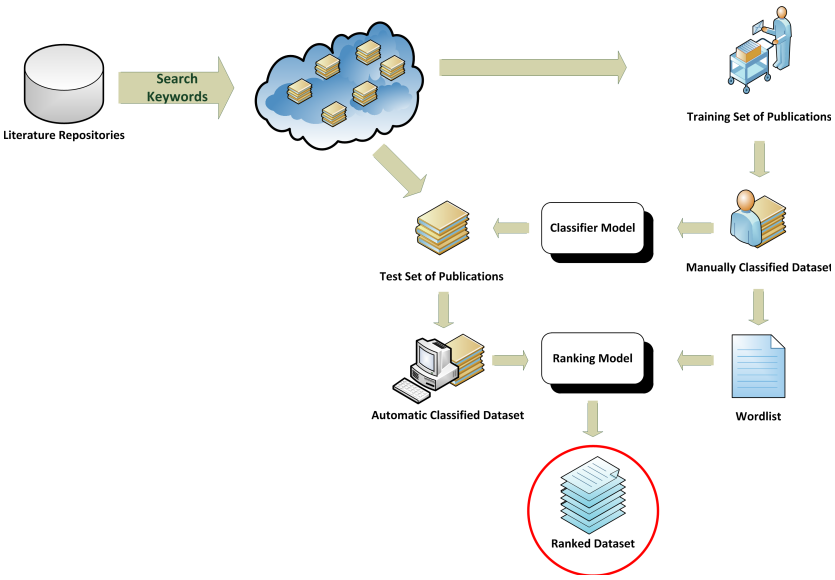


Figure 1 Architecture of the model process.

We are interested in establishing an automatic process able to classify and rank publications from a personal literature collection. Our main goal is to achieve the same level of relevance as performed by a human expert. We have seen that this process consists in various parts, following the steps described in Section 2 and selected the Naive Bayes algorithm as our first attempt to classify scientific papers. Although, Naive Bayes algorithm is not considered the most precise, otherwise is very simple to work with and to configure [23]. Using a previously retrieved data set, a human expert in a specific domain has analysed each one of the observations and classified the priority of the references regarding two main criteria: relevance of the reference and adequacy to the interested scientific domain. Its analysis consisted in classifying the reference into three priority classes:

- *Prio1*: References that are very relevant and adequate to the expected search;
- *Prio2*: References that are not so relevant but still adequate;
- *Prio3*: References that somehow interesting to the new research, but not the main source of knowledge.

A Naive Bayes model could then, be trained using the classification given by the expert. The classifier Naive Bayes is a supervised learning algorithm based on the Bayes theorem, which has strong independence features. Naive Bayes can be used with other models and play the role of vectorizer [11], obtaining hybrid models that best fit in certain classifications.

Figure 2 shows the Rapidminer model with the selected blocks responsible for training and performance evaluation of the Naive Bayes. The process of automatic classifying publications starts then, with a selected set of keywords that represent the context and the area of interest. We make a search in literature databases looking for the references that matches our filtering criteria (defined by our systematic review protocol). This set of references is the main data set we want to analyse, then it's divided into two parts: training and test. The training set corresponds to a smaller fraction of references that will be submitted to the user (expert in that domain) so that it should be classified manually. Simultaneously, based on the most important concepts presented on the analysed set, is created a *dictionary* of terms.

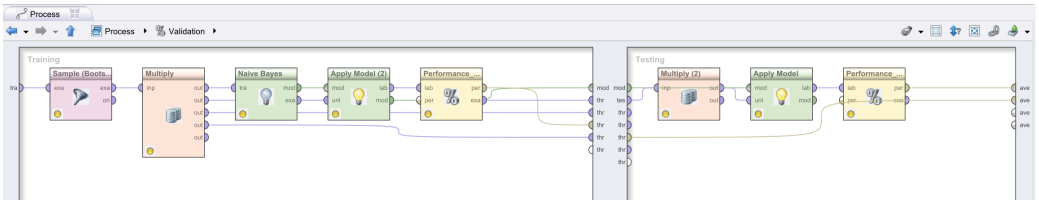


Figure 2 Naive Bayes Training process.

Table 2 Naive Bayes Results.

	true prio1	true prio2	true prio3	class precision
pred. prio1	833	0	0	100.00%
pred. prio2	0	211	63	77.01%
pred. prio3	0	0	2423	100.00%
class recall	100.00%	100.00%	97.47%	

3.2 Results

A Naive Bayes model is created and learns the classification patterns used by the domain expert. When this classifier model is applied to test the reference data set, the result is an automatic classified set of references. As seen in Table 2, after tuning sets, the final model obtained: 98.22% of accuracy, 92.84% of precision and 99.15% of recall. The importance of the Naive Bayes model created here is to guide the configuration of new models for different study fields.

We have used K Fold Cross Validation for estimating the performance of the classifier. In k fold cross validation sometimes called rotation estimation the data set D is randomly split into k mutually exclusive subsets the folds $D_1, D_2.....D_k$ of approximately equal size. The inducer is trained and tested k times; each time t $\in \{1, 2...k\}$, it is trained on $D \setminus D_t$ and tested on D_t. The cross validation estimate of accuracy is the overall number of correct classifications divided by the number of instances in the data set. Formally let $D_{(i)}$ be the test set that includes instance $x_i = (v_i, y_i)$ then the cross validation estimate of accuracy [13].

$$acc_{cv} = \frac{1}{n} \sum_{(v_i, y_i) \in D} \delta(I(D \setminus D_(i), v_i), y_i) \tag{1}$$

For many methods of text analysis, specifically the so called "bag-of-word" approaches, we created a common data structure for the text (Document Term Matrix – DTM) [14, 21, 9]. This is a matrix in which the rows represent references and columns represent terms. The values represent how often each word occurred in each reference. Not all terms are equally informative of the underlying semantic structures of texts, and some terms are rather useless for this purpose. In order to produce text statistics, for instance, the most common terms in the text, we used the Term Frequency Inverse Document Frequency (TFIDF) [9].

TFIDF, is a numerical statistic which indicates how important a term is to a reference in our collection. It is often used as a weighting factor in text mining. The TFIDF value increases proportionally to the number of times a term appears in the reference, but is offset by the frequency of the term in the collection, which helps control the fact that some terms are generally more common than others. Variations of the TFIDF weighting scheme are

often used by search engines as a central tool in scoring and ranking a reference's relevance given a user query [9]. TFIDF was successfully used for stop-words filtering and classification. One of the simplest ranking functions is computed by summing the TFIDF for each query term; many more sophisticated ranking functions are variants of this simple model[10, 9].

$$TFIDF(i) = \frac{Frequency(i) * N}{df(i)}, \tag{2}$$

$$\mathrm{R} = (\alpha * \frac{1}{prio}) * (\frac{wordsinwordlist}{totalwords}), \tag{3}$$

where *wordinwordlist* is the frequency of words in all documents, and *totalwords* is the number of words in the collection. Here, we apply the *dictionary-based approach* and create a ranking mechanism to obtain a relevance score (R) for each paper. The relevance score R is calculated in the Equation (3), where a paper is considered more relevant depending on its priority (1, 2 or 3) and the percentage of the most relevant terms are present in its abstract. Finally, we prune the ranked publications set recommending the top 10 most relevant references for a specific search.

4 Conclusion And Future Work

We proposed a model for recommending scientific articles to users based on abstract content using a personal collection of references. In general, building a large amount of labelled training data for text classification is a labour-intensive and time-consuming task. Our study showed that this approach works well considering our initial purpose and make good predictions on recommending scientific articles based on references collection. We believe that our approach have promising results, mainly because it's suitable to be applied in all domains. The results demonstrated the effectiveness and applicability of automated reference classification methods for management and updating a systematic literature review, required in all research project. In future work we will compare our model with Support Vector Machines and Boosting, besides integrating the first model developed previously in [9].

Acknowledgements. The first author thanks Universidade do Estado de Mato Grosso (UNEMAT – Brazil) and Faculdade de Engenharia da Universidade do Porto (FEUP – Portugal), for the support given. The work of Thiago, has been funded through a IBRASIL Grant. IBRASIL is a Full Doctorate programme selected under Erasmus Mundus, Action 2 STRAND 1, Lot 16 and coordinated by University of Lille. Simone developed her postdoctoral research funded by CAPES (Brazil) – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Proc no. BEX 1338/14-5).

References

1 Yindalon Aphinyanaphongs and Constantin F. Aliferis. Text categorization models for retrieval of high quality articles in internal medicine. In *AMIA Annual Symposium Proceedings*, pages 31–5, 2003.
2 David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
3 David M. Blei and John D. Lafferty. Topic models. In *Text Mining: Classification, Clustering, and Applications*. Chapman & Hall/CRC, 2009.
4 Harris M. Cooper. *The structure of knowledge synthesis*. Knowledge in Society. 1988.
5 Ludovic Denoyer and Patrick Gallinari. Bayesian network model for semi-structured document classification. *Information Processing & Management*, 40(5):807–827, 2004.

6 Tracy Edinger and Aaron M. Cohen. A large-scale analysis of the reasons given for excluding articles that are retrieved by literature search during systematic review. In *AMIA Annual Symposium Proceedings*, pages 379–387, 2013.

7 Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in R. *Journal of Statistical Software*, 25(5):1–54, 2008.

8 Louise Francis and Matthew Flynn. Text mining handbook. In *Casualty Actuarial Society E-Forum*. Casualty Actuarial Society E-Forum, 2010.

9 Carlos A.S.J. Gulo and Thiago R.P.M. Rúbio. Text mining and scientific articles and using the R language. In *Proceedings of the 10th Doctoral Symposium in Informatics Engineering – DSIE*, pages 60–69, Porto, 2015.

10 Andreas Hotho, Andreas Nürnberger, and Gerhard Paab. A brief survey of text mining. *LDV Forum – GLDV Journal for Computational Linguistics and Language Technology*, 2005.

11 D. Isa, L.H. Lee, V. Kallimani, and R. RajKumar. Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1264–1272, 2008.

12 Mohammad S. Khorsheed and Abdulmohsen O. Al-Thubaity. Comparative evaluation of text classification techniques using a large diverse arabic dataset. *Language Resources and Evaluation*, 47(2):513–538, 2013.

13 Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI – International Joint Conference on Artificial Intelligence*, pages 1137–1145. Morgan Kaufmann, 1995.

14 Guy Lebanon, Yi Mao, and Joshua Dillon. The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, 8:2405–2441, 2007.

15 S. Massung, ChengXiang Zhai, and J. Hockenmaier. Structural parse tree features for text representation. In *ICSC – International Conference on Semantic Computing*, pages 9–16, 2013.

16 Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*, 1999.

17 Chitu Okoli and Kira Schabram. A guide to conducting a systematic literature review of information systems research. *Sprouts: Working Papers on Information Systems*, 10(26), 2010.

18 Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

19 M. Mahdi Shafiei and Evangelos E. Milios. A statistical model for topic segmentation and clustering. In Sabine Bergler, editor, *Advances in Artificial Intelligence*, volume 5032, pages 283–295, 2008.

20 Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Knowledge Discovery and Data Mining*, 2011.

21 Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang. *Fundamentals of Predictive Text Mining*. Springer, 2010.

22 Sholom M. Weiss, Nitin Indurkhya, Tong Zhang, and Fred J. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, 2005.

23 Hwanjo Yu, ChengXiang Zhai, and Jiawei Han. Text classification from positive and unlabeled documents. In *Proceedings of the twelfth International Conference on Information and Knowledge Management*, pages 232–239, 2003.

24 Yangchang Zhao. R and data mining: Examples and case studies. In Yangchang Zhao, editor, *R and Data Mining*, pages 1–4. Academic Press, 2013.

Computing Argumentation with Matrices

Evgenios Hadjisoteriou

Department of Computer Science, University of Cyprus
75 Kallipoleos Str., 1678 Nicosia, Cyprus
csp7he2@cs.ucy.ac.cy

Abstract

Abstract argumentation frameworks with finitely many arguments can be presented in matrix form. For this reason, the strengths and weaknesses of matrix operations are migrated from a mathematical representation to a computer science interpretation. We present matrix operation algorithms that can answer whether a given set of arguments is part of an argumentation extension.

1998 ACM Subject Classification I.2.3 Answer/reason extraction

Keywords and phrases Argumentation, Semantics, Extension, Algorithm, Matrix

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.29

1 Introduction and Motivation

Argumentation theory tries to mimic the process of human reasoning. It is often used by agents to reason under specific knowledge, often incomplete, with the alternative to choose. Agents can construct arguments for and against a specific goal, in order to reach a conclusion. The construction of these arguments often follows a semantics, that is given in an abstract argumentation framework either by extensions [5] or by labellings [2, 14]. For what follows we use the former. Many semantics have been established, such as grounded (yields exactly one unique extension), complete, preferred, stable and all of them come with interesting properties [5, 1, 6]. Nowadays, agents perform tasks under incomplete information and dynamic environments, thus decisions must be precise and easily computable. Agents need a tool that is able to produce extensions under the working environment to help them decide what their next move should be.

Finding extensions can be a complex procedure when done without any computational help, when the argumentation framework contains several arguments and attacks. We have developed an algorithm that answers whether a given set of arguments is an extension. Part of this algorithm has been implemented in a system and presented at the ICCMA'15 [1] competition. Our solver, called $A\mathcal{SS}A$ [2] finds the stable extension(s) of an argumentation framework.

In Section 2 we give some basic notions and in Section 3 we present the algorithm tests we perform. A comparison to related work is done in Section 4, and Section 5 concludes with future work.

[1] http://argumentationcompetition.org/index.html
[2] The name was inspired from left and right matrix multiplication: $A\mathcal{S}$ and $\mathcal{S}A$

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 29–36

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Matrices & Argumentation Frameworks

In this section, we present the basic information on matrices and argumentation extensions. We assume that the reader is familiar with basic matrix tools and the fundamentals on argumentation frameworks (see e.g. [11, 5, 1, 6] for more details). A link between the fields of graph theory and logic programs has been presented in [4, 13], where it is shown that stable extensions corresponds to the kernels of the adjacency matrix.

2.1 Matrices

A **matrix** is a structure in rows and columns, where each one of its elements contains information. When the number of rows and columns are equal, the matrix is called **square matrix**. Square matrices have diagonals. A row vector is a $1 \times n$ matrix $\begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$ and a column vector is a $n \times 1$ matrix $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, often written as the transpose matrix $\begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T$.

Computers can perform matrix operations in a relatively fast way. The computational complexity for multiplying two matrices with n digit numbers is $\mathcal{O}(n^3)$ [12]. There are methods that can optimize this result [12, 10]. Our algorithm for computing argumentation extensions and verifying if something is an extension is based on matrix multiplication, thus it is of logarithmic space complexity [8]. In this initial paper, we will not be concerned with complexity issues. It is based on matrices because (a) matrices can be easily represented and handled by a computer, (b) illustrated information is compact, (c) through different operation tools matrices can be easily manipulated and (d) matrices can capture all the information of an abstract argumentation framework.

2.2 Argumentation Frameworks

We review some of the notions introduced by Dung [5] such as acceptable, conflict-free, complete extensions.

▶ **Definition 1** (argumentation framework)**.** An argumentation framework is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation on $\mathcal{A}$ called attack relation.

Square matrices can capture the arguments and the attacking relations of an argumentation framework in a relatively easy way. We first need to label each argument with a distinct natural number and use the rows and columns of the matrix to represent the arguments and the attacks respectively. For example, the third row $a_{3,*}$ of a square matrix consists of the elements $\{a_{3,1}, a_{3,2}, \ldots, a_{3,n}\}$. If argument a_3 attacks argument a_4 then the element $a_{3,4}$ of the square matrix will be one (1), otherwise it will be zero (0). The value of the element $a_{3,3}$ will show if argument a_3 is self attacking.

▶ **Definition 2** (mapping an argumentation framework to a matrix)**.** Let $A = (a_{i,j})$ be the **adjacency matrix** of an argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ such that: $a_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{R} \\ 0 & \text{if } (i,j) \notin \mathcal{R} \end{cases}$

It is important to know who attacks who. The attacker is represented by the row of the adjacency matrix and each column represents the attacked argument. Therefore, $a_{3,4}$ represents the attack from argument a_3 to argument a_4 while $a_{4,3}$ represents the attack from a_4 to a_3.

(a) **(b)**

Figure 1 (a) and (b) show respectively the directed graph and matrix representation of Example 3.

▶ **Example 3.** Let $\{a, b, c\} = \mathcal{A}$ be three arguments such that $\{(a, b), (b, c)\} = \mathcal{R}$. Figure 1 depicts this example.

Matrix operations exist, i.e. multiplication, where when performed on the adjacency matrix of an argumentation framework, an interpretation exists connecting the operating result with the argumentation framework.

▶ **Definition 4** (representing a set of arguments as a column vector)**.** Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework with A its adjacency matrix and $S \subseteq \mathcal{A}$. Set S is represented by a column vector $\mathcal{S}_{n\times 1} = (s_{i,1})$, where $s_{i,1} = \begin{cases} 1 & \text{if } a_i \in S \\ 0 & \text{if } a_i \notin S \end{cases}$

▶ **Proposition 5.** *Let A be the adjacency matrix of an argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ and let $S \subseteq \mathcal{A}$ be a set of arguments with $\mathcal{S}$ its column (resp. $\mathcal{S}^T$ its row) vector representation. The product $A\mathcal{S}$ is a column (resp. $\mathcal{S}^T A$ is a row) vector where the entry $(A\mathcal{S})_{i,1}$ (resp. $(\mathcal{S}^T A)_{1,i}$) shows how many times argument $a_i \in \mathcal{A}$ attacks (resp. is attacked by) S.*

Proof. Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework with n arguments and $A = (a_{i,j})$ its adjacency matrix. Let $S \subseteq \mathcal{A}$ be a set of arguments with $\mathcal{S}$ its column vector representation. The product (of the two matrices A and $\mathcal{S}$) $A\mathcal{S}$ is defined as follows: $(A\mathcal{S})_{i,1} = \sum_{t=1}^{n}((i,t)_{th}$ element of $A \times (t,1)_{th}$ element of $\mathcal{S}) = \sum_{t=1}^{n} a_{i,t}s_{t,1}$. Based on Definition 4 and Definition 2, $(A\mathcal{S})_{i,1}$ is an addition of zeroes if at least one of the entries $a_{i,t}$ or $s_{t,1}$ is zero as $0 \times 1 = 1 \times 0 = 0 \times 0 = 0$ or an addition of ones if both entries $a_{i,t} = s_{t,1} = 1$ since $1 \times 1 = 1$. Intuitively, it is an addition of ones if and only if there exists an attack from a_i to a_t in AF and $a_t \in S$. Similar results hold for $\mathcal{S}^T A$. ◀

3 Algorithms

In this section, we give a computerized method under which given a set of arguments, we can answer whether this set of arguments is conflict-free, admissible, stable, or complete.

3.1 Conflict-free test

Given a set of arguments we can check if this set is conflict-free by running a conflict-free test as follows.

▶ **Definition 6** (conflict-free)**.** A set of arguments S is said to be conflict-free if there are no arguments $a, b \in S$ such that $(a, b) \in \mathcal{R}$.

▶ **Proposition 7** (conflict-free test)**.** *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework and A its adjacency matrix. Let $S \subseteq \mathcal{A}$ be a given set of arguments with $\mathcal{S}$ its column vector representation. Let $\Gamma = \mathcal{S}^T A$. $\mathcal{S}$ passes the conflict-free test if and only if whenever $\gamma_i \neq 0 \in \Gamma$ then $s_i = 0 \in \mathcal{S}$.*

Proof. Based on Proposition 5, $\gamma_i \in \Gamma$ shows how many times argument $a_i \in \mathcal{A}$ is attacked by $\mathcal{S}$. Therefore, when $\gamma_i \neq 0 \in \Gamma$ means that argument a_i is attacked in γ_i ways by the arguments in S. This does not conform with Definition 6. To pass the test arguments that are attacked should not be part of $\mathcal{S}$, thus $s_i = 0 \in \mathcal{S}$. ◀

By constructing a matrix multiplication we can answer if a given set of arguments S is conflict-free. When a row matrix passes (resp. fails) the test we conclude that S is (resp. is not) conflict-free. Note that the empty set always passes the conflict-free test as the generated matrix Γ has zeroes everywhere.

▶ **Example 8.** (a) Consider Example 3 illustrated in Figure 1. Let $S_1 = \{a_1\}$ with $\mathcal{S} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ and $\mathcal{S}^T A = \Gamma = (\,0\;1\;0\,)$. $\gamma_i \neq 0$ when $i = 2$ and $s_2 = 0$. For this reason it passes the conflict-free test. Therefore, the set $S_1 = \{a_1\}$ is conflict-free, i.e. not self attacking. Let us now consider the set $S_2 = \{a_1, a_2\}$ with $\mathcal{S} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ and $\mathcal{S}^T A = \Gamma = (\,0\;1\;1\,)$. $\gamma_2 = s_2 = 1$, thus the set S_2 fails the conflict-free test.

3.2 Admissibility test

To check the admissibility of a given set of arguments S we perform two tests: (a) the conflict-free test and (b) the defending test. For S to be admissible it has to pass both tests.

▶ **Definition 9** (defendable)**.** An argument a is defendable with respect to a set S if and only if each argument attacking a is attacked by an argument in S.

▶ **Proposition 10** (defending test)**.** *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework with adjacency matrix A and let $S \subseteq \mathcal{A}$ be a set of arguments. Let $\mathcal{S}$ be the column vector representation of S and $\Gamma = (\gamma_i) = A\mathcal{S}$. For every i that $\gamma_i \neq 0$ create a column vector $\Delta^{(i)} = (\delta_j^{(i)})$, such that: $\delta_j^{(i)} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$. Set S passes the defending test if and only if at least one of the following holds: (1) Γ is a zero matrix. (2) $E^{(i)} = A\Delta^{(i)}$ and $\forall i, \exists e_k^{(i)} \in E^{(i)}$ such that $e_k^{(i)} \neq 0$ and $0 \neq s_k \in \mathcal{S}$.*

Proof. Based on Definition 9. Matrix $\Gamma = (\gamma_i) = A\mathcal{S}$ shows if S is under attack. When (a) $\Gamma = \mathbf{0}$, S passes the defending test as no attackers exist. When (b) $\Gamma \neq \mathbf{0}$, i.e. attackers exist, we check if S counter attacks them. $\gamma_i \neq 0$ shows the number of attacks but the information who attacks who is lost through the algebraic operations of matrix multiplication. To retrieve this critical information we create $\Delta^{(i)}, \forall i$ that $\gamma_i \neq 0$ and evaluate $E^{(i)} = A\Delta^{(i)}$. $E^{(i)}$ shows if any arguments in A attack $\Delta^{(i)}$. To ensure that when counter attacks exist, they come from arguments in S, we add the restriction $\forall i, \exists e_k^{(i)} \in E^{(i)}$ when $e_k^{(i)} \neq 0$, $s_k \neq 0$, i.e. for every attack there exists at least one argument counter attacking it and this argument belongs to S. ◀

The defending test is based on matrix multiplication. $\Gamma = A\mathcal{S}$ shows if S is under attack, i.e. if any arguments that exist in our argumentation framework can attack S. When $\Gamma = \mathbf{0}$, set S passes the defending test as no attackers exist and there are no arguments to defend against. When $\Gamma \neq \mathbf{0}$, set S is under attack and we have to check if S counter attacks them. Because we do not know who attacks who, instead we get an indication of the number of times each argument in the AF attacks S, we need to separate the arguments under attack and create the vectors $\Delta^{(i)}$. Based on another matrix multiplication, $E^{(i)} = A\Delta^{(i)}$, we find $E^{(i)}$ that shows any arguments in A that attack $\Delta^{(i)}$. To make sure that when these counter

attacks exist, they come from arguments in S, we compare $\mathcal{S}$ with $E^{(i)}, \forall i$. If the comparison shows that any attack is counter attacked by S, we conclude that set S has passed the defending test.

▶ **Example 11** (defending test examples). In Example 3 consider the sets $S^{'} = \{\}$ and $S = \{a, c\}$.

For $S^{'}$: all entries for its column vector representation are zeroes and since $S^{'}$ is represented by a zero matrix, the empty set passes the defending test.

For S: $\mathcal{S} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ and $A\mathcal{S} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \Gamma$. Using $\mathcal{S} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$, we are interested in which arguments can attack arguments a and c. The answer is $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \Gamma$ which means that argument b attacks the argument set $S = \{a, c\}$. From Γ and its non zero entries we find $\Delta^{(1)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$. To check if all attackers (i.e. b) are attacked back by S we evaluate $E^{(1)} = A\Delta^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. The result is $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ and since $e_1^{(1)} \neq 0 \in E^{(1)}$ and $s_1 \neq 0 \in \mathcal{S}$ we conclude that S has passed the defending test.

▶ **Definition 12** (admissible extension). Let S be a conflict-free set of arguments. S is admissible if and only if S is conflict-free and defends itself.

▶ **Proposition 13** (admissibility test). *Let S be a given set of arguments. For S to be admissible its matrix representation $\mathcal{S}$ has to pass both the conflict-free and defending test.*

Proof. The proof follows directly from Definition 12, Proposition 7 and Proposition 10. ◀

3.3 Stable extensions test

▶ **Definition 14** (stable extensions). Let S be a conflict-free set of arguments. S is called a stable extension if and only if every argument not in S is attacked by an argument in S.

Stable extensions are conflict-free and admissible. For any given set S to be stable it first has to pass the conflict-free test. We then want to see if S can attack all other arguments which do not belong to it. This is done with the stable extension test.

▶ **Proposition 15** (stable extensions test). *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework with adjacency matrix A. Let $S \subseteq \mathcal{A}$ be a given set of arguments and $\mathcal{S}$ the column vector of S and $\Gamma = \mathcal{S}^T A$. The set S passes the stable extensions test if and only if:*
1. *$\mathcal{S}$ passes the conflict-free test, and*
2. *$\forall i$ such that $s_i = 0, \gamma_i \neq 0$, ($s_i \in \mathcal{S}$, $\gamma_i \in \Gamma$).*

Proof. Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework, $S \subseteq \mathcal{A}$ be a set of arguments with $\mathcal{S}$ its column vector representation. Based on Definition 14, S should be conflict-free, i.e. it should pass the conflict-free test and every argument not in S should be attacked by an argument in S. $\Gamma = \mathcal{S}^T A$ is a row vector where its entry $(\mathcal{S}^T A)_{i,1}$ shows how many times argument $a_i \in \mathcal{A}$ is attacked by S. Fulfilling the constrain $\forall i$ such that $s_i = 0, \gamma_i \neq 0$ we make sure that every argument not in S should be attacked by an argument in S. ◀

Note that for the stable extension test we do not use the admissibility test. Intuitively, a set attacking anything that is "outside" of it means that it attacks all its potential attackers. This is true since passing the conflict-free test shows that there do not exist attacks coming "inside" of it thus any existing attacks should be from "outside" and it attack them back anyway.

▶ **Example 16** (stable extension test). For Example 3 illustrated in Figure 1, we check if set $S = \{a, c\}$, $\mathcal{S} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ passes the stable extension test. $\mathcal{S}^T A = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} = \Gamma$. First of all it passes the conflict-free test as $\gamma_2 \neq 0$ and $s_2 = 0$. Using $\mathcal{S}^T = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$ we ask which arguments in A are attacked by the set $S = \{a, c\}$. The answer is $\Gamma = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$ which means that it only attacks argument b. Based on the stable extension test, we compare matrices $\Gamma = \mathcal{S}^T A = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$ and $(\mathcal{S})^T = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$. $s_2 = 0$ and $\gamma_2 = 1$ and therefore it is stable.

3.4 Complete extensions test

▶ **Definition 17** (complete extension). An admissible set S of arguments is a complete extension if and only if S contains all arguments it defends.

Let S be a given set. For S to be complete it has to be admissible and for this reason its matrix representation has to first pass the admissibility test. We then have to check if it contains all arguments it defends. Note that if an argument has no attackers, it is trivially defended by any set. This check can be done through two matrix multiplications. The first multiplication $B = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}_{1 \times n} A_{n \times n}$ that finds all arguments that are not under attack and these arguments should be in any complete extension. The second multiplication is to check if the set S contains all arguments it defends. $\Gamma = (\mathcal{S}^T A)A = \mathcal{S}^T A^2$, where $Z = \mathcal{S}^T A$ will show (when mapped to the argumentation framework) all arguments that are under attack by the set S and $\Gamma = ZA$, will show all arguments that set S can defend. To find all complete extensions, we have to run the admissibility test with an extended set S that contains all possible combinations of the arguments in $\mathcal{A}$. This technic is time consuming and its computational complexity exponentially growths as the number of arguments not in S become bigger. To consider all possible combinations with n-many arguments, a matrix with 2^n number of columns has to be created.

▶ **Proposition 18** (complete extensions test). *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework with adjacency matrix $A_{n \times n}$. Let $S \subseteq \mathcal{A}$ with $\mathcal{S}$ its column vector. S passes the complete extension test if and only if:*

1. *$\mathcal{S}$ passes the admissibility test*
2. *Compute $B = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}_{1 \times n} A$ and $\Gamma = \mathcal{S}A^2$. For each entry $b_i = 0$ then $s_i \subseteq S$, and for each entry $\gamma_j \neq 0$ then $s_j \subseteq S$.*

Proof. The proof follows directly from Proposition 13, Definition 9 and Definition 17. ◀

▶ **Example 19.** Consider Example 3, its set $S = \{a, c\}$ and its matrix representation $\mathcal{S}^T = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$. Condition 1 is satisfied as $\mathcal{S}$ passes the admissibility test (see Example 11) and therefore S is admissible. To check if S is complete condition 2 should also be satisfied. Evaluate $B = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$ and $\Gamma = \mathcal{S}^T A^2 = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$. Note that $b_1 = 0$ and $s_1 \subseteq \mathcal{S}$. Additionally, $\gamma_3 \neq 0$ and $s_3 \subseteq \mathcal{S}$. Thus condition 2 is satisfied and $S = \{a, c\}$ is complete.

4 Related work

We have introduced a matrix-based mathematical approach for answering questions of the form: "Is set A an extension?". Similar to this approach, is the work presented in [15]. Their approach is structured as follows: Consider the adjacency matrix of an argumentation framework and then define several parts of the adjacency matrix, which they call sub-blocks. Finding all sub-blocks that have zeroes everywhere, it is like finding the conflict-free sets

of the argumentation framework. Map each one of these sets into its matrix representation norm form, with respect to a specific semantics (stable, admissible, complete). Based on matrix criteria they find among the conflict-free sets which qualify as stable, admissible or complete extensions. Finally, select all extensions passing a given criteria based on matrix operations.

Our approach differs in the fact that we do not use sub-blocks but we also create sets of arguments in a matrix representation to define tests for the different argumentation semantics. Our work when considering all possible sets of arguments, comes close to their work as it answers similar questions but with a different algorithm. Our approach focuses on a given set of arguments as we do not try to find all arguments passing a given criteria.

5 Conclusion & Future work

Given a set S of arguments, we can present it as a matrix and perform different kinds of tests. Based on the results, we can answer whether a set S has a property, i.e. belongs to an argumentation extension of some kind.

In the work of Modgil et all [9] several questions are answered: 1) Does an extension exist? 2) Give an extension. 3) Give all extensions. Let A be a set of arguments. We can answer the question: "Is A an extension?". To find several extensions or all extensions we have to create such a matrix $\mathcal{S}^{'}$ where each one of its columns will represent all possible combination sets that the argumentation framework can define. Then by running tests under $\mathcal{S}^{'}$ we can tackle similar questions by comparing each column of the resulting matrix.

We have implemented a system, called $\mathcal{ASSA}$ that is able to answer different questions for stable semantics for any given abstract argumentation framework. This program creates all possible instances of selected set S into a vector form $\mathcal{S}$. Instead of handling each vector as an individual, it handles them all together when all vectors are combined into a massive matrix $\mathcal{S}^{'}$. By handling $\mathcal{S}^{'}$ in a similar way as $\mathcal{S}$ we manage to find all stable extensions that exist in an abstract argumentation framework. $\mathcal{ASSA}$ at this time is ineffective as it needs to create a massive matrix with 2^n number of columns. We plan to study ways to make this more effective. We also plan to extend our algorithms to cover other extensions (e.g. grounded, ideal) and to extend the $\mathcal{ASSA}$ system to handle these semantics as well.

There is a lot of interest in computing extensions at a point where a competition exist (see ICCMA'15). Other approaches that tackle similar results exist, such as the ASPARTIX (Answer Set Programming Argumentation Reasoning Tool) and DIAMOND [7, 8, 3] but all of them do not use matrices to compute extensions. As a future work, a comparison to these methods in relation to speed and complexity can be studied.

As matrix representation of argumentation frameworks has not received much attentions so far, and our approach constitutes an interesting new research direction, we hope many researches will be inspired and find the content stimulating and thought provoking. We are optimistic that our technique can be used to query graph related problems. Known properties of directed graph can improve our understanding of argumentation extensions since directed graphs allow us to approach the formalizations in a way that ignores the logical meaning and concentrates on their structural properties.

References

1 Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

2 Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

3 Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – A survey. *Artif. Intell.*, 220:28–63, 2015.

4 Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244, 1996.

5 Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

6 Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

7 Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. ASPARTIX: implementing argumentation frameworks using answer-set programming. In *Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings*, pages 734–738, 2008.

8 Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In *Computational Models of Argument – Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, pages 233–240, 2014.

9 S. Modgil and Martin W.A. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G. Simari, editors, *Argumentation in Artif. Intell.*, pages 105–129. Springer Publishing Company, Incorporated, 2009.

10 Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

11 Gilbert W Stewart. *Introduction to matrix computations*. Academic Press, 1973.

12 Andrew James Stothers. *On the complexity of matrix multiplication*. PhD thesis, The University of Edinburgh, 2010.

13 A. Torres. Negation as failure to support. In L. M. Pereira and A. Nerode, editors, *Logic Programming and Non-Monotonic Reasoning: Proc. of the Second International Workshop. Cambridge*, pages 223–243. MIT Press, MA, 1993.

14 Y. Wu and M.W.A. Caminada. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.

15 Yuming Xu. A matrix approach for computing extensions of argumentation frameworks. *CoRR*, abs/1209.1899, 2012.

A Survey of CUDA-based Multidimensional Scaling on GPU Architecture

Hasmik Osipyan[1], Martin Kruliš[2], and Stéphane Marchand-Maillet[3]

1 **National Polytechnic University of Armenia**
Yerevan, Armenia
hasmik.osipyan.external@worldline.com
2 **Charles University in Prague**
Prague, Czech Republic
krulis@ksi.mff.cuni.cz
3 **University of Geneva**
Geneva, Switzerland
stephane.marchand-maillet@unige.ch

Abstract

The need to analyze large amounts of multivariate data raises the fundamental problem of dimensionality reduction which is defined as a process of mapping data from high-dimensional space into low-dimensional. One of the most popular methods for handling this problem is multidimensional scaling. Due to the technological advances, the dimensionality of the input data as well as the amount of processed data is increasing steadily but the requirement of processing these data within a reasonable time frame still remains an open problem. Recent development in graphics hardware allows to perform generic parallel computations on powerful hardware and provides an opportunity to solve many time-constrained problems in both graphical and non-graphical domain. The purpose of this survey is to describe and analyze recent implementations of multidimensional scaling algorithms on graphics processing units and present the applicability of these algorithms on such architectures based on the experimental results which show a decrease of execution time for multi-level approaches.

1998 ACM Subject Classification C.1.2 Multiple Data Stream Architectures (Multiprocessors), D.1.3 Concurrent Programming, I.2.6 Learning, I.3.1 Hardware Architecture

Keywords and phrases Dimensionality Reduction, Pattern Recognition, Graphics Hardware, Compute Unified Device Architecture, Parallel Programming

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.37

1 Introduction

In different pattern recognition problems, a large amount of measurements is obtained from hardware devices. Applying clustering methods or classifiers to these measurements is problematic due to the parameter estimation problem which arises when learning methods applied to high-dimensional data sets with a limited number of samples. In order to handle these problems, the dimensionality reduction techniques are used which helped to extract a small number of useful features from the set of many measurements [1].

High-dimensional data sets exist in most scientific domains such as visualization [4], classification [5], recommender systems/filtering [6, 7], signal/image processing [8, 9], etc. That is why the contributions to this field have come from many disciplines and approaches to the problem are quite different. In the recent years, many different linear and non-linear

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 37–45

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

dimensionality reduction techniques such as multidimensional scaling (MDS) [14], t-SNE algorithm [2] and locally linear embedding (LLE) [3] have been developed for using in different applications. But the main idea behind all these dimensionality reduction techniques is the same: transformation of high n-dimensional input data $X = x_1, x_2, ..., x_n$ into a relevant k-dimensional space $Y = y_1, y_2, ..., y_k$, where $k \leq n$ (and possibly $k << n$). Based on the existing implementations/adaptations of these techniques on multi-core architectures and due to a limited space, in this paper we will only focus on the MDS algorithm. Albeit an early dimensionality reduction method, the MDS is still very popular in the implementation of interactive visualization technique in the context of visual analytics [29] triggered by the need to inspect Big Data visually.

Unfortunately, classical MDS quickly became quite expensive for high-dimensional data sets, where both the number of variables and samples are high. Therefore, there is a real need to reduce execution time of these algorithms by suggesting a faster algorithms or implementing existing ones on parallel architectures such as graphics processing units (GPUs).

With the evolution of the NVIDIA CUDA API (Compute Unified Device Architecture Application Programming Interface) [15], GPU hardware becomes more power-efficient than central processing units (CPUs). Unlike CPUs, GPUs give the low cost alternative for implementing non-graphical problems in parallel. However, achieving a good utilization of GPU requires not only careful implementation which needs detailed understanding of the underlying architecture, but also the right usage of all the capabilities provided by these units.

In this survey, we review the multidimensional scaling – a more recent state-of-the-art dimensionality reduction technique – and discuss its adaptation on the GPU architecture. To analyze the performance issues of MDS algorithms in a good way we have also run experiments for the same hardware and data set. Earlier survey papers [10, 11, 12, 13] review dimensionality reduction methods on classical CPUs and, to the best of our knowledge, this is the first survey paper which reviews the adaptation of MDS algorithms on the GPU architecture.

The rest of this paper is organized as follows. Sections 2 briefly reviews the MDS algorithm. Section 3 introduces the fundamentals of GPU architecture as well as the parallel programming model adapted to program using the all advantages of these units. Section 4 shows the adaptation of MDS methods on the GPU. Finally, we discuss the results based on the experiments in Section 5 and conclude in Section 6.

2 Multidimensional Scaling

In this section we review the classical MDS and emphasize the main computationally intensive components of the algorithm. Classical MDS algorithms take an $n \times n$ square matrix D containing all possible dissimilarities between n data objects and map these dissimilarities into a lower dimensional Euclidean space. The goal of MDS is to minimize a loss function which measures the lack of fit between the distances of the lower dimensional objects $\|x_i - x_j\|$ (where $\|.\|$ is the Euclidean distance) and the dissimilarities d_{ij} of the full dimensional data objects (Eq. 1) .

$$\sum_{i,j}(\|x_i - x_j\| - d_{ij})^2 \rightarrow min \tag{1}$$

In order to compute classical MDS, the input matrix D is first converted into a dot product matrix (Eq. 2).

$$J = I - \frac{1}{n}[1], B = -\frac{1}{2}JD^2J \tag{2}$$

Then, the m largest positive eigenvalues $\gamma_1, \ldots, \gamma_m$ and eigenvectors $\varepsilon_1, \ldots, \varepsilon_m$ of B matrix are extracted from which the final low-dimensional data set $x_1, x_2, ..., x_n$ is computed as

$$X = E_m\sqrt{\gamma_m} \tag{3}$$

where γ_m is the diagonal matrix of the m eigenvalues and E_m is the matrix of m eigenvectors.

Solving the eigenpairs problem is the main bottleneck of the original MDS algorithm in terms of execution time. The demand of working with large scale data sets creates a need for a new algorithms with reduced time complexity. For example, Morrison [16] suggested a new hybrid approach which is based on sampling, interpolation, and the use of spring models [17]. They showed the effectiveness of their approach by reducing the computational complexity from $O(n^2)$ to $O(n\sqrt{n})$. In another work of Yang et al. [18], the authors presented new sampling-based fast approximation method by dividing the original matrix into sub-matrices and then combining sub-solutions. For his part, this method reduced the time complexity to $O(n \log n)$. But even with these optimizations, more rapid processing (e.g. parallel) is required when the number of input objects reaches the order of millions.

Besides the classical MDS algorithm, other types such as metric MDS, non-metric MDS, or generalized MDS exists. Furthermore, there are various approaches to implement a given MDS problem. For example, conventional methods for classical scaling can be solved by eigendecomposition, while the distance scaling methods demand iterative minimization which is much more expensive in terms of execution time [10, 11, 12, 13]. The methods which have proven to be applicable for GPU architectures are compared in Section 4.

3 GPU Fundamentals

GPU architectures differ from CPUs in multiple ways. In this section, we present the GPU architecture fundamentals with particular emphasis on aspects, which have a great importance in the light of the studied problem.

A GPU card is a peripheral device with its own memory which is connected to the host system via the PCI-Express (PCIe) bus. It has several *streaming multiprocessor* units (SMPs), which share only the main memory bus and the L2 cache of the GPU.

The GPU architecture follows the Single Instruction Multiple Thread (SIMT) execution model by processing multiple data items simultaneously by the same function called *kernel*. When a kernel is invoked, the caller specifies, how many threads are attached to this kernel. Each thread executes the kernel code, but has an unique thread ID, which is used to identify the portion of the work processed by the thread.

Threads are grouped together into blocks of the same size. Threads from different blocks run independently as they are usually assigned to different SMPs. On the other hand, threads within one block may synchronize their work efficiently using barriers and closely cooperate using internal resources of the SMP, especially the shared memory. Threads in a block are divided into subgroups called *warps* (all current NVIDIA architectures use 32 threads per warp) and they are executed simultaneously in a lockstep, which means they are all issued the same instruction at a time.

GPU hardware has its own memory hierarchy which is shown in Figure 1. The *host memory* is the operational memory of the computer which directly cannot be accessed by

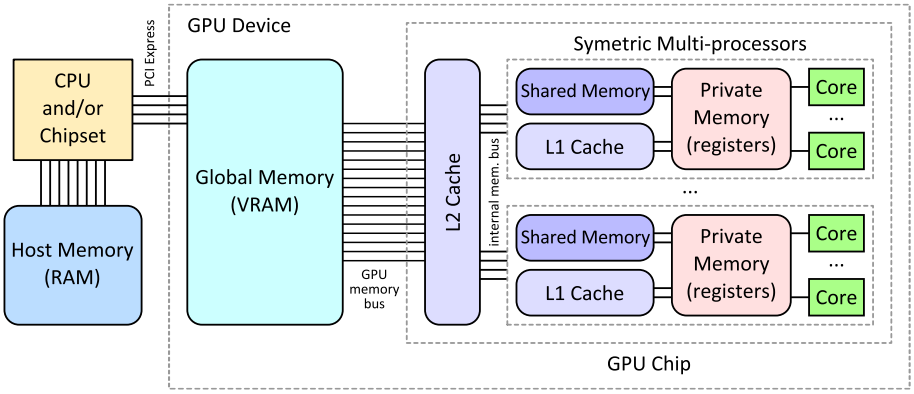


Figure 1 Host and GPU memory organization scheme.

the GPU device. Before execution, the input data need to be transferred from the host memory (RAM) to the graphic device global memory (VRAM). Data are transferred via the PCI-Express bus which is rather slow (8 GB/s) in comparison to the internal memory buses.

The *global memory* can be accessed from the GPU cores, so the input data and the results computed by a kernel are stored here. The *shared memory* is shared among threads within one group. It is rather small (tens of KB) but almost as fast as the GPU registers which makes it quite effective for intermediate results exchange between the threads in the block. The *private memory* belongs exclusively to a single thread and corresponds to the GPU core registers. Its size is very limited (tens to hundreds of words per thread), so it is suitable only for a few local variables. Furthermore, a GPU employs two level of caches to reduce the latency of memory transfers. The *L2 cache* is shared by all SMPs and caches all access to global memory. The *L1 cache* is private to each SMP and caches data from global memory selectively by a specialized loading instruction.

One of the critical aspects of developing GPU algorithms is the thread synchronization and data exchange. These operations disrupt the work flow of individual threads and reduce the utilization of GPU cores. Hence, they can severely limit the performance of any parallel algorithm.

4 MDS Approaches Utilizing GPU Architectures

In this section, we are going to review the most recent approaches of MDS algorithm on GPUs by highlighting the main performance issues.

4.1 Single-level Approaches

Fester et al. [19] proposed a CUDA implementation of MDS algorithm based on the high-throughput multidimensional scaling (HiT-MDS) [20]. As the time consuming part of the HiT-MDS algorithm is the computation of Pearson correlation coefficient, the steps of HiT-MDS which implemented on the GPU are as follows:

1. Partial-sum calculation – time complexity of this task for parallel implementation is $O(\log n)$ performing only $O(n)$ addition operations [21]. To use all the advantages of CUDA parallel model, authors used the shared memory of SMPs to load data portion by portion and compute partial-sum by threads within the block.

2. Euclidean distance calculation – as this operation should be done for each data point it best suits on the GPU architecture when each thread is responsible for calculating the distance of one point.
3. Floyd-Warshall algorithm – an additional operation for getting information out of sparse graphs. It is based on the work of Harish [22] and uses the texture memory access method.
4. Degree based distance manipulation – also an additional operation for visualizing network structures by computing the distances out of adjacency. In this step, CUDA was used only for visualization of the node positions.

In another approach [25], authors suggest a new efficient parallel algorithm for MDS mapping based on virtual particle dynamics (VPD-MDS) [26] and demonstrated the performance of this algorithm on the GPU architecture. The VPD-MDS uses cost function converted into a more general form (Eq. 4).

$$V(X) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} V(\|D_{ij}^k - d_{ij}^k\|) \tag{4}$$

The $V()$ is the difference of distances d_{ij} and respective dissimilarities D_{ij} of points in X where the points interact with each other via partial forces $f_{ij} = - \bigtriangledown V(\|D_{ij} - d_{ij}\|)$. As the most expensive part of this algorithm in terms of complexity is the computation of the partial forces ($O(M^2)$), in the paper authors concentrated on the parallelization of that procedure.

The implementation computes only the partitioning of the data and the calculation of the partial forces on the GPU. In addition, they utilize asynchronous data transfers, which means that the read/write operations from/to global memory overlapped with the calculations. One of the greatest challenges solved by this implementation was the problem of the size of the dissimilarity matrix which may not fit the (rather limited) memory of the GPU. To overcome this problem, authors proposed a parallel implementation using MPI (Message Passing Interface) library on a cluster where the data can be distributed among several GPUs and processors.

4.2 Multi-level Approaches

Though the iterative approaches of MDS algorithm are more flexible because they find a stress-optimal configuration, they do not always guarantee an optimal solution due to the difficulty of finding an appropriate termination point. To overcome this problem, the Glimmer algorithm [23] for the GPU architectures was proposed. It divides the input data into hierarchical levels and executes the algorithm recursively. The method is based on a stochastic approach [17].

In brevity, the algorithm extracts subsets from the input data where each level is obtained from the parent level (*restriction*). The stochastic force algorithm is performed on each level starting at the bottom (*relaxation*) and the results of each level are interpolated up to the parent level (*interpolation*). The algorithm finishes when the stochastic force algorithm have processed the highest level of the hierarchy which contains all points of the data set.

The restriction procedure randomly splits data into the levels and due to the inefficient implementation on the GPU, this portion kept on the CPU side. During stochastic force, a set of *near points* and a set of *random points* are defined for every data point. These sets are filled randomly at the beginning and after each iteration, the sets are updated according to the distances between data points. The stress error metric, which is used in the termination condition, is defined to approximate $\epsilon = \frac{1}{10000}$ value.

After the restriction procedure, the GPU stochastic force algorithm (GPU-SF) keeps the data in the texture memory. At the beginning, a random index set index is defined for each data point and a set of high and low dimensional (Euclidean) distances is calculated to simplify the subsequent updates of the near set index. In the final step, a proper force is calculated and each point is shifted to according to that force. In calculation of the Euclidean distances the sum of the values was implemented by a parallel reduction algorithm and the near sort by even-odd sorting network.

Another recent technique showed CUDA-based fast multidimensional scaling (CFMDS) approximate solution [24]. It implements two approaches: If the data can fit the memory of the GPU entirely, a classical algorithm is executed. Otherwise, the input data are divided into smaller portions that fit the global memory. The required memory is computed at the beginning and the program selects appropriate version of the MDS.

For the partial MDS the sampling of the input data was done in two ways: randomly and by MaxMin approach. During the MaxMin approach, data points were chosen one at a time and each new point maximized the minimal distance to any of the previously sampled points.

5 Discussion

To better analyze the performance issues for each algorithm (HiT-MDS, VPD-MDS, Glimmer, and CFMDS) we present the results obtained on the same GPU hardware for the same data sets.

Due to a limited space, we present only the results for fixed set of parameters. The results of each presented algorithm were measured for three data sets of sizes 0.2m, 0.5m, and 1m (m is a million) extracted from CoPHIR data set [27] which consists of 106-million MPEG-7 global visual descriptors. The descriptors are scalable color, color structure, color layout, edge histogram, and homogeneous texture. The vectors in each data set are constructed by combining the first three descriptors, which result in vectors of size 208-dimension.

Our experiments were conducted on a PC with an Intel Core i3-4010U CPU clocked at 1.7 GHz, which have 4 physical cores and 4 GB of RAM. The desktop PC is equipped with NVIDIA GeForce GT 740M (Kepler architecture [28]), which have 2 SMPs comprising 192 cores each (384 cores total) and 4 GB of global memory. The host used Windows 8.1 as operating system and CUDA 5.1 framework for the GPGPU computations.

The experiments were timed using the real-time clock of the operating system. Each experiment was conducted 10× and the arithmetic average of the measured values is presented as the result. All measured values were within 1% deviation from their respective averages. The comparison is mainly based on the speedup of the algorithms, since the accuracy of the results were presented in the related papers.

Due to a limited hardware availability in our experiments for VPD-MDS method data were not distributed among several GPUs. Instead, the data were loaded to global memory of the GPU portion by portion which significantly affected the final results. The portions of the work could have been processed on several GPUs in some cases to increase the overall speed; however, the main objective of this survey is to compare the methods relatively.

Table 1 shows the execution time (seconds) of each presented algorithm on the GPU using different thread block configurations. In case of HiT-MDS algorithm, the Floyd-Warshall algorithm and the degree based distance manipulation points (step 3 and 4) are not included in the final results. Furthermore, we include only the MaxMin results of the CFMDS implementation, since it has proven to be faster than the random method.

We have run experiments for different block sizes (2, 4, 8, 16, 32 warps – i.e., 64 – 1,024

Table 1 Performance comparison of presented GPU implementations.

Warps	Glimmer			CFMDS			HiT-MDS			VPD-MDS		
	0.2m	**0.5m**	**1m**	**0.2m**	**0.5m**	**1m**	**0.2m**	**0.5m**	**1m**	**0.2m**	**0.5m**	**1m**
2	1032	1615	2375	1051	1465	2817	1118	1652	3005	1362	1589	3206
4	512	742	1297	564	680	1422	524	755	1503	604	795	1597
8	410	602	1231	422	620	1404	416	699	1405	565	759	1540
16	417	594	1220	425	619	1394	421	711	1400	557	784	1563
32	**416**	**597**	**1211**	**415**	**620**	**1390**	**426**	**694**	**1401**	**557**	**754**	**1541**

threads) to analyze the effect of the GPU configuration on the performance of the algorithms. On the Kepler architecture, at least 4 warps must be used in order to achieve full occupancy of the cores, no matter which algorithm was tested. Due to a limited space, we analyze only the results when 32 warps (1,024 threads) were used, since these results are the best (or very close to the best) for every algorithm.

The results indicate that the best performance was achieved with the Glimmer algorithm. It can be explained by the fact that stochastic approach is very well suited for the GPU architectures, because each point of the data set references only a small set of the other points during the main computation. Furthermore, the Glimmer algorithm spends the most time performing the relaxation procedure, which also performs well on GPUs. In terms of execution time, CFMDS algorithm is quite competitive with Glimmer algorithm, which can be explained by the multilevel origin of the two methods. CFMDS gives a little bit slower results, since the classical MDS algorithm is performed for each level of CFMDS and it is less suited for GPUs than the stochastic approach. Finally, the HiT-MDS solution outperformed VPD-MDS method because of the shared memory usage for partial-sum calculation.

6 Conclusion

Being a most popular method of dimensionality reduction, MDS algorithm is a main research topic in different science fields. Due to the rapid increase of the number of processed data, utilization of massively parallel hardware such as GPUs becomes a necessity. We have analyzed the performance issues of existing MDS algorithms on GPU architecture and compared their performance.

Comparison of existing works reveals that GPU solutions showed a promising potential for future scaling. Meanwhile, the differences in execution times of presented implementations are not very significant due to the similarity of methods. We believe that the effectiveness of each described method strongly depends on the used data set and the application, which means that the provided results could be roughly different for other inputs. However, comparison shows that a multi-level approaches show better results for large scale data.

The nature of dimensionality reduction algorithms allows adaptation of these methods for heterogeneous system consisting of multiple GPUs and multi-core CPUs which can significantly improve the performance even further. Based on the performance results obtained for the Glimmer algorithm, we are planing to use this method to visualize Maya hieroglyphs on a multi-GPU system.

Acknowledgements. This paper was supported by the Swiss National Foundation (SNF) under the project MAAYA (grant number 144238) and by Czech Science Foundation (GAČR) project no. P103-14-14292P.

References

1 C. J. C. Burges. *Dimension Reduction: A Guided Tour*. Foundations and Trends in Machine Learning 2, 2010

2 L. J. P. van der Maaten and G. E. Hinton. *Visualizing High-Dimensional Data Using t-SNE*. Journal of Machine Learning Research, 9(Nov), pp. 2579–2605, 2008.

3 S. Roweis and L. Saul. *Nonlinear dimensionality reduction by locally linear embedding*. Science, 290(5500):2323–2326, 2000.

4 D. Engel, L. Huttenberger, and B. Hamann. *A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization*. Proceedings of IRTG 1131 Workshop, Dagstuhl, Germany, pp. 135–149, 2012.

5 F. Plastria, S. de Bruyne, and E. Carrizosa. *Dimensionality Reduction for Classification*. In ADMA (Lecture Notes in Computer Science), Springer, 411–418, 2008.

6 B. M. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl. *Application of Dimensionality Reduction in Recommender System – A Case Study*. In ACM WEBKDD Workshop, 2000.

7 P. Imkeller, S. Namachchivaya, N. Perkowski, and H. C. Yeong. *Dimensional Reduction in Nonlinear Filtering: a Homogenization Approach*. Timo Sepplinen (Ed.), Vol. 23, The Annals of Applied Probability, 2013.

8 M. Ishteva, L. de Lathauwer, P. A. Absil, and S. van Huffel. *Dimensionality reduction for higher-order tensors: algorithms and applications*. International Journal of Pure and Applied Mathematics 42, 337–343, 2008.

9 I.F.L. Journaux and P. Gouton. *Multispectral Satellite Images Processing through Dimensionality Reduction*. In Signal Processing for Image Enhancement and Multimedia Processing, 2008.

10 I. K. Fodor. *A survey of dimension reduction techniques*. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, USA, 2002.

11 L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. *Dimensionality Reduction: A Comparative Review*. Technical Report TiCC-TR, Tilburg University, 2009.

12 N. Varghese, V. Verghese, P. Gayathri, and N. Jaisankar. *A survey of dimensionality reduction and classification methods*. International Journal of Computer Science and Engineering Survey (IJCSES) 3, pp. 45–54, 2012.

13 A. Sarveniazi. *An actual survey of dimensionality reduction*. American Journal of Computational Mathematics 4, pp. 55–72, 2014.

14 W. S. Torgerson. *Multidimensional scaling: I. Theory and method*. Psychometrika 17, pp. 401–419, 1952

15 NVIDIA Corporation. *NVIDIA CUDA: Compute Unified Device Architecture. Reference Manual*. NVIDIA Corporation, 2701 San Tomas Expressway, Santa Clara, 2008.

16 A. Morrison, G. Ross, and M. Chalmers. *Fast multidimensional scaling through sampling, springs and interpolation*. Information Visualization 2, pp. 68–77, 2003.

17 M. Chalmers. *A Linear Iteration Time Layout Algorithm for Visualising High-dimensional Data*. In Proceedings of the 7th Conference on Visualization. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.

18 T. Yang, J. Liu, L. McMillan, and W. Wang. *A fast approximation to multidimensional scaling*. In Proceedings of the ECCV Workshop on Computation Intensive Methods for Computer Vision, pp. 354–359, 2006.

19 T. Fester, F. Schreiber, and M. Strickert. *CUDA-based Multi-core Implementation of MDS-based Bioinformatics Algorithms*. In GCB (LNI), Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, and Peter F. Stadler (Eds.), Vol. 157, 67–79, 2009

20 M. Strickert, S. Teichmann, N. Sreenivasulu, and U. Seiffert. *High-Throughput Multidimensional Scaling (HiT-MDS) for cDNA-Array Expression Data*. In ICANN (1) (Lecture Notes in Computer Science), Wlodzislaw Duch, Janusz Kacprzyk, Erkki Oja, and Slawomir Zadrozny (Eds.), Vol. 3696. Springer, pp. 625–633, 2005.

21 M. Harris, S. Sengupta, and J.D. Owens. *Parallel Prefix Sum (Scan) with CUDA*. In GPU Gems 3, 2007.

22 P. Harish and P. J. Narayanan. *Accelerating Large Graph Algorithms on the GPU Using CUDA*. In Proceedings of the 14th International Conference on High Performance Computing. Springer-Verlag, Berlin, Heidelberg, pp. 197–208, 2007.

23 S. Ingram, T. Munzner, and M. Olano. *Glimmer: Multilevel MDS on the GPU*. IEEE Trans. Vis. Comput. Graph. 15, pp. 249–261, 2009.

24 S. Park, S. Y. Shin, and K. B. Hwang. *CFMDS: CUDA-based fast multidimensional scaling for genome-scale data*. BMC Bioinformatics 13, S-17, S23, 2012.

25 P. Pawliczek, W. Dzwinel, and D. A. Yuen. *Visual exploration of data by using multidimensional scaling on multicore CPU, GPU, and MPI cluster*. 2014.

26 W. Dzwinel and J. Blasiak. *Method of particles in visual clustering of multi-dimensional and large data sets*. Future Generation Comp. Syst. 15, pp. 365–379, 1999.

27 P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. *CoPhIR: a test collection for content-based image retrieval*. CoRR, 2009.

28 NVIDIA. *Kepler GPU Architecture*. `http://www.nvidia.com/object/nvidia-kepler.html`.

29 D. Keim, G. Andrienko, J. D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual Analytics: Definition, Process, and Challenges*. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North (Eds.), Information Visualization – Human-Centered Issues and Perspectives, pp. 154–175, Lecture Notes in Computer Science 4950, Springer Berlin Heidelberg. (2008).

Modelling SO-CAL in an Inheritance-based Sentiment Analysis Framework

F. Sharmila Satthar

University of Brighton
Watts Building, Lewes Road, Brighton, UK
F.Satthar@brighton.ac.uk

Abstract

Sentiment analysis is the computational study of people's opinions, as expressed in text. This is an active area of research in Natural Language Processing with many applications in social media. There are two main approaches to sentiment analysis: machine learning and lexicon-based. The machine learning approach uses statistical modelling techniques, whereas the lexicon-based approach uses 'sentiment lexicons' containing explicit sentiment values for individual words to calculate sentiment scores for documents. In this paper we present a novel method for modelling lexicon-based sentiment analysis using a lexical inheritance network. Further, we present a case study of applying inheritance-based modelling to an existing sentiment analysis system as proof of concept, before developing the ideas further in future work.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases Sentiment analysis, NLP, Inheritance network, Lexicon-based

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.46

1 Introduction

Sentiment analysis is the computational study of people's opinions, as expressed in text. It is important when a company or service provider wants to understand their users' needs, or share users' opinions and reviews about products or services with other potential users [8] [6]. A commonly used way of detecting sentiment is to caluclate 'semantic orientation' (SO), a numeric measure of subjectivity and opinion in text, for example in film reviews [13]. In machine learning approaches to sentiment analysis, semantic orientation scores are learned using statistical modelling from prepared training data. In lexicon-based approaches, semantic orientation scores are associated with individual words (such as +3 for 'good', −3 for 'bad'), and the total score for a text is calculated using heuristic rules. Key advantages of the lexicon-based approaches is that they do not require preparation of extensive training data sets, and their heuristic rules can utilize linguistic context to determine the sentiment of complex constructions, for example valence shifters [9], [5] such as intensifiers or negators. The major challenge for lexicon-based methods is coverage – handling words that are not in the lexicon or constructions that were not predicted by the rule designers.

One branch of previous research in natural language lexicons makes use of non-monotonic (default) inheritance networks to represent lexical information [1]. Regular and irregular words can be represented in a hierarchical structure with abstraction that shares common properties and behaviours, but also allows irregular words to specify only those aspects that deviate from the regular case. Our intuition is that lexicon-based sentiment analysis systems can be made more accurate by using such default inheritance-based lexical knowledge representation, and that this approach will allow us to address some of the coverage limitations of previous approaches. As a first step towards this goal, in this paper we present our work on modelling

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 46–53

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

an existing lexicon-based approach to sentiment analysis in an inheritance-based framework using the lexical representation language DATR [4].

The system we modelled was Taboada el al.'s 'semantic orientation calculator' (SO-CAL) [11] [12] [10]. In SO-CAL, sentiment is represented by the semantic orientation of the text, which is expressed as both the word's (semantic) polarity and its strength (intensity). So, a semantic orientation score of a word/text determines whether it is positive or negative depending on its sign and how strong it is depending on its magnitude. SO-CAL uses a pure lexical method in which they calculate semantic orientation of a text by aggregating the semantic orientation of each opinion word present in the text, applying various heuristic rules to take account of contextual constructions.

In section 2, we briefly describe SO-CAL and the features used in its heuristics. In section 3, we describe the inheritance-based framework and our sentiment analysis system, Galadriel. In section 4, we describe how we model SO-CAL in Galadriel and in section 5, we present an evaluation that shows how Galadriel's performance compares with SO-CAL. Finally, section 6 contains discussion and future work.

2 The SO-CAL system

In SO-CAL [11], Taboada et al. aimed to analyse semantic orientation of individual words and contextual valence shifters in depth. However, they did not focus on linguistic analysis. First they extracted sentiment-bearing words including adjectives, adverbs, nouns and verb in a document. Then they used the semantic orientation score for each word from semantic orientation dictionaries to calculate a score for the whole document, taking into account valence shifters such as intensifiers and negators. Semantic orientation dictionaries are special dictionaries which include words with their semantic orientation. Taboada et al. created their dictionaries manually, as they believed that the way of creating dictionaries affects the overall accuracy of final results. Therefore, as a first step they created dictionaries for the words, which contain adjectives, adverbs, verbs and nouns with sentiment scores between $+5$ and -5 ($+$ sign refers to the positive polarity and $-$ sign refers to the negative polarity, and a semantically neutral word has a zero score).

To obtain the semantic score for a given document, SO-CAL calculates the overall SO value by adding together the semantic scores of words present in the document. In addition, for various classes of words, rules are invoked to modify the SO scores. For example, intensifier words modify the SO score of the word they are attached to (eg 'very good'): the SO dictionary specifies how big this modification is for each intensifier as a percentage (so 'very' involves a bigger change than 'slightly'). As well as intensifiers, SO-CAL has rules for negators, irrealis (hypothetical statements), repetition and positive bias which are explained in more detail below. This approach makes two key assumptions: the semantic orientation of a word is independent of its (broader) context and the semantic orientation can be expressed in numerical value.

3 The Inheritance-based Framework

In order to model SO-CAL, we start out with the 'Galadriel', which is a sentiment framework using the DATR/ELF lexicon representation system. DATR/ELF is a default inheritance based language processing system. DATR/ELF can encode very complex lexical information relating to phonology, morphology, syntax and semantics. Our research aims to exploit this information for sentiment analysis.

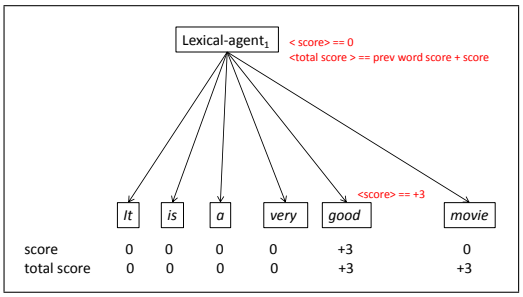


Figure 1 Simple sentiment model: add up raw sentiment score of all words.

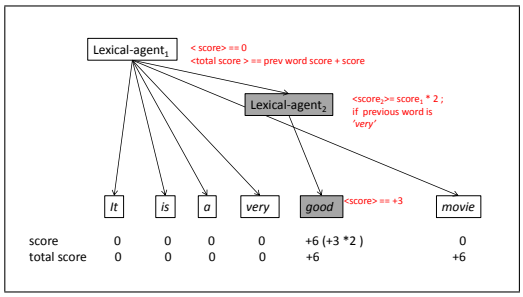


Figure 2 Sentiment model with intensifiers: *'very'* changes sentiment score of following word.

3.1 DATR and ELF

Evans and Gazdar [4] designed a lexical description language, DATR, to model the structure of the lexicon using default inheritance to capture complex class, subclass and exception relationships between words. More recently, Evans [3] introduced the Extended Lexicon Framework (ELF), a development which uses DATR to represent words not as isolated individuals, but as instances occurring in sentences. In ELF, information is still represented on a word-by-word basis, but the information about a word can depend upon information about its neighbours in a sentence. This allows ELF to represent more complex properties of whole sentences, while retaining its default lexical character to express exceptional cases.

ELF is based on two core ideas: the first is to view each word as a 'lexical agent', containing fixed information about the word itself, represented as features with values, and rules for calculating more complex values. These rules can refer to other features of the word, but also to features of adjacent words when the word appears in a sentence. For example the lexical agent for 'a' can look at the word to its right to decide whether its form feature should be 'a' or 'an'. The second is that these specifications of values and rules for lexical agents are organised into a default inheritance hierarchy, so that words with similar behaviour share the rules defining that behaviour.

3.2 The Basic Galadriel System

The basic Galadriel system uses ELF lexical agents to implement simple semantic orientation calculation. In Figure 1, each word is a lexical agent which has two features – score and total. All the lexical agents for actual words inherit from an abstract lexical agent node called lexical-agent$_1$. This nodes specifies a value for score of 0 (neutral) and a rules for calculating the total, by adding the score to 'prev total' – the total from the previous word. All the word nodes inherit both these specifications, except the word 'good' which specifies its own score of +3, overriding the (default) inheritance from lexical-agent$_1$. The resulting values for score and total are shown in the figure, and the SO score for the whole sentence can be read off from the value of the total feature for the last word.

In Figure 2, we extend this model with an another agent, lexical-agent$_2$, which describes a rule for intensifiers. This rule says that if the previous word is 'very', then this word's sentiment score has to be multiplied by a factor of 2. In this example, lexical-agent$_2$ is only used for sentiment-bearing words, such as 'good' – neutral words just inherit from lexical-agent$_1$ as before. Therefore, the sentiment score of 'good' changes and all other words' scores remain as before.

4 Modelling SO-CAL in Galadriel

In order to test out the Galadriel system architecture, we aimed to model SO-CAL in Galadriel. In this section we provide the key steps of the modelling process. We ended up creating a total of 6 models in Galadriel for SO-CAL features. Each model is used to capture one feature of SO-CAL. In Galadriel, we named the models as sent1, sent2 and so on.

4.1 Model sent1: Aggregating SO scores

We have four different dictionaries (used for SO-CAL) for the parts of speech adjectives, adverbs, nouns and verbs with their SO values (between +5 and-5). As discussed above, in order to get total SO value of document, SO-CAL aggregates the SO value of each word present in the document. In Galadriel, model sent1 is a simple model where each word has associated with it its own SO score and a total score for the document up to that point. This is as show in Figure 1, above, except that the SO scores come from the dictionaries, rather than being explicitly specified.

4.2 Model sent2: Intensification

Intensifiers do not contribute propositional meaning of a clause, and they generally do not have any sentiment of their own. But they give additional emotional context to a word they modify, which means intensifiers change the semantic intensity of that word. The words whose SO values are being modified by intensifiers are usually their neighbouring lexical item. Taboada et al. represented value of an intensifier as percentage, and these values are listed in the SO-CAL dictionaries. Figure 3 shows our modelling of intensifiers, which uses the same approach as in Figure 2, but allowing for different intensification factors (from the dictionaries), and making more explicit the inheritance between models sent2 and sent1.

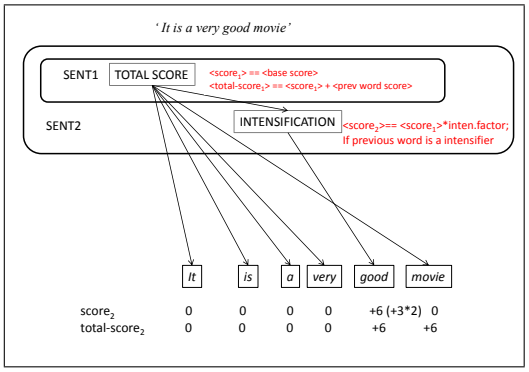


Figure 3 Model sent2 for intensifiers inheriting from model sent1.

4.3 Model sent3: Negation

Negation words usually reverse the opinion of a sentence. Two methods are applied for dealing with negators. They are the switch negation method, where the polarity of the lexical item next to negator will be switched, and the shift negation method, where the SO value of aword which needs to be negated is shifted towards the opposite polarity by a fixed amount. Negation words include 'not','never','no','nobody'. Similar to intensifiers, negators do not have SO values themselves and so are categorised as neutral. Taboata et al. defined any negator as negating the opinion expressed within the same clause. In order to identify a clause or sentence, a list of end punctuation words is created. This allows the identification of clauses and sentences in a document. Moreover, Taboada et al. argued that the switch negation does not work in certain cases. Therefore they implemented the shift negation method. They introduced a constant number 4 and instead of changing the sign they shifted SO value toward the opposite polarity by the constant 4.

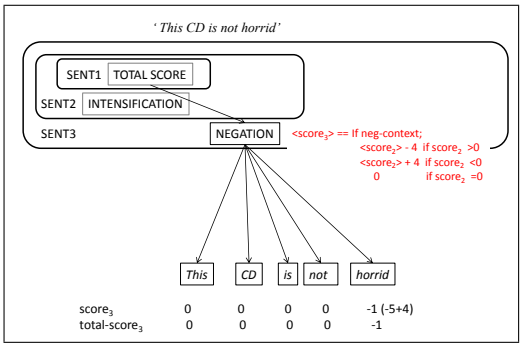


Figure 4 Model sent3 for negation:neg-context adjust the sentiment score.

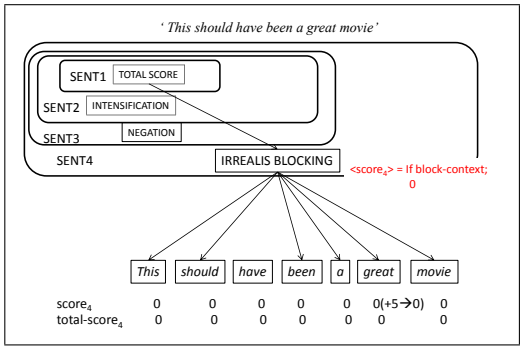


Figure 5 Model sent4:block-context changes sentiment scores to 0.

To model negation in Galadriel, first clauses and sentences are identified. Then any negation words within a clause/sentence negates opinion expressed within the same clause or sentence. In this model a new feature called 'neg-context' is introduced for each and every word in the document. The feature 'neg-context' takes the value either 'yes' or 'no'. Any word which could be negated by a negator, is assigned a 'neg-context' value as 'yes' otherwise 'no'. Finally, following SO-CAL, the shift negation rule is applied to the words which have a neg-context value of yes. (See Figure 4.)

4.4 Model sent4: Irrealis Blocking

Taboada et al. identified a number of irrealis markers which introduce non-factual context. Such markers indicate the words appearing in a clause/sentence are not reliable for the purpose of sentiment analysis. These words change the meaning of sentiment-bearing words and such words are named 'irrealis markers'. Their list of irrealis markers includes conditional markers ('if'), certain verbs, ('doubt', 'expect'), negative polarity items, words enclosed in quotes and questions.

In Galadriel, Taboada et al.'s list of irrealis marker is categorized under a hierarchical lexical node called 'mark'. To model SO-CAL's irrealis blocking feature in Galadriel, a new feature called 'block-context' with possible values 'yes' or 'no' is introduced. Similar to model sent3, the 'block-context' feature also uses end punctuation words to assign its own value, as irrealis blocking applies only within a clause or sentence. In addition, a 'ques-context' feature is used to decide whether the clause/sentence is a question. Then, if any determiners are found within the clause/sentence, irrealis blocking is ignored (see Figure 5).

4.5 Model sent5 and Model sent6: Text-Level Features

Taboada et al. believed lexicon-based sentiment classifiers generally favour positive language statements and so previous sentiment research shows a positive bias. Moreover they said, the repetition of a sentiment word found in a sentence shows sentiment depending on how many times the sentiment word is present in the sentence. SO-CAL may show strong positive sentiment, for example in Figure 6, due to the repetition of 'excellent' word. However, Taboada et al. suggested the number of appearance of a sentiment word in a sentence should not decide its overall sentiment intensity. In order to overcome above problems, firstly SO-CAL increased the final SO value of any negative expression by 50% . Secondly, they decreased the weight of words, which appear more often in the document. In this way, they

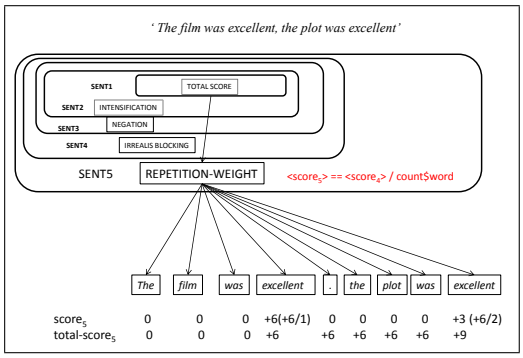


Figure 6 Model sent5: changes sentiment score of the word, dependent on its word count.

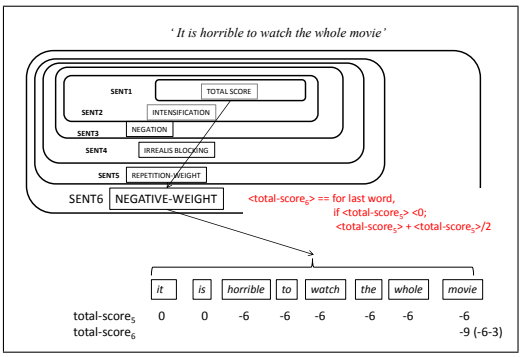


Figure 7 Model sent6: changes the total score, if it is negative.

Table 1 Performance of SO-CAL and Galadriel models for only adjective and all words.

	Only adjectives		All words		
Datasets	**SO-CAL**	**sent1**	**SO-CAL**	**sent1**	**sent6**
Epinions	72.25%	68.89%	80%	65.04%	60.68%
Movie reviews	76.63%	71%	76.37%	70%	65%

decided to override the SO value of the nth appearance of a word with 1/n of its full SO value.

To model SO-CAL's feature for repetition weight of words in Galadriel, a new feature called 'count $word' is introduced, where '$word' is a DATR variable, so this definition works for different actual words, for instance <count excellent>, <count horrid> . This feature allows us to count how many times a word is present in a document. Thus the sentiment score of the word ($word) is divided by 'count $word' to produce the final score for the word (see Figure 6). To model negation weighting, first the system decides whether the overall sentiment is negative. If so, the total score is increased by 50% (see figure 7).

5 Evaluation

We have collected the whole dataset and the dictionary used by SO-CAL. SO-CAL's dictionary contains list of words (adjectives, adverbs, nouns and verbs) with their SO (semantic orientation) values (between −5 and +5). In addition, it has a list of intensifiers with their values in factors (with plus and minus sign). We tested SO-CAL in Galadriel using two data sets, which were based on those used in [11]. The data sets are:

- **Epinions**: 50 reviews each of: books, cars, computers, cookware, hotels, movies, music and phones. As a first step of evaluation we used total 46 (24 positive and 22 negative) reviews.
- **Movies**: 1900 texts from the polarity data set [7]. We used 20 (10 positive and 10 negative) reviews.

We tested Galadriel in several configurations, simulating SO-CAL's 'only adjectives' and 'all words' (including sentiment for adverbs, nouns and verbs) settings, and for all six Galadriel models (sent1 – sent6). Table 1 shows the performances of SO-CAL and Galadriel with adjectives and all words in sent1 and sent6. Table 2 and Table 3 show performances of SO-CAL (all words) with different features and different models of Galadriel (all words)

Table 2 Performance of SO-CAL (all words) using various options.

Features	Epinions	Movies
simple	65.25%	68.05%
negation	67.75%	70.10%
neg+intensifiers	69.25%	73.47%
neg+inten+irrealis	78.25%	75.08%
neg+inten+irr+ neg weight	80.00%	76.37%
neg+inten+irr+ neg w+rep w	80.00%	76.37%

Table 3 Performance of Galadriel models (all words).

Models	Epinions	Movies
sent1	65.04%	70%
sent2	68.02%	72%
sent3	64.03%	69%
sent4	66.50%	67%
sent5	62.12%	67%
sent6	60.68%	65%

Table 4 Comparing performance of SO-CAL and Galadriel on positive and negative reviews.

Reviews	SO-CAL			Galadriel		
	Pos-F	Neg-F	Accuracy	Pos-F	Neg-F	Accuracy
Books	0.69	0.77	0.74	0.82	0.68	0.75
Cars	0.80	0.75	0.78	0.73	0.63	0.68
Computers	0.90	0.89	0.90	0.71	0.44	0.58
Cookware	0.79	0.76	0.78	0.82	0.25	0.54
Hotels	0.80	0.70	0.76	0.75	0.28	0.52
Movies	0.76	0.79	0.78	0.78	0.44	0.61
Music	0.83	0.81	0.82	0.75	0.33	0.54
Phones	0.85	0.83	0.84	0.75	0.66	0.71
Total	0.81	0.79	0.80	0.76	0.46	0.61

respectively. Table 4 shows comparison of the performance of SO-CAL and Galadriel across review types and on positive and negative reviews.

6 Discussion

In this paper, we have shown how the lexicon-based approach to sentiment analysis can be modelled by using inheritance based modelling techniques. Although we are not aiming to match performance of SO-CAL, we provided Galadriel performance figures in different experimental set-ups. We only aimed to show that SO-CAL features can be modelled in Galadriel.

We also have been modelling slightly different existing lexicon-based sentiment analysis approach [2] which is an aspect-based model in Galadriel and merging with SO-CAL, while identifying novel techniques. These models will be evaluated by comparing the existing original methods. From these analyses, an integrated inheritance model of sentiment knowledge of words will be identified and it will be extended to a model of sentiment analysis. In this way the entire sentiment analysis task will be coded as a 'lexical description' task.

We aim to introduce insights from other systems, in particular machine learning approaches, into model. To illustrate, we aim to use Galadriel to handle phrases that are commonly used to express sentiment. In order to handle such phrases, we will focus on building a model in Galadriel, using a corpus-based machine learning methodology to refine this model with examples derived from corpus data. This allows supporting exceptions to general rules.

Acknowledgements. I would like to thank Dr. Maite Taboada, who kindly gave me access to the SO-CAL system, its dictionaries and the datasets, and my supervisors Dr. Roger Evans and Dr. Gulden Uchyigit, for their support in producing this paper.

References

1 Walter Daelemans, Koenraad De Smedt, and Gerald Gazdar. Inheritance in natural language processing. *Computational Linguistics*, 18(2):205–218, 1992.

2 Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.

3 Roger Evans. The extended lexicon: language processing as lexical description. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 270–276, Hissar, Bulgaria, September 2013. RANLP 2011 Organising Committee.

4 Roger Evans and Gerald Gazdar. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216, 1996.

5 Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125, 2006.

6 Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

7 Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. ACL, 2004.

8 Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

9 Livia Polanyi and Annie Zaenen. Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications*, pages 1–10. Springer, 2006.

10 Maite Taboada, Caroline Anthony, and Kimberly Voll. Methods for creating semantic orientation dictionaries. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 427–432, 2006.

11 Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

12 Maite Taboada and Jack Grieve. Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Re# port SS# 04# 07), Stanford University, CA, pp. 158q161. AAAI Press*, 2004.

13 Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. ACL, 2002.

Interactive 3D Reconstruction: New Opportunities for Getting CAD-ready Models

Julius Schöning

Institute of Cognitive Science
University of Osnabrück, Germany
juschoening@uos.de

Abstract

A multitude of image-based 3D reconstruction and modeling techniques exist, which have achieved significant success in recent years. However, these techniques still lack certain abilities. For example, current 3D reconstruction techniques cannot decompose an object into its individual subparts. Thus, a printed model will consist of one single monolithic piece, which does not allow composing or decomposing parts, does not allow movable or flexible parts, and does not allow manufacturing the model from multiple different materials like wood, metal, or plastic. I reviewed the work in the research area of 3D reconstruction and provide an analysis of neglected research objectives and current drawbacks. Furthermore, I propose a mock-up of an interactive tool as a guideline for future research which describes a possible architecture, user interfaces, and processing pipeline, to overcome existing drawbacks of 3D reconstruction techniques.

1998 ACM Subject Classification I.4.5 Reconstruction

Keywords and phrases 3D Reconstruction, 3D Modeling, Interactive Reconstruction, CAD-ready

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.54

1 Introduction

The reconstruction of 3D objects and scenes from image and video footage captured in 2D (monocular) or 3D (stereo) is an active research topic in the area of computer vision. Many approaches reconstruct objects or scenes with automatic algorithms from image sequences [28, 20, 22, 23, 34, 27]. Even today, 3D reconstruction is a difficult task due to the numerous irregularities of objects and scenes. Thus, many automatic algorithms fail to create a 3D model when, e.g., only few images are available [14], the images are not taken in a controlled environment, the object contains textureless surfaces, or parts of objects are occluded. In addition, when a 3D model is created by an automatic or semi-automatic method [15, 2, 8, 17, 12], the re-transformation via a 3D printer or CNC-machine ends up with a solid monolithic block of, e.g., plastic. This is because most reconstruction algorithms focus on the surfaces or the skeleton tracking of an object and not on its subparts, despite it being possible to manufacture more than solid blocks with today's techniques of CNC-machines and 3D printers out of these models.

When people look at objects or scenes they can directly extract the visible geometry, as well as the hidden geometry. In addition, humans are capable to do this extraction even without the necessity for stereo information and can identify materials and the degree of freedom of individual subparts. But an even more basic decision a person can make without great difficulty is whether a part is fixed or movable. I want to expand the computational power of today's computer vision techniques with the conceptional background knowledge provided by the user who, since birth, acquired real world knowledge and became a domain

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 54–61

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

expert. Currently, this knowledge cannot be formalized into computer algorithms and it is unlikely that this situation will change in the foreseeable future. The idea of uniting the strengths of computer vision with the user's domain knowledge and his or her decision-making ability is, however, not new in this research area but is rarely found. I therefore provide a literature review of 3D reconstruction and modeling techniques and analyze weaknesses of existing methods. I provide a discussion of how an interactive process could overcome existing weaknesses and propose a possible working process for the interactive creation of non-monolithic CAD-ready 3D models with flexible subparts out of defined material. Such CAD-models allow manufacturing each subpart on CNC-machines or 3D printers.

2 State of the Art Review

For 3D reconstruction techniques three main research areas are observable: Fully automatic reconstruction, interactive reconstruction, and supervised reconstruction. All of them work "either-or" with monocular or stereo images or video footage. The 3D reconstruction techniques discussed in the following do not necessarily focus on reconstructing scenes or objects for modeling purposes only. Instead other applications like, e.g., robot navigation, might be considered.

2.1 Fully automatic reconstruction

Based on large image collections fully automatic 3D reconstruction methods have been shown to work well [25, 15]. The photo explorer by Snavely et al. [27] uses an unstructured image collection of a scene, e.g., acquired from the Internet, and converts them to a 3D model which can be exported. Instead of an image collection Pollefeys et al. [23] use a monocular hand-held camera to build visual models of scenes for fusing real and virtual scenes. 3D modeling of indoor environments for robot navigation is done by RGB-D mapping using a depth camera [10]. Another famous approach, known as *KinectFusion* project [18, 16], uses a RGB-D camera as hand scanner. The reconstruction of objects on consumer mobile phones which are moved around the object of interest is described in [28]. A method shown by Pan et al. [20] for probabilistic feature-based online rapid model acquisition reconstructs freely rotated objects in front of a static video camera in near real-time. Due to the fact that the system guides the user with respect to the manipulation, i.e., rotation of the object, this method might not entirely fit in the category of fully automatic methods. However, like some other methods in this category it still requires some interaction to be initialized before reconstruction.

The handling of delicate structures, textureless surfaces, hidden boundaries, illumination, specularity, or even dynamic or moving objects, like in natural recordings, are not taken into account by these automatic reconstruction techniques. High quality automatic reconstruction usually depends on a predefined recording environment with, e.g., special or fixed lighting conditions, a rotating object with a fixed camera or a rotating camera with a fixed object, and special camera systems like ToF cameras [30, 7], stereo cameras [13], IR cameras [9] or laser sensors [29, 1].

2.2 Interactive semi-automatic reconstruction

Kowdle et al. [15] noticed the above mentioned issues of fully automatic reconstruction for object creation and came up with a semi-automatic approach. In this approach they put the user in the loop of computational reconstruction. This interactive approach yields

a reconstruction of a monolithic high quality object or scene from an image collection. Interactive approaches for reconstructing man-made architecture [2, 8, 17, 12] are based on the presence of common geometric primitives. Usually such approaches use a computational pre-processing step that extracts edges or point clouds to create an initial 3D model. In the interactive processing step the user iteratively refines the model by editing the polygon based model or sketching new polygons. This kind of 3D drawing for reconstruction purposes is also mentioned by Hengel et al. [32, 31].

With interactive 3D reconstruction techniques, models can be reconstructed from a single image [6]. In general, the potential of semi-automatic approaches is shown in many computer vision systems and computer vision related disciplines. To point out a few examples where interactive concepts led to a significant improvement: interactive segmentation of images [19, 4, 24], the integration of domain knowledge in a visual recognition system [5], or the inter-active training of a video recognition system [11].

2.3 Supervised or active learning reconstruction

Supervised and active learning as a field of machine learning improves many computer vision issues, e.g., annotation, object categorization and video categorization. Nowadays, supervised or active learning are not yet used in interactive 3D reconstruction. Using active learning, 3D reconstruction improves itself by asking the user for specific information. Using this specific information a supervised learner can be trained. As a result the supervised learner is able to process common reconstruction task like reconstructing walls or entire rooms after a few iteration of active learning. These learning loops improve an interactive reconstruction process and reduce the interaction tasks of the user. A plausible reason why supervised or active learning are not used in 3D reconstruction yet, might be the high dimensional solution space, to which the learning algorithms must be adapted.

2.4 Discussion

What all activities have in common is that reconstructed 3D objects or scenes are monolithic and not interpreted in terms of parts, subparts and more deeper details. Depending on the scope, monolithic results are sufficient for tasks like indoor navigation [21, 10], reconstruction of urban outdoor environments [22] or reconstruction of human body as a deformable collection of objects [3], but not quite sufficient in creating detailed models with composable subparts for, e.g., replicas. Current reconstructed 3D models are only a kind of a solid monolithic geometry without subparts. As consequence these 3D models cannot be applied to many use cases like simulation tasks with these models might not be meaningful because the interrelationships between the subparts of the object are missing.

3 Interactive 3D reconstruction

Are the existing techniques sufficient to build 3D replica from images? Can these techniques be applied by a non-expert user? Is special hardware required? Does the user need a rotating plate, laser, stereo camera or even a main frame computer? The answer to these questions is mostly *No*. These and other questions, together with the discussed considerations, are the starting-point for our proposed research guideline. A major question is: “How can I get models of real world objects in such a way that I will be able to translate them back to real world replicas?”. In the following an overview of the relevant research questions concerning this problem is given.

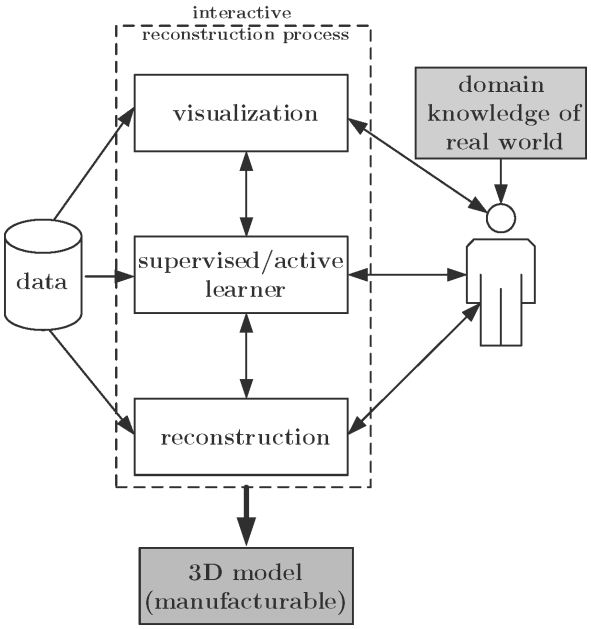


Figure 1 Interactive reconstruction architecture.

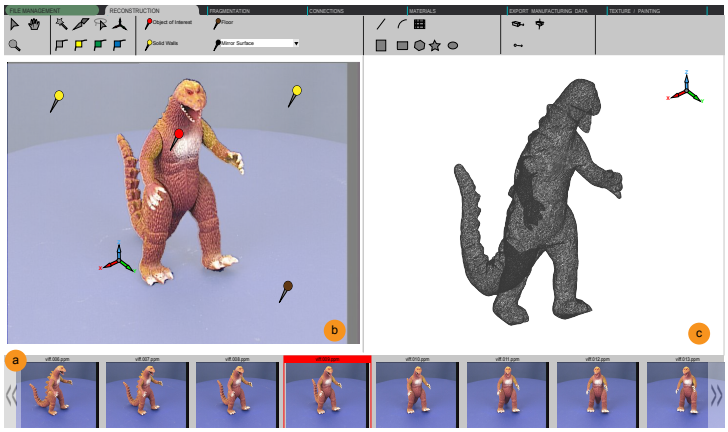
In an early stage of the architectural design, it became obvious that due to hidden information like material attributes, the fragmentation of objects or scenes, and the determination of degrees of freedom, deep real world knowledge must be present. The necessary real world knowledge cannot be explicitly integrated in existing algorithms due to the sheer complexity it would create, e.g., for material detection. Interactive modeling techniques are, however, able to integrate such complex domain knowledge iteratively.

A possible setup for such an interactive architecture is shown in Figure 1. The architecture consists of three main parts: i) the input data in the form of monocular or stereo images / video footage, as well as additional data like physical interrelationships – gravity, reflection etc.–, ii) the interactive reconstruction process and iii) the user as domain expert [26]. The interactive reconstruction process should join the computational power of today's computers with the conceptional knowledge of the user in order to solve at present computationally unfeasible issues. Thus, the computer remains the "work horse" of the process, while supervised and active learning algorithms translocate the load from the user to the computer, if possible.

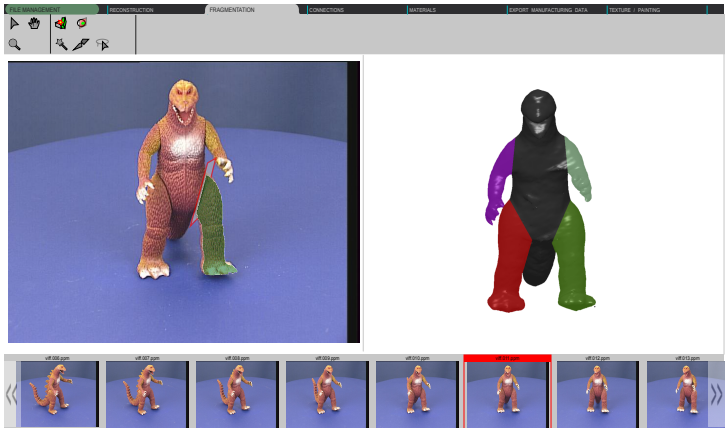
To achieve a better understanding of already existing techniques, algorithms, methods and processing pipeline, I designed a mock-up application, without underlying functionality, which is shown in Figure 2. For presentation purposes, the interactive reconstruction process is described using the Oxford "dinosaur" image collection [33], which already can be automatically reconstructed in high quality due to sufficient and overlapping images. However, all known reconstruction methods – automatic or semi-automatic – create only a monolithic or a skeleton model, not a model consisting of subparts. Throughout the following discussion, normal user interaction techniques like undo, redo, zoom can be assumed to be available.

3.1 Reconstruction and Fragmentation

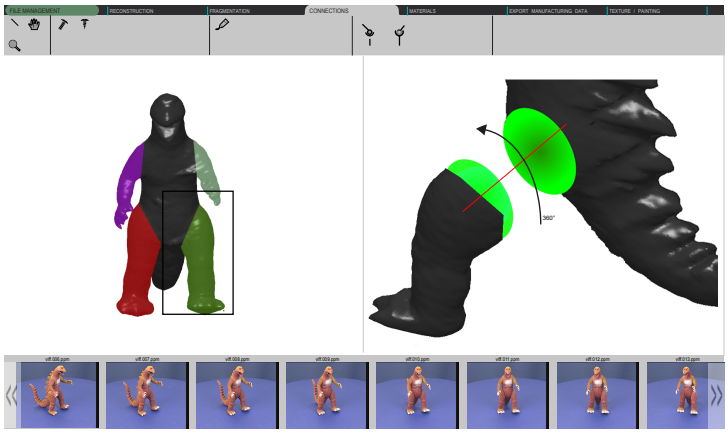
Based on an image or keyframe from a storyboard ⓐ the user may identify objects of interest with a marker pin. This triggers the calculation of a point cloud [25], which is then presented



a Reconstruction view, interactive creation of a 3D model out of monocular or stereo images or video footage.



b Fragmentation view, breaking the monolithic 3D model down into its parts.



c Connection view, adding detachable or permanent connections and theirs degrees of freedom.

Figure 2 Mock-up of interactive 3D Reconstructor – Several subfigures.

to the user. For fine trimming etc. common tools like scalpel, marker for borders and edges, and scribble tools can be used.

The shared view of the pre-processing ⓑ and the post-processing window ⓒ, as shown in Figure 1a, enable the user to directly modify the point cloud or the meshed point cloud. This allows the user to add missing or occluded information to the 3D model.

After the monolithic 3D model is reconstructed with a desired level of detail, the next step is to break down the model to its components or subparts until every subpart itself is monolithic, as shown in Figure 1b.

3.2 Connections and Materials

If all subparts have been identified the user has to model the connections between these parts. This step has to account for specific details like the type of connection, e.g., a ball joint, as well as specific information like extra allowance, rotation axis and maximal rotation angle, as illustrated in Figure 1c.

The assignment of materials to each subpart is, in general, straightforward. However, an automatic consistency check should be included to ensure the compatibility of connection types and materials.

3.3 Exporting manufacturing data

The last step in creating a replica is the re-transformation to the real world. This is done using 3D printers or CNC-machines. Since a variety of online services exist, the actual presence of such a machine is no requirement. After determining the dimensions for scaling the model, the export of the manufacturing data can be performed automatically without further user interaction. The replica itself is manufactured out of the assigned materials. In case of the dinosaur a painting or a texture would be necessary. How this manufacturing step can be performed without standardized hardware is, at the moment, difficult to assess. Of course, possible solutions like printing the texture to self-adhesive foil exist. Because painting and texture are really the last step of creating replicas and it is not necessary in lots of cases, this topic is not considered in detail yet. Next to the exporting manufacturing data, exports for CAD applications like simulation and modeling tools are provided. Using the export to CAD function provides engineers and architects with models, which they would call a model.

4 Application

The creation of replica is the main focus of this approach. This might enable an improvement in biological-inspired technologies, thus creating replica of plants and animals from images. Also an improvement for reverse engineering tasks, or an improvement for rapid prototyping out of mock-ups, or simply sharing sculpture with other artists is possible.

Under consideration of a 3D model with all parts and their physical coherence many applications might benefit. The reconstruction of scenes [21] can be used for robot navigation or for planning and executing relocations.

5 Conclusion

The literature review of 3D reconstruction and the discussion of our mock-up application point out the main weakness of current methods – the impossibility of creating non-monolithic 3D models. To overcome this weakness I started to implement the proposed semi-automatic

and interactive architecture for image-based reconstruction as software prototype. I expect the identification of even more weaknesses of current 3D reconstruction and computer vision methods in the course of research conducted in the proposed directions. Finally I am optimistic an interactive 3D reconstruction tool could create CAD-ready models of real world objects which can be translated back to the real world with 3D printers and CNC-machines.

References

1 A. Aouina, M. Devy, and A. M. Hernandez. 3D modeling with a moving tilting laser sensor for indoor environments. *International Federation of Automatic Control World Congress*, 2014.

2 M. Arikan, M. Schwärzler, S. Flöry, M. Wimmer, and St. Maierhofer. O-Snap: Optimization-based snapping for modeling architecture. *ACM T GRAPHIC*, 32(1):1–15, 2013.

3 A. Barmpoutis. Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D. *IEEE Transactions on Cybernetics*, 43(5):1347–1356, 2013.

4 D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

5 S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European conference on Computer vision (ECCV)*, pages 438–451. Springer, 2010.

6 T. Chen, Z. Zhu, S.-M. Shamir, A.and Hu, and D. Cohen-Or. 3-Sweep. *ACM T GRAPHIC*, 32(6):1–10, 2013.

7 Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3D shape scanning with a time-of-flight camera. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1180. IEEE, 2010.

8 P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer graphics and interactive techniques - SIGGRAPH*, 1996.

9 P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *INT J ROBOT RES*, 31(5):647–663, 2012.

10 P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental Robotics*, pages 477–491. Springer, 2014.

11 B. Höferlin, M. Höferlin, D. Weiskopf, and G. Heidemann. Scalable video visual analytics. *Information Visualization Journal*, 2013.

12 N. Kholgade, T. Simon, A. Efros, and Y. Sheikh. 3D object manipulation in a single photograph using stock 3D models. *ACM T GRAPHIC*, 33(4), 2014.

13 K. Konolige. Projected texture stereo. In *IEEE Robotics and Automation (ICRA)*, pages 148–155. IEEE, 2010.

14 A. Kowdle, Y.-J. Chang, D. Batra, and T. Chen. Scribble based interactive 3D reconstruction via scene co-segmentation. In *IEEE Image Processing (ICIP)*, pages 2577–2580. IEEE, 2011.

15 A. Kowdle, Y.-J. Chang, A. Gallagher, D. Batra, and T. Chen. Putting the user in the loop for image-based modeling. *INT J COMPUT VISION*, 108(1-2):30–48, 2014.

16 Microsoft Research. 3D surface reconstruction `http://research.microsoft.com/en-us/projects/surfacerecon/`, Nov 2014.

17 P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. Gool, and W. Purgathofer. A survey of urban reconstruction. In *Computer Graphics Forum*, volume 32, pages 146–177. Wiley Online Library, 2013.

18 R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

19 J. Ning, L. Zhang, D. Zhang, and C. Wu. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43(2):445–456, 2010.

20 Q. Pan, G. Reitmayr, and T. Drummond. ProFORMA: Probabilistic feature-based on-line rapid model acquisition. *British Machine Vision Conference (BMVC)*, 2009.

21 G. Pintore and E. Gobbetti. Effective mobile mapping of multi-room indoor structures. *The Visual Computer*, 30(6-8):707–716, 2014.

22 M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, and et al. Detailed real-time urban 3D reconstruction from video. *INT J COMPUT VISION*, 78(2-3):143–167, 2008.

23 M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *INT J COMPUT VISION*, 59(3):207–232, 2004.

24 L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM T GRAPHIC*, volume 25, pages 599–604. ACM, 2006.

25 J. Schöning and G. Heidemann. Evaluation of multi-view 3D reconstruction software. In *Computer Analysis of Images and Patterns (CAIP)*. In Press, Sep. 2015.

26 J. Schöning and G. Heidemann. Interactive 3D modeling - a survey-based perspective on interactive 3D reconstruction. In *Pattern Recognition Applications and Methods (ICPRAM)*, volume 2, pages 289–294. SCITEPRESS, 2015.

27 N. Snavely, St. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM T GRAPHIC*, 25(3):835, 2006.

28 P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3D reconstruction on mobile phones. In *IEEE Computer Vision (ICCV)*, pages 65–72. IEEE, 2013.

29 S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *IEEE Robotics and Automation (ICRA)*, volume 1, pages 321–328. IEEE, 2000.

30 M. van den Bergh and L. van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72. IEEE, 2011.

31 A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr. VideoTrace:rapid interactive scene modelling from video. *ACM T GRAPHIC*, 26(3):86, 2007.

32 A. van den Hengel, R. Hill, B. Ward, and A. Dick. In situ image-based modeling. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

33 Visual Geometry Group, University of Oxford. Multi-view and oxford colleges building reconstruction - dinosaur `http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`, Aug. 2014.

34 M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *ACM T GRAPHIC*, 33(4):1–12, 2014.

Representing Temporal Patterns in Computer-interpretable Clinical Guidelines

António Silva[1], Tiago Oliveira[2], Paulo Novais[2], and José Neves[2]

1 **Department of Informatics, University of Minho**
Braga, Portugal
pg25307@alunos.uminho.pt

2 **Algoritmi Research Centre/Department of Informatics, University of Minho**
Braga, Portugal
{toliveira,pjon,jneves}@di.uminho.pt

Abstract

Computer-interpretable Guidelines (CIGs) as machine-readable versions of clinical protocols have to provide appropriate constructs for the representation of different aspects of medical knowledge, namely administrative information, workflows of procedures, clinical constraints and temporal constraints. This work focuses on the latter, by aiming to develop a comprehensive representation of temporal constraints for machine readable formats of clinical protocols and provide a proper execution engine that deals with different time patterns and constraints placed on them. A model for the representation of time is presented for the CompGuide ontology in Ontology Web language (OWL) along with a comparison with the available formalisms in this field.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods, J.3 Life and Medical Sciences – Medical Information Systems

Keywords and phrases Computer-interpretable Guidelines, Temporal Constraints, Clinical Decision Support, Ontologies

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.62

1 Introduction

CDSSs providing patient-specific recommendations follow a more consulting style of communication and require substantial modelling activity [8]. The models used to represent medical knowledge in such systems range from probabilistic models and decision trees to task-network models (TNMs). The latter are arguably the most used, mainly because they allow the representation of chains of events and a wide variety of situations [7, 11]. TNMs are the basis for Computer-interpretable Guidelines (CIGs), machine readable versions of clinical protocols. There are many CIG models available, but they have yet to overcome, for the most part, the stage of academic project. As a result, there is no standard for the representation of CIGs. There are, however, influential CIG approaches such as GLIF3 [3], PROforma [6], Asbru [12], SAGE [15], and GLARE [14]. The downside of this is each model tends to focus solely on one aspect of the representation of clinical protocols, disregarding, or not paying as much attention to, the other important aspects. In the representation of CIGs, according to [16], one should take into account: the representation of administrative information, the construction of workflows of clinical procedures, the representation of clinical constraints and patient state conditions, and the representation of temporal constraints. The CompGuide project [9] aims to build a comprehensive ontology for CIGs which gathers the main strengths of the existing approaches. It explores Ontology Web Language (OWL)

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 62–69
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

as the support language for the definition of representation primitives and the procedural logic of clinical protocols. The work presented in this paper concerns the representation of temporal patterns within CompGuide, in a way that balances complexity and expressiveness. The representation of temporal patterns conveyed in clinical protocols is quite complex, firstly because they may appear in various textual expressions, such as in the expression "history and physical every 3–6 months for 2 years, then every 6 months for a total of 5 years" extracted from a clinical protocol for the treatment of colon cancer [2], and secondly because they be extremely intricate. As such, temporal model is proposed for the definition of temporal constraints and the management of time in medical algorithms. At the same time, it is a complementary model to the existing approaches, by gathering the most developed aspects of each approach and filling in the gaps that often render said approaches incomplete. The main contribution of this model is an integrated representation of durations, periodicities, stop conditions for clinical tasks and temporal restrictions for conditions about the state of a patient, which is something the existing CIG models have not achieved.

The paper is organized as follows. Section 2 provides related work about the representation of temporal constraints in CIG models. Some background on the CompGuide ontology is presented in Section 3. The temporal model along with examples for discussion are provided in Section 4. Finally, Section 5 presents conclusions about the work developed so far and future work considerations.

2 Related Work on the Temporal Representation of Clinical Protocols

The management of time is one of the main concerns in CIG modelling, as clinical processes are chains of events unfolding over time. A few CIG approaches have been specifically devised to deal with temporal constraints. GLIF3 [3] deals with both temporal constraints placed on patient state conditions and durations of actions. Asbru [12] provides a comprehensive representation model for durations as well. In fact, this CIG model presents clinical protocols as time-oriented skeletal plans for which it is possible to define time annotations, which may be constraints on the starting time and ending time of tasks (such as earliest possible start and earliest possible ending), maximal and minimal durations, and cyclical time points (e.g., every morning, every day, etc.). A step further is given in GLARE [13], which introduced the representation of periodicities for events and repetition schemes. This formalism was later expanded in [1]. The new version of the work provides an enhanced formalism to express periodicities, with the possibility of defining delays between the cycles of the periodic event. It also became possible to define more complex periodicity periodicity patterns. For instance, each cycle of a periodic event may have itself an associated periodicity. Another interesting development is the mapping of the high-level time patterns to a Simple Temporal Problem (STP) [4] reasoning framework, in which a graph representation of the guideline, resulting from the calculation of the relative time constraints of tasks, is provided. Despite being one the most used temporal reasoning AI techniques, the authors mention they were unable to represent complex time patterns with it, as the STP-tree structure they produce is not suited for events that repeat over time. A similar approach is adopted in [5], but in this case the temporal representation was tailored for clinical plans in the oncology. This reflects the dominant view of pure AI approaches [10], stemming mainly from logic, being faced with obstacles when applied to the medical domain. In both Asbru and GLARE it is not possible to express temporal constraints about clinical parameters of the state of the patient.

Temporal constraints in clinical protocols can be divided into two groups. One consists of the constraints placed on conditions about the state of a patient in order to express for

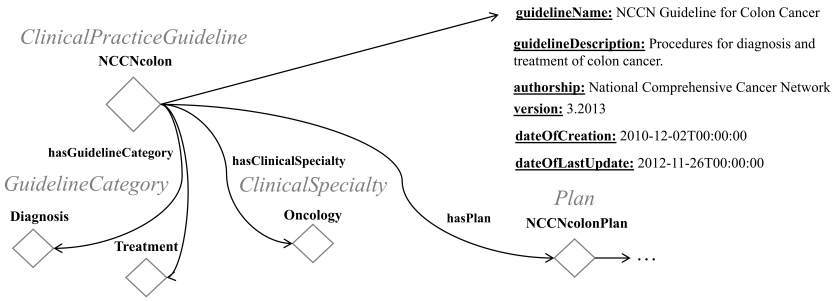


Figure 1 Initial definition of a National Comprehensive Cancer Network guideline for the treatment of colon cancer in the CompGuide ontology.

how long a clinical parameter holds a certain value. The other group consists of constraints placed on the execution of tasks. In this group, one can have qualitative and quantitative constraints. The former are expressed through the control relationships in the guideline and represent the relative order of tasks. The latter include patterns such as durations, delays, periodicities and repetitions. As far as our knowledge goes there is a lack of an approach encompassing all these aspects of temporal constraints.

3 The CompGuide Ontology

The CompGuide [9] ontology provides a task network model representation for clinical protocols in OWL. In order to fulfill this purpose, it follows a logic in which complex information elements are represented as instances with multiple object properties connecting them to other instances, and simple information which cannot be further decomposed is represented using data properties. However, simple information that is reusable and will most likely be needed across different guidelines is represented as instances from specific classes as well. In this regard the representation is similar to a linked list of procedures.

As such, a clinical protocol is represented as an instance of the *ClinicalPracticeGuideline* class. Individuals from this class have a set of data and object properties allowing the representation of descriptive and administrative guideline information such as the name of the guideline, its general description, date of creation and last update, version, clinical specialty, category, intended users, and target population. An example of the initial definition of a guideline is given in Figure 1. The guideline is the NCCN Clinical Practice Guideline for Colon Cancer [2], one of the case studies we are developing.

Every instance representing a guideline is linked to an instance from the class *Plan*, which is a container of tasks, a complex task. In turn, an instance from *Plan* is linked to other instances symbolizing basic tasks. These basic tasks are represented using three classes: *Action*, *Decision* and *Question*. The objective here is to create a recommendation plan containing references to specific types of tasks. The *Action* class expresses a procedure should be carried out by a health care professional. There are several subtypes of actions in the ontology specifying their nature with more detail. The *Decision* class is used to make assertions about the state of the patient, to infer new information from the existing one. The most obvious example of such a task is clinical diagnosis. The *Question* task is used to get information about the symptoms, health condition or other parameters that may help to characterize the state of a patient. *Questions* are also used to register information from the observations of the physician, and to store results from clinical exams. This type of task

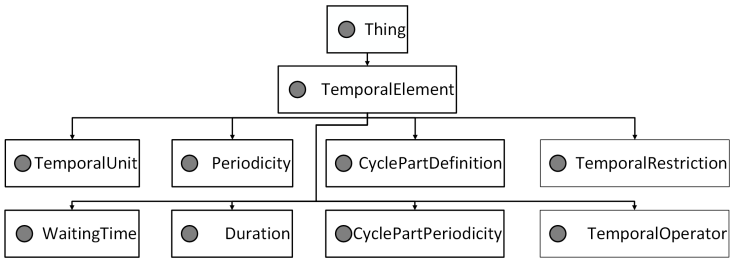


Figure 2 Classes of the model for the representation of temporal constraints in the CompGuide ontology.

gathers all the information necessary for the execution of a clinical protocol. Through object properties, it is possible to define the different control relations that may exist between tasks, the sequence of execution of tasks or if they should be executed simultaneously or concurrently. Regarding this, it is possible to define the sequential execution of tasks, the parallel execution of tasks, points in which one of various alternative tasks is selected (either automatically or by choice of the user), and careflow synchronization points. Additionally, the ontology provides different types of clinical constraints. From simple conditions determining the selection of an option in a decision task, to trigger conditions, which are used to selected a task from a list of alternatives. It is also possible to express pre-conditions, which are requirements to be verified before the execution of a task, expected outcomes of tasks in terms of changes in the state of a patient, and repetition conditions for clinical tasks. Comparatively, CompGuide does not require proficiency in a constraint programming language in order to define these conditions, unlike the existing approaches. Moreover, it provides an increased expressiveness in the definition of tasks and control relationships.

4 Proposal for a Temporal Representation Model

The definition of temporal representation primitives follows the logic described in Section 3. The classes of the temporal model are shown in Figure 2. The main classes are represented as subclasses of *TemporalElement*. One of those subclasses is *TemporalUnit*, which represents the different granularities a temporal constraint may have, such as *second*, *minute*, *hour*, *day*, *week*, *month*, and *year*. The control relations mentioned in Section 3 are responsible not only for the establishment of a workflow of tasks, but also for the definition of qualitative temporal constraints, i.e., the relative order of tasks within the guideline. At runtime, a guideline execution engine analyses these constraints and builds a map of task execution. As this is an aspect already included in the ontology and in conformity with existing approaches, this section is concerned mainly with quantitative temporal constraints.

4.1 Temporal Constraints on the Execution of Tasks

Expressing for how long a task should be executed is one of the main temporal patterns in clinical protocols. In the proposed model this possible with the *Duration* class. The attributes characterizing this class are encoded as necessary conditions in OWL (as it happens with all the other classes). As such, to define a duration, one should choose either to define a maximal and minimal duration, trough the *maxDurationValue* and *minDurationValue* data properties, or to define an exact value for the duration, through the *durationValue* data property. The range of these data properties are decimal numerical values. Regardless of

the type of value one defines, it is always necessary to define a temporal granularity for a decision, which is done through the *hasTemporalUnit* object property which links instances of *Duration* to instances of *TemporalUnit*. Compared with Asbru [12], this form of temporal representation is simpler because it does not feature annotations about the earliest and latest start and ending times of tasks. Such a simplification may be regarded as limitation, yet the objective was to meet the temporal restrictions conveyed in clinical protocols for which the temporal constructors of Asbru may be excessive and impractical. Durations are defined for *Actions* and *Plans* since these are the only tasks which can be executed over a certain time. In fact, in most cases, the information about the duration of tasks is conveyed either as an exact value or as an interval, such as in the case of the natural language expression "perform neoadjuvant therapy for 2–3 months" extracted from [2]. This removes the need for complex time annotations. However, it is still possible to express delays between tasks with the class *WaitingTime*, whose values (max, min, exact) and temporal granularity are defined in a similar way as in *Duration*. Delays can be defined for all classes of tasks in CompGuide, allowing the representation of situations such as the *Action* "re-evaluation for cancer resection after 2 months of preoperative chemotherapy", in which there is a "re-evaluation" task which should be delayed for "2 months", after the end of "preoperative chemotherapy". However, the proposed model provides a better definition of intervals and exact values. For instance, in [1], durations are always represented as intervals. To express exact durations, it is necessary to state the upper and lower bounds of the interval are the same. By having a data structure for exact durations, one can make the processing of such constraints simpler.

The representation of periodic tasks is the most complex pattern. In CompGuide, this pattern is represented by the class *Periodicity*. A periodicity can be defined for any type of task. However, the periodic event is bound by either a duration, a repetition constraint or a stop condition about the state of the patient. The duration is defined through the reuse of the *Duration* class. As such, an instance of *Periodicity* can also be linked to an instance of *Duration* through the *hasDuration* object property, thus determining for how long a periodic event should take place. On the other hand, if one wants to state the number of times the event should be carried out (the same is to say the number of cycles of the periodic event), it is necessary to formulate a repetition constraint, which is possible through the *repetitionValue* data property with a range of integer numerical values. Alternatively, it could be the case the periodic task should only occur until a condition about the state of a patient is met. To express this, one uses the *hasStopCondition* object property to link an instance of periodicity to instances of the class *StopCondition*. While it is possible for a periodicity to have a duration and a stop condition, a repetition value and a stop condition, or just a stop condition, it is not possible to have both a duration and a repetition value for it is considered to be redundant information. With a duration and a frequency it is already possible to calculate the number of repetitions of an event and vice versa. The stop condition takes precedence over the other temporal restrictions, so if the condition is met, the task is immediately stopped.The frequency of the periodicity and the temporal granularity are defined in the data property *periodicityValue* and through the *hasTemporalUnit* object property, respectively. A cycle of a periodic event may have itself an associated periodicity or duration. In order to represent this, the object property *hasCyclePartDefinition* is used. It links instances of *Periodicity* to instances of *CyclePartDefinition*, where one actually defines the periodicity of the cycle or its duration. In the class *CyclePartDefinition*, the periodicity of the cycle is defined through the object property *hasCyclePartPeriodicity*, which links instances of *CyclePartDefinition* to instances of *CyclePartPeriodicity*. *CyclePartPeriodicity* is different from the main *Periodicity* class. This intricate representation of periodic tasks is depicted in

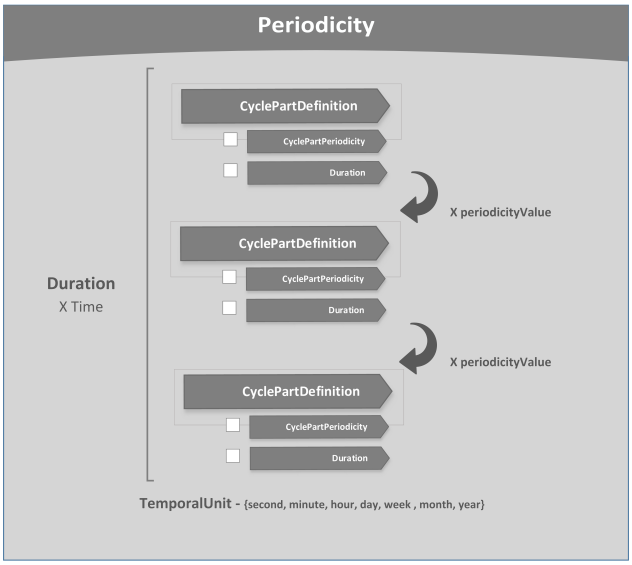


Figure 3 Schematic representation of different levels of periodicity for a task, with a distinction between *Periodicity* and *CyclePartPeriodicity*.

Figure 3 in order to convey the distinction between *Periodicicty* and *CyclePartPeriodicity*. One can argue it would be simpler to reuse the *Periodicity* class rather than defining another class for the definition of the periodicity of each cycle, but by doing so it would be possible to nest periodicities inside one another infinitely, which does not quite fit the cases appearing in clinical protocols. Cancer treatment guidelines are usually very rich in periodicity temporal patterns, mainly because of the chemotherapy/radiotherapy regimens they recommend. An action representing this is the following for CapeOX chemotherapy: "CapeOX every 3 weeks with the administration of capecitabine twice daily for 14 days" [2]. In the statement, it is possible to distinguish the *Periodicity* (every 3 weeks) from the *CyclePartPeriodicity* (twice daily for 14 days). In terms of expressiveness, this approach allows the representation of temporal constraints which are not representable in Asbru, which is one of the dominant CIG models. It is, at the same time, an adaptation (mainly in the periodicity components) of the formalism presented in [1].

4.2 Temporal Constraints on the State of a Patient

Most CIG approaches represent temporal constraints for the conditions about the state of a patient as strings in description fields. Considering the different conditions in CompGuide, such as conditions for decisions, trigger conditions, expected outcomes, repetition types of conditions, and pre-conditions, it would be advantageous to develop a form of automated reasoning about them, thus making necessary the development of a structured way to represent them.

In CompGuide, a temporal constraint for conditions about the state of a patient is represented by an instance of the *TemporalRestriction* class. To link the constraint to an instance representing a condition, it is necessary to use the *hasTemporalRestriction* property, which, although non-mandatory, can be asserted for any of the above-mentioned conditions. For each instance of *TemporalRestriction* it is necessary to specify a temporal operator through the *hasTemporalOperator* object property. This object property points to individuals of the class *TemporalOperator*. This is an enumerated class that can only have a limited number of

instances, namely the following: *currently*, *within_the_last*, *during*, *within_the_following*. The temporal operators represent the reach of a temporal constraint and are coupled with temporal ganularities, defined through the *hasTemporalUnit* object property, and temporal restriction values. The latter are expressed through data properties such as *maxTemporalRestrictionValue* and *minTemporalRestrictionValue*, or *temporalRestrictionValue* for exact values. The operator *currently* expresses the condition must hold at the time of execution, when the clinical task is being considered for implementation. On the other hand, *within_the_last* is used when one wants to express a condition must have held true at least once, within a period of time just before execution time. But, if the intention is to state that a certain condition must have to continuously hold true during a period of time just before the execution time, then one should use the operator *during*. These three temporal operators fit in temporal restrictions of conditions for decisions, trigger conditions, repetition conditions, and pre-conditions because these are used to reason about the present or the past. Yet, in an expected outcome, it is necessary to express a condition about the future, in which one expects to observe the effect a clinical task has after being applied to a patient. For such, it is possible to use the operator *wihin_the_following*, which bounds the observation of the condition to a certain period in the future, starting from the time of execution. An example of a temporal restriction using a temporal operator is an expected outcome of a chemotherapy regimen, such the statement "the tumor should become operable within 6–7 months of FOLFOX or CapeOX chemotherapy", extracted from [2].

5 Conclusions and Future Work

Besides the need to further evaluate the expressiveness of the model with an array of case studies containing a wide variety of temporal patterns, it is necessary to focus on another important aspect of CIG temporal representation, which is interpretation. CIG interpretation can happen in two distinct moments: acquisition and execution. During acquisition, the objective of the interpretation is to check the consistency, which in this case is the temporal consistency, of a guideline, and during execution, the objective is to provide timely recommendations. In both situations, the treatment of temporal constraints goes beyond the isolated processing of each constraint. Instead, it is necessary to take into account the so called part-of relations, which exist when an atomic task is inside a complex task such as *Plan*. In such case, and just to give an example, the combined durations of the tasks inside a *Plan* cannot be greater than the duration of the *Plan* itself. A time manager is currently under development for the CompGuide ontology. The next steps include the evaluation of the tractability, correctness and completeness of this approach. The presentation of the temporal plans during execution will have the form of workflows with timely notifications about when the tasks should be executed and how long they should last.

Acknowledgements. This work is part-funded by ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980 and project scope UID/CEC/00319/2013. The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/85291/2012.

References

1 Luca Anselma, Paolo Terenziani, Stefania Montani, and Alessio Bottrighi. Towards a Comprehensive Treatment of Repetitions, Periodicity and Temporal Constraints in Clinical Guidelines. *Artificial Intelligence in Medicine*, 38(2):171–195, 2006.

2 Al Benson, Tanios Bekaii-Saab, Emily Chan, Yi-Jen Chen, Michael Choti, Harry Cooper, and Paul Engstrom. NCCN Clinical Practice Guideline in Oncology Colon Cancer. Technical report, National Comprehensive Cancer Network, 2013.

3 Aziz a Boxwala, Mor Peleg, Samson Tu, Omolola Ogunyemi, Qing T Zeng, Dongwen Wang, Vimla L Patel, Robert a Greenes, and Edward H Shortliffe. GLIF3: A Representation Format for Sharable Computer-Interpretable Clinical Practice Guidelines. *Journal of Biomedical Informatics*, 37(3):147–61, June 2004.

4 Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1):61–95, 1991.

5 Juan Fernandez-Olivares, Luis Castillo, Juan a. Cózar, and Oscar García Pérez. Supporting clinical processes and decisions by hierarchical planning and scheduling. *Computational Intelligence*, 27(1):103–122, 2011.

6 John Fox and Richard Thomson Ma. Decision Support for Health Care : The PROforma Evidence Base. *Informatics in Primary Care*, 14(1):49–54, 2006.

7 D Isern and A Moreno. Computer-based Execution of Clinical Guidelines: a Review. *International Journal of Medical Informatics*, 77(12):787–808, 2008.

8 Mark A. Musen, Yuval Shahar, and Edward H. Shortliffe. Clinical decision-support systems. In EdwardH. Shortliffe and JamesJ. Cimino, editors, *Biomedical Informatics*, Health Informatics, pages 698–736. Springer New York, 2006.

9 Tiago Oliveira, Paulo Novais, and José Neves. Representation of Clinical Practice Guideline Components in OWL. In *Trends in Practical Applications of Agents and Multiagent Systems SE – 10*, volume 221 of *Advances in Intelligent Systems and Computing*, pages 77–85. Springer International Publishing, 2013.

10 A. K. Pani and G. P. Bhattacharjee. Temporal representation and reasoning in artificial intelligence: A review. *Mathematical and Computer Modelling*, 34(1-2):55–80, 2001.

11 Mor Peleg. Computer-interpretable Clinical Guidelines: A Methodological Review. *Journal of Biomedical Informatics*, 46(4):744–63, 2013.

12 Y Shahar, S Miksch, and P Johnson. The Asgaard Project: A Task-specific Framework for the Application and Critiquing of Time-oriented Clinical Guidelines. *Artificial intelligence in Medicine*, 14(1-2):29–51, 1998.

13 Paolo Terenziani, Carlo Carlini, and Stefania Montani. Towards a comprehensive treatment of temporal constraints in clinical guidelines. In *Proceedings of Ninth International Symposium on Temporal Representation and Reasoning 2002*, pages 20–27. IEEE, 2002.

14 Paolo Terenziani, Stefania Montani, Alessio Bottrighi, Mauro Torchio, Gianpaolo Molino, and Gianluca Correndo. The GLARE Approach to Clinical Guidelines: Main Features. *Studies in Health Technology and Informatics*, 101(3):162–6, 2004.

15 S W Tu, J R Campbell, J Glasgow, M A Nyman, R McClure, J McClay, C Parker, K M Hrabak, D Berg, T Weida, and Others. The SAGE Guideline Model: Achievements and Overview. *Journal of the American Medical Informatics Association*, 14(5):589, 2007.

16 Dongwen Wang, Mor Peleg, Samson W Tu, Aziz A Boxwala, Robert A Greenes, Vimla L Patel, and Edward H Shortliffe. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *International Journal of Medical Informatics*, 68(1-3):59–70, 2002.

Hunt for the Collapse of Semantics in Infinite Abstract Argumentation Frameworks*

Christof Spanring[1,2]

1 **Department of Computer Science**[†]**, University of Liverpool, UK**
c.spanring@liverpool.ac.uk
2 **Institute of Information Systems, TU Wien, Austria**

Abstract

In this work we discuss examples of infinite abstract argumentation frameworks (AFs). Our focus is mainly on existence of extensions of semantics such as semi-stable and stage semantics, as opposed to the collapse where some argumentation frameworks prevent any extension. We visit known examples from the literature and present novel variants. Finally, we also give insights into extension existence conditions.

1998 ACM Subject Classification F.2.2 Computations on discrete structures, F.4.1 Mathematical Logic, I.2 Artificial Intelligence

Keywords and phrases Abstract Argumentation, Infinity, Argumentation Semantics, Discrete Structures

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.70

1 Introduction

In everyday life we hardly ever think of dealing with actually infinite structures. Our time on earth may be complicated but it appears to be strictly finite, we deal with finite space, finite distances and finite cardinalities, i.e. natural numbers. Computer scientists in particular, tend to prefer working with finite structures, e.g. algorithms are supposed to terminate in a finite amount of time. Often enough infinity introduces odd behaviour and exceptions to the languages we use, and grew to love. For instance consider some countable language of finite words over some finite alphabet (e.g. English or C++). Think about effectively spelling an infinite word now. However, infinite structures actually are important even in our everyday life, see [16] for a fabulous overview in that matter.

In abstract argumentation, as introduced by Dung in his seminal paper [9], we break down the art of reaching consensus to abstract arguments and attacks. Due to the practical nature of argumentation most work in the literature restricts itself to the case of only finitely many arguments and attacks. Nonetheless already Dung discussed some non-finite cases and helpful definitions. For abstract argumentation in particular, several commonly used ways of instantiation naturally produce infinite structures [1, 7], and thus on the abstract level provide reason to investigate infinite frameworks.

We focus on range-based semantics and discuss conditions for existence of extensions, respectively examples where there is no extension. Section 2 can be seen as an introduction into argumentation, Section 3 presents known and novel examples, Section 4 closes with a final discussion.

* This research has been supported by FWF (project I1102).
† http://cgi.csc.liv.ac.uk/~christof

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 70–77

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

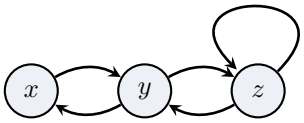


Figure 1 A simple AF as discussed in Examples 2 and 4.

2 Abstract Argumentation

Abstract Argumentation was introduced by Dung in [9], motivated by philosophical works, such as [14, 19], and further on used in various fields, ranging from legal reasoning [4], to non-monotonic logic [6], artificial intelligence [5] and others.

▶ **Definition 1.** An *argumentation framework (AF)* is an ordered pair $F = (A, R)$ where A is an arbitrary set of *arguments* and $R \subseteq A \times A$ is called the *attack* relation. For $(a, b) \in R$ we say that a attacks b, for $(a, b), (b, c) \in R$ we say that a defends c against b. Furthermore, for $S \subseteq A$ and $a \in A$ we say that a attacks S (or S attacks a) if for some $b \in S$ we have a attacks b (or b attacks a). We extend this notion also for $S, T \subseteq A$ accordingly. Finally, for $S \subseteq A$ we call $S^+ = S \cup \{a \in A \mid S \text{ attacks } a\}$ the range of S in F.

AFs frequently are visualized as a graphs where nodes reflect arguments and directed edges reflect attacks between arguments.

▶ **Example 2.** Consider the AF $F = (A, R)$ depicted in Figure 1. We have $A = \{x, y, z\}$ and $R = \{(x, y), (y, x), (y, z), (z, y), (z, z)\}$. Here the arguments could for instance refer to sentences such as x:*(everything is finite)*, y:*(infinity is real)*, z:*(reality is finite infinity)*.

Investigating some arbitrary AF we will consider sets of arguments, and investigate whether these sets appear to be justified under some principles, also called argumentation semantics. For a comprehensive introduction into argumentation semantics see [2].

▶ **Definition 3.** An *argumentation semantics* is a mapping from AFs to sets of arguments, where for any AF $F = (A, R)$ and semantics σ we have that if $S \in \sigma(F)$ then $S \subseteq A$. The members of $\sigma(F)$ are then called σ-*extensions* of F. By stating properties a specific extension has to fulfill, we will now define the semantics of interest for this work.

A set of arguments $S \subseteq A$ is called *conflict-free* (cf) if no member attacks any other member, i.e. $S \in cf(F)$ if for all $a, b \in S$ we have $(a, b) \notin R$. S is further called *admissible* (ad) if it defends itself against attacks from the outside, i.e. $S \in ad(F)$ if $S \in cf(F)$ and for any $a \in A$ such that a attacks S we have that also S attacks a. An extension $S \subseteq A$ is called

- *naive* (na), $S \in na(F)$ if $S \in cf(F)$ and there is no $S' \in cf(F)$ with $S \subsetneq S'$,
- *preferred* (pr), $S \in pr(F)$ if $S \in ad(F)$ and there is no $S' \in ad(F)$ with $S \subsetneq S'$,
- *stage* (sg), $S \in sg(F)$ if $S \in cf(F)$ and there is no $S' \in cf(F)$ with $S^+ \subsetneq S'^+$,
- *semi-stable* (sm), $S \in sm(F)$ if $S \in ad(F)$ and there is no $S' \in ad(F)$ with $S^+ \subsetneq S'^+$.

▶ **Example 4.** Consider the AF F from Example 2. We have $cf(F) = ad(F) = \{\emptyset, \{x\}, \{y\}\}$, $na(F) = pr(F) = \{\{x\}, \{y\}\}$, $sg(F) = sm(F) = \{\{y\}\}$. Observe that these equality relations do not hold for arbitrary AFs. However, for general AFs by definition we always have $sg \subseteq na \subseteq cf$ and $sm \subseteq pr \subseteq ad \subseteq cf$.

▶ **Definition 5.** An AF $F = (A, R)$ is called *finite* if $|F| := |A| < \infty$, it is called *infinite* if it is not finite. Regardless of whether F is finite or infinite it is called *finitary* [9] if each argument has only finitely many attackers, i.e. for all $a \in A$ we have $|\{b \in A \mid (b, a) \in R\}| < \infty$.

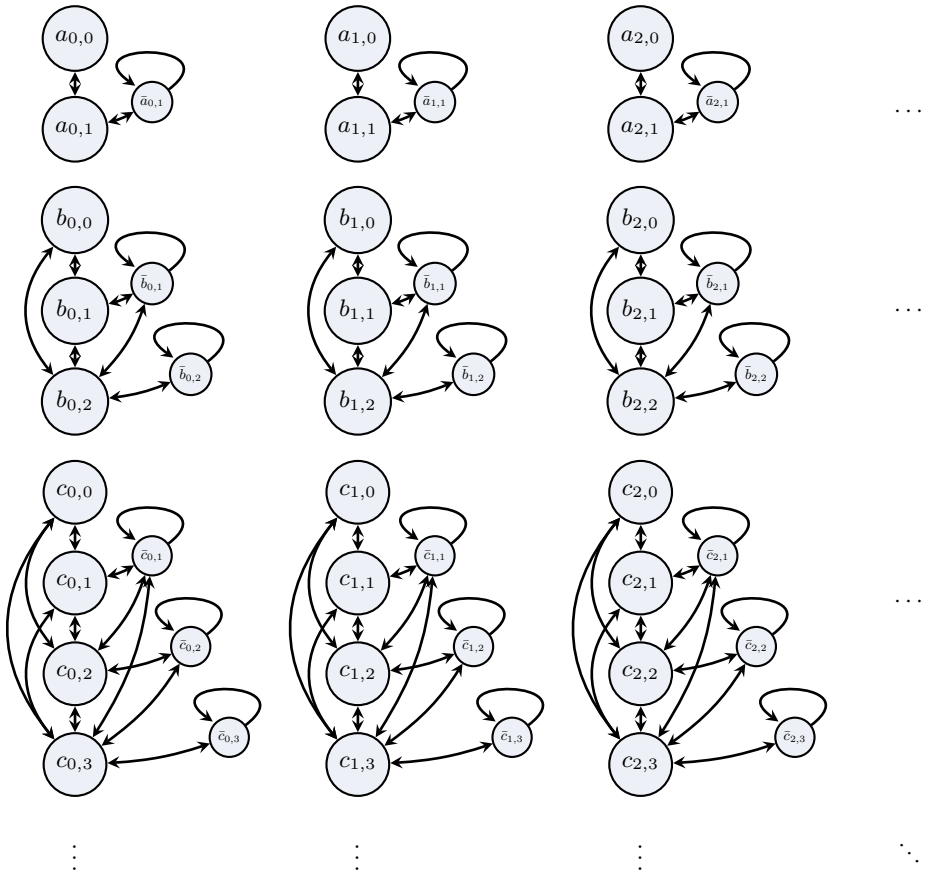


Figure 2 Transfinitely many steps might be necessary when constructing *sg* or *sm* extensions, cf. Example 6.

For finite AFs and the given semantics it might be that we sometimes receive only empty extensions ($\sigma(F) = \{\emptyset\}$), but for finite AFs at least there will always be extensions ($\sigma(F) \neq \emptyset$). If there are infinitely many arguments similar statements are not quite as obvious.

3 The infinite realm

In [17] it was shown that existence of *na* or *pr* extensions for arbitrary AFs is equivalent to the axiom of choice. For this work we assume that the axiom of choice holds and thus *na* and *pr* extensions exist for arbitrary AFs. Thus further on we will focus on *sg* and *sm* semantics. For range-maximality we have that [18, 8] discusses cases where no *sg* or *sm* extension exists, and [20, 3] discuss and prove existence conditions, including finitariness. In the following we will review examples and discuss variations. Let us first take a look at difficulties we might run into with naive approaches to constructing range-maximal sets.

▶ **Example 6** (Forest of Arguments)**.** Consider the AF $F = (A, R)$ depicted in Figure 2, where $A = \{a_{i,0}, a_{i,1}, \bar{a}_{i,1}, b_{i,0}, b_{i,1}, \bar{b}_{i,1}, b_{i,2}, \bar{b}_{i,2}, c_{i,0}, c_{i,1}, \bar{c}_{i,1}, c_{i,2}, \bar{c}_{i,2}, c_{i,3}, \bar{c}_{i,3}, \cdots \mid i \in \mathbb{N}\}$and $R = \{(x_{i,j}, x_{i,k}) \mid j \neq k\} \cup \{(x_{i,j}, \bar{x}_{i,k}), (\bar{x}_{i,k}, \bar{x}_{i,k}), (\bar{x}_{i,k}, x_{i,j}) \mid k \leq j\}$.

As in this AF all attacks are symmetric we have coinciding extension-sets of *cf* and *ad*, of *na* and *pr*, and of *sg* and *sm*. Consider the *na* set $S_0 = \{x_{i,0} \mid x \in \{a, b, c, \cdots\}, i \in \mathbb{N}\}$. If we replace $a_{0,0}$ with $a_{0,1}$ we receive $S_1 = (S_0 \cup \{a_{0,1}\}) \setminus \{a_{0,0}\}$. We further construct

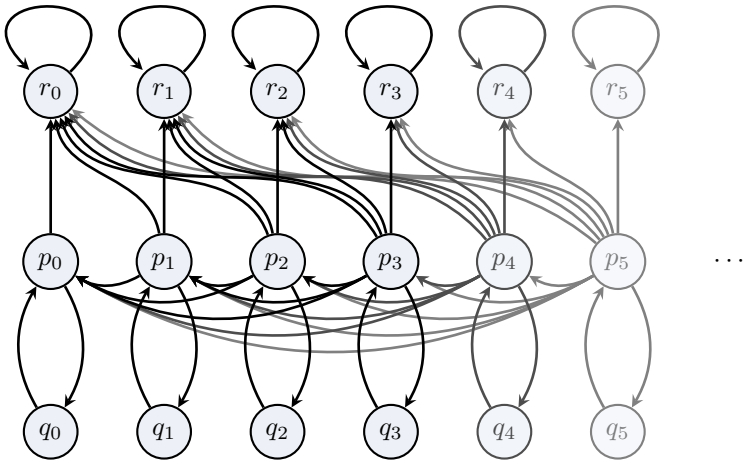


Figure 3 A first example without semi-stable or stage extensions, cf. Example 7.

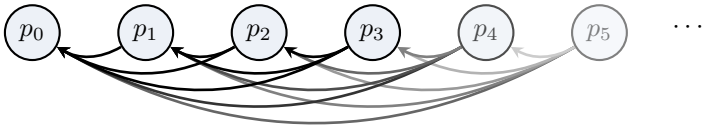


Figure 4 Minimal AF without stage extensions, cf. Example 8.

$S_2 = (S_1 \cup \{b_{0,1}\}) \setminus \{b_{0,0}\}$, $S_3 = (S_2 \cup \{c_{0,1}\}) \setminus \{c_{0,0}\}$ and so on. We receive an infinite chain $S_0, S_1, S_2, \cdots$ of range increasing *pr* extensions S_i. In this case the limit $T_0 = \{a \in A \mid$ *a occurs infinitely often in* $S_i\}$ is a *pr* extension with $S_i^+ \subseteq T_0^+$ for all S_i. However still T_0 is no *sm* extension as for instance for $T_1 = (T_0 \cup \{b_{0,2}) \setminus \{b_{0,1}\}$ we have $T_0^+ \subsetneq T_1^+$.

Thus just picking random sets with greater range can result in the need of transfinitely many steps, brute-force induction might not work. Observe that the given AF is finitary, as each connected component consists of only finitely many arguments. We leave it as an exercise for the interested reader to come up with a valid *sm* extension.

3.1 Preliminary Examples

We now discuss examples first introduced into abstract argumentation in [18] (Examples 7 and 8). We will use the term *collapse* to refer to some semantics not providing any extension for a given AF, i.e. if $\sigma(F) = \emptyset$ we say that σ collapses for F. The intuition of this wording is that existence of such AFs is problematic for modular approaches, i.e. if $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$ do not share any arguments and σ collapses for F_1, then σ also collapses for $F = F_1 \cup F_2 = (A_1 \cup A_2, R_1 \cup R_2)$ regardless of possible σ-extensions for F_2.

▶ **Example 7.** Consider the AF $F = (A, R)$ illustrated in Figure 3 with $A = \{p_i, q_i, r_i \mid i \in \mathbb{N}\}$ and $R = \{(p_i, q_i), (q_i, p_i), (p_i, r_i), (r_i, r_i) \mid i \in \mathbb{N}\} \cup \{(p_i, p_j), (p_i, r_j) \mid j < i\}$. We have as only *pr* and *na* extensions $S = \{q_i \mid i \in \mathbb{N}\}$ and for $n \in \mathbb{N}$ the sets $S_n = (S \cup \{p_n\}) \setminus \{q_n\}$, where for $i < j$ we have $S^+ \subsetneq S_i^+ \subsetneq S_j^+$. So in effect for any *pr* or *na* extension there is another one of larger range and thus *sm* and *sg* collapse.

▶ **Example 8.** Contained as a subframework in Example 7 is the AF $F = (A, R)$, as illustrated in Figure 4, with $A = \{p_i \mid i \in \mathbb{N}\}$ and $R = \{(p_i, p_j) \mid j < i\}$. Here the only admissible set is the empty set and hence $pr(F) = sm(F) = \{\emptyset\}$. The singletons p_i, on the other hand, are conflict-free and even serve as naive sets. For stage semantics, however, given $S_i = \{p_i\}$

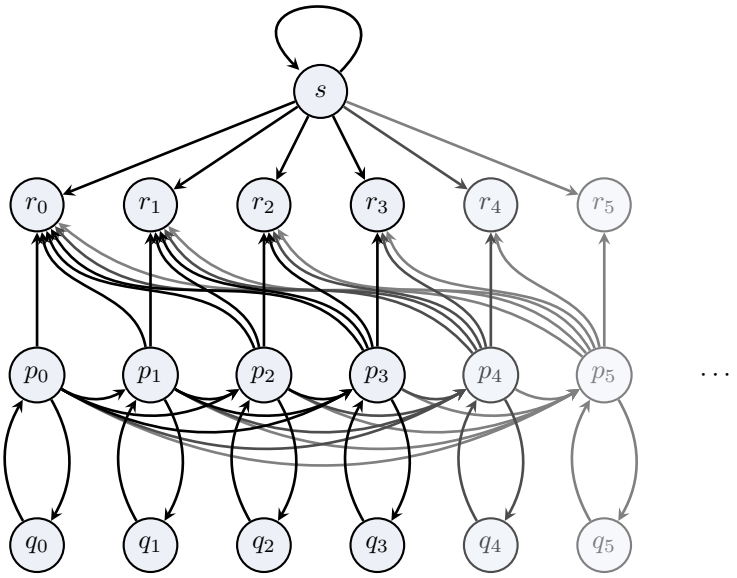


Figure 5 Avoiding self-attacks, no semi-stable but stage extensions, cf. Example 11.

we have that for instance S_{i+1} has larger range and thus *sg* collapses. So for this AF *sg* collapses but *sm* does not.

We now discuss minor modifications of Example 7 and restrictions such as Example 8.

▶ **Example 9.** Consider the AF $F = (A, R)$ from Example 7 and a symmetric version thereof $F' = (A, R')$ where $R' = R \cup \{(b, a) \mid (a, b) \in R\}$. For this AF *sg* and *sm* still collapse. However, observe that the restriction to the p_i, $F'_p = F'|_{\{p_i \mid i \in \mathbb{N}\}}$ now represents an AF where each $\{p_i\}$ is a *sg* and *sm* extension.

▶ **Example 10.** Consider the AF $F = (A, R)$ from Example 7. We now reverse the attacks between the p_i, $F' = (A, R')$, $R' = R \setminus \{(p_i, p_j) \mid j < i\} \cup \{(p_j, p_i) \mid j < i\}$. Again for this AF still *sm* and *sg* collapse. In other words the direction of the attacks between the p_i does not matter. Observe that the restriction $F'_p = F'|_{\{p_i \mid i \in \mathbb{N}\}}$ now represents an AF where $\{p_0\}$ serves as sole *sg* and *sm* extension.

3.2 Further advancements

In this section we present novel examples and ideas. Reconsider that AFs that do not provide any stage or semi-stable extension necessarily are non-finitary, i.e. there are arguments that are attacked by an infinite amount of other arguments. In this section we approach to reduce the amount of non-finitary arguments in such AFs.

▶ **Example 11.** Consider the AF $F = (A, R)$ from Example 10, which is Example 7 with reversed attacks between the p_i. We replace the self-attacks as illustrated in Figure 5, $F' = (A \cup \{s\}, R')$ with $R' = R \setminus \{(r_i, r_i) \mid i \in \mathbb{N}\} \cup \{(s, s), (s, r_i) \mid i \in \mathbb{N}\}$.

For admissibility based semantics it does not matter whether some argument is self-attacking, or attacked by some other argument it can not be defended against. Thus, we still have *sm* collapsing for F'. However, for instance $\{r_i, q_i \mid i \in \mathbb{N}\}$ serves as stage extension.

In all the previous examples there were pairwise attacks between all p_i, i.e. an infinite clique of arguments. In the following two examples we will approach a maximum of letting go of the infinitary components.

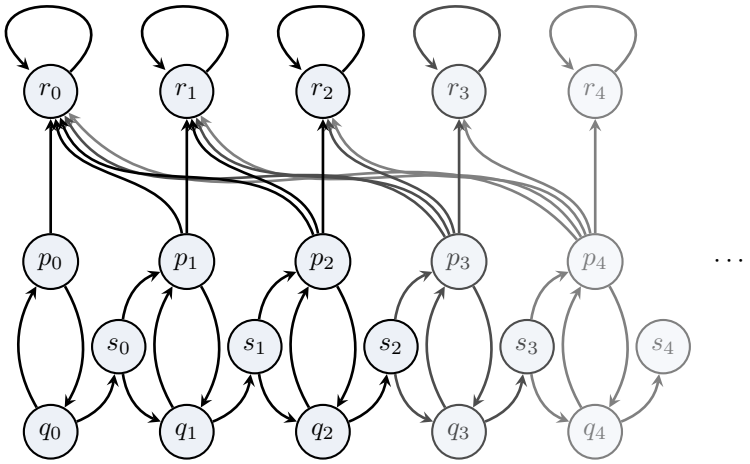


Figure 6 Avoiding infinite cliques, no semi-stable or stage extensions, cf. Example 12.

▶ **Example 12.** Consider the AF $F = (A, R)$ from Example 7. We replace the attacks between the p_i with an infinite chain of admissibility as illustrated in Figure 6, $F' = (A \cup \{s_i \mid i \in \mathbb{N}\}, R')$ where $R' = R \setminus \{(p_i, p_j)\} \cup \{(q_i, s_i), (s_i, p_{i+1}), (s_i, q_{i+1}) \mid i \in \mathbb{N}\}$.

Now observe that the only preferred extensions are $S_q = \{q_i \mid i \in \mathbb{N}\}$ and for each $n \in \mathbb{N}$ the sets $S_n = \{q_i, p_n, s_j \mid i < n, j \geq n\}$. Here p_n defends s_n, and accepting s_n for admissibility reasons means that we will accept each s_j for $j > n$. Again for $i < j$ we have $S_q^+ \subsetneq S_i^+ \subsetneq S_j^+$, and hence the collapse of semi-stable semantics.

For stage semantics, on the other hand, we need to consider more candidates, as also $S_p = \{p_i \mid i \in \mathbb{N}\}$ and any feasible combination between p_i, q_j and s_k serve as naive extensions. Now take some $S \in na(F)$ as given. If there is a maximal $n \in \mathbb{N}$ with $p_i \notin S$ for $i > n$, then S_{n+1} as defined above has larger range than S. Hence assume that for each $n \in \mathbb{N}$ there is some $i > n$ with $p_i \in S$. We conclude that for some $m \in \mathbb{N}$ we have both $s_m \notin S^+$ and one of $p_{m+1} \in S$ or $q_{m+1} \in S$. We construct $S' = \{q_j \mid j \leq m\} \cup (S \cap \{p_i, q_i, s_i \mid i > m\})$. By construction $S^+ \subsetneq S'^+$, and hence stage semantics collapses for this AF as well.

Now that we have seen a vast amount of examples illustrating how close we can get to finitariness while keeping the collapse for stage semantics, we take one step further for semi-stable semantics.

▶ **Example 13.** Consider AF $F = (A, R)$ illustrated in Figure 7, with $A = \{x_i, y_i, z_i \mid i \in \mathbb{N}\}$ and $R = \{(z_i, z_i), (z_i, y_i), (x_i, y_i), (x_i, z_0), (y_i, x_i), (y_i, z_{i+1}) \mid i \in \mathbb{N}\}$. Observe that only z_0 violates the finitary condition here.

We have as only preferred extensions the set $S_x = \{x_i \mid i \in \mathbb{N}\}$ and for each $n \in \mathbb{N}$ the sets $S_n = \{x_i, y_j \mid j \leq n, i > n\}$. Again for $i < j$ we have $S_x^+ \subsetneq S_i^+ \subsetneq S_j^+$ and hence semi-stable semantics collapses. For stage semantics, however, the set $S_y = \{y_i \mid i \in \mathbb{N}\}$ is maximal in range, as only $z_0 \notin S_y^+$, but attacking z_0 means including x_j for some j and thus not attacking z_{j+1}.

For all known examples of AFs where stage semantics is collapsing we have that there is an infinite amount of arguments with infinitely many attackers. For semi-stable semantics one such argument suffices. It appears that the collapse of stage semantics requires naive range-increasing extension chains that gradually loose arguments but keep attacking them and their range.

Further for all known *sg*-collapsing examples we have that removal of a finite amount of arguments does not affect the collapse. Further insight from techniques used in [3] appears

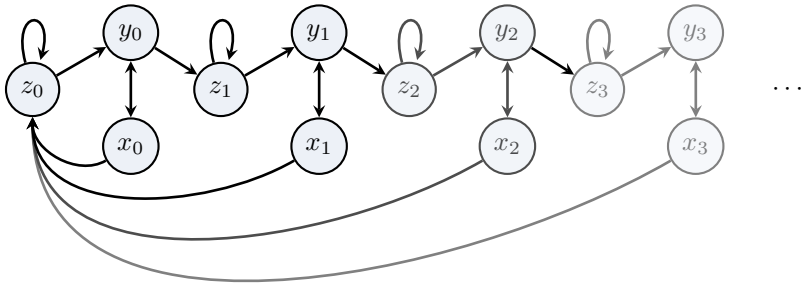


Figure 7 Some minimal AF illustrating the collapse of semi-stable semantics, cf. Example 13.

to suggest that for *sg* this property in general holds. In more detail we claim that for any *sg* extension in any AF each member of a certain class of smaller AFs contains a corresponding extension. It is neither purpose nor objective of this paper to give a proof of this rather technical claim. However especially in the context of collapsing semantics and the idea of reducing non-finitary arguments in mind we present the following conjecture.

▶ **Conjecture 14.** *If stage semantics collapses, then there is an infinite amount of arguments with infinitely many attackers.*

As suggested technique for proving Conjecture 14 we suggest standard induction over the number of (non-finitary) arguments. Reconsider Example 13 and stage semantics. Starting with e.g. S_0 and clutching up the chain of the S_i we receive as limit the set $S_y = \{y_i \mid i \in \mathbb{N}\}$, which is a stage extension not having z_0 in range. Anyhow, as $z_0 \notin S_y^+$, but $z_0 \in S_i$ for all $i \in \mathbb{N}$ we have a shift in range in the limit step. This is merely a hint on why Conjecture 14 is hard to prove.

4 Discussion and Future Work

We approached listing and classifying known and novel examples of AFs where stage or semi-stable semantics collapse. As the only known guarantee of existence for both semantics is an AF being finitary (see [20, 3]), the main effort was to reduce the impact of non-finitariness to a minimum. In Example 13 there is only one argument with infinitely many attackers. In this sense of minimizing non-finitary arguments, we completed the picture of collapse for semi-stable semantics, and conjectured completion for stage semantics.

Intertranslatability [12, 11] and signature [10] have shown to be valuable instruments for the investigation and comparison of (finite) AFs and semantics. In regards of infinite AFs the possible collapse of semi-stable and stage semantics is of interest on its own. On the one hand stage and stable semantics might produce the same extension-sets. On the other hand the collapse of semi-stable semantics immediately distinguishes it from preferred semantics, as opposed to the finite case. As infinite AFs have not been studied in the intertranslatability or signature context yet, also a comparison of the discussed and other semantics might still yield interesting results.

Immanent future work is further classification of conditions under which presented semantics might collapse or are ensured not to collapse. Graph-theoretical properties (other than symmetry, cf. Example 9) might be of interest. However, it seems to be more useful to investigate properties naturally induced by environments making use of argumentation. Promising candidates are any forms of instantiated argumentation, e.g. [13, 15] or [7].

References

1 Pietro Baroni, Federico Cerutti, Paul E. Dunne, and Massimiliano Giacomin. Automata for infinite argumentation structures. *Artif. Intell.*, 203:104–150, 2013.

2 Pietro Baroni and Massimiliano Giacomin. Semantics of abstract argument systems. In Iyad Rahwan and Guillermo Ricardo Simari, editors, *Argumentation in Artificial Intelligence*, chapter 2, pages 25–44. Springer, 2009.

3 Ringo Baumann and Christof Spanring. Infinite argumentation frameworks. *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, pages 281–295, 2015.

4 Trevor J. M. Bench-Capon, Katie Atkinson, and Alison Chorley. Persuasion and value in legal argument. *J. Log. Comput.*, 15(6):1075–1097, 2005.

5 Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

6 Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109–145, 2009.

7 Martin W. A. Caminada and Nir Oren. Grounded semantics and infinitary argumentation frameworks. In *Procs. of the 26th Benelux Conference on Artificial Intelligence (BNAIC'14)*, 2014.

8 Martin W. A. Caminada and Bart Verheij. On the existence of semi-stable extensions. In *Procs. of the 22nd Benelux Conference on Artificial Intelligence (BNAIC'10)*, 2010.

9 Phan Minh Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

10 Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 72–81. AAAI Press, 2014.

11 Wolfgang Dvořák and Christof Spanring. Comparing the expressiveness of argumentation semantics. In Bart Verheij, Stefan Szeider, and Stefan Woltran, editors, *Proceedings of the 4th Conference on Computational Models of Argument (COMMA 2012)*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 261–272. IOS Press, 2012.

12 Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. *J. Artif. Intell. Res. (JAIR)*, 41:445–475, 2011.

13 Nikos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artif. Intell.*, 175(9):1479–1497, 2011.

14 Charles Leonard Hamblin. *Fallacies*. Methuen London, 1970.

15 Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

16 Rudy Rucker. *Infinity and the Mind: The Science and Philosophy of the Infinite (Princeton Science Library)*. Princeton University Press, 2004.

17 Christof Spanring. Axiom of choice, maximal independent sets, argumentation and dialogue games. *2014 Imperial College Computing Student Workshop*, pages 91–98, 2014.

18 Bart Verheij. Deflog: on the logical interpretation of prima facie justified assumptions. *J. Log. Comput.*, 13(3):319–346, 2003.

19 Douglas N Walton. *Logical Dialogue-Games*. University Press of America, Lanham, Maryland, 1984.

20 Emil Weydert. Semi-stable extensions for infinite frameworks. In *Procs. of the 23rd Benelux Conference on Artificial Intelligence (BNAIC'11)*, pages 336–343, 2011.

Checking WECTLK Properties of TRWISs via SMT-based Bounded Model Checking* †

Agnieszka M. Zbrzezny and Andrzej Zbrzezny

IMCS, Jan Długosz University
Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
{agnieszka.zbrzezny,a.zbrzezny}@ajd.czest.pl

Abstract

In this paper, we present a Satisfiability Modulo Theory based (SMT-based) bounded model checking (BMC) method for Timed Real-Weighted Interpreted Systems and for the existential fragment of the Weighted Epistemic Computation Tree Logic. SMT-based bounded model checking consists in translating the existential model checking problem for a modal logic and for a model to the satisfiability problem of a quantifier-free first-order formula. We have implemented the SMT-BMC method and performed the BMC algorithm on Timed Weighted Generic Pipeline Paradigm benchmark. The preliminary experimental results demonstrate the feasibility of the method. To perform the experiments, we used the state of the art SMT-solver Z3.

1998 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases SMT, Timed Real-Weighted Interpreted Systems, Bounded Model Checking

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.78

1 Introduction

The formalism of *interpreted systems* (ISs) was introduced in [1] to model multi-agent systems (MASs) [6], which are intended for reasoning about the agents' epistemic and temporal properties. *Timed interpreted systems* (TIS) was proposed in [8] to extend interpreted systems in order to make possible reasoning about real-time aspects of MASs. The formalism of weighted interpreted systems (WISs) [9] extends ISs to make the reasoning possible about not only temporal and epistemic properties, but also agents's quantitative properties.

Multi-agent systems (MASs) are composed of many intelligent agents that interact with each other. The agents can share a common goal or they can pursue their own interests. Also, the agents may have a deadline or other timing constraints to achieve intended targets. As it was shown in [1], knowledge is a useful concept for analysing the information state and the behaviour of agents in multi-agent systems. In this paper, we consider the existential fragment of a weighted epistemic computation tree logic (WECTLK) interpreted over Timed Real-Weighted Interpreted Systems (TRWISs).

To the best of our knowledge, there is no work that considers SMT-based BMC methods to check multi-agent systems modelled by means of timed weighted interpreted systems. Thus, in this paper such a method is offered.

* Partly supported by National Science Centre under the grant No. 2014/15/N/ST6/05079.
† A longer version of this paper will appear in the proceedings of the 17th Portuguese Conference on Artificial Intelligence (EPIA'2015) in September 2015.

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 78–86

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We do not compare our results with other model checkers for MASs, e.g. MCMAS [4] or MCK [2], simply because they do not support the WECTLK language and the timed weighted interpreted systems.

Firstly, we define and implement an SMT-based BMC method for WECTLK and for TRWISs. Secondly, we report on the initial experimental evaluation of our SMT-based BMC method. To this aim, we use a scalable benchmark: the *timed weighted generic pipeline paradigm* [7, 9].

2 Preliminaries

Let $\mathbb{N}$ be a set of natural numbers, $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$, $\mathbb{R}$ be the set of non-negative real numbers, and $\mathcal{X}$ be a finite set of non-negative natural variables, called *clocks* ranging over a set of non-negative natural numbers. A clock valuation is a function $v : \mathcal{X} \to \mathbb{N}$ that assigns to each clock $x \in \mathcal{X}$ a non-negative natural value $v(x)$. A set of all the clock valuations is denoted by $\mathbb{N}^{|\mathcal{X}|}$. The valuation $v' = v[\mathcal{X}' := 0]$, for $\mathcal{X}' \subseteq \mathcal{X}$ is defined as: $\forall_{x \in \mathcal{X}'} v'(x) = 0$ and $\forall_{x \in \mathcal{X} \setminus \mathcal{X}'} v'(x) = v(x)$. For $\delta \in \mathbb{N}$, $v + \delta$ denotes the valuation that assigns the value $v(x) + \delta$ to each clock x.

The grammar $\varphi := \mathbf{true} \mid x < c \mid x \leq c \mid x = c \mid x \geq c \mid x > c \mid \varphi \wedge \varphi$ generates the set $\mathcal{C}(\mathcal{X})$ of clock constraints over $\mathcal{X}$, where $x \in \mathcal{X}$ and $c \in \mathbb{N}$. A clock valuation v satisfies a clock constraint φ, written as $v \models \varphi$, iff φ evaluates to be true using the clock values given by v.

Let c_{max} be a constant and $v, v' \in \mathbb{N}^{|\mathcal{X}|}$ two clock valuation. We say that $v \simeq v'$ iff the following condition holds for each $x \in \mathcal{X}$: $v(x) > c_{max}$ and $v'(x) > c_{max}$ or $v(x) \leq c_{max}$ and $v'(x) \leq c_{max}$ and $v(x) = v'(x)$ The clock valuation v' such that for each clock $x \in \mathcal{X}$, $v'(x) = v(x) + 1$ if $v(x) \leq c_{max}$, and $v'(x) = c_{max} + 1$ otherwise, is called a time successor of v (written $succ(v)$).

TRWISs. Let $Ag = \{1, \ldots, n\}$ denotes a non-empty and finite set of agents, and $\mathcal{E}$ be a special agent that is used to model the environment in which the agents operate and $\mathcal{PV} = \bigcup_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}} \mathcal{PV}_{\mathbf{c}}$ be a set of propositional variables, such that $\mathcal{PV}_{\mathbf{c}_1} \bigcap \mathcal{PV}_{\mathbf{c}_2} = \emptyset$ for all $\mathbf{c}_1, \mathbf{c}_2 \in Ag \cup \{\mathcal{E}\}$. The *timed real-weighted interpreted system* (TRWIS) is a tuple $(\{L_{\mathbf{c}}, Act_{\mathbf{c}}, \mathcal{X}_{\mathbf{c}}, P_{\mathbf{c}}, t_{\mathbf{c}}, \mathcal{V}_{\mathbf{c}}, \mathcal{I}_{\mathbf{c}}, d_{\mathbf{c}}\}_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}}, \iota)$, where $L_{\mathbf{c}}$ is a non-empty set of *local states* of the agent $\mathbf{c}$, $S = L_1 \times \ldots \times L_n \times L_{\mathcal{E}}$ is the set of all global states, $\iota \subseteq S$ is a non-empty set of initial states, $Act_{\mathbf{c}}$ is a non-empty set of *possible actions* of the agent $\mathbf{c}$, $Act = Act_1 \times \ldots \times Act_n \times Act_{\mathcal{E}}$ is the set of *joint actions*, $\mathcal{X}_{\mathbf{c}}$ is a non-empty set of *clocks*, $P_{\mathbf{c}} : L_{\mathbf{c}} \to 2^{Act_{\mathbf{c}}}$ is a *protocol function*, $t_{\mathbf{c}} : L_{\mathbf{c}} \times \mathcal{C}(\mathcal{X}_{\mathbf{c}}) \times 2^{\mathcal{X}_c} \times Act \to L_{\mathbf{c}}$ is a (partial) *evolution function*, $\mathcal{V}_{\mathbf{c}} : L_{\mathbf{c}} \to 2^{\mathcal{PV}}$ is a *valuation function* assigning to each local state a set of propositional variables that are assumed to be true at that state, $\mathcal{I}_{\mathbf{c}}$: $L_{\mathbf{c}} \to \mathcal{C}(\mathcal{X}_{\mathbf{c}})$ is an *invariant function*, that specifies an amount of time the agent $\mathbf{c}$ may spend in a given local state, and $d_{\mathbf{c}} : Act_{\mathbf{c}} \to \mathbb{R}$ is a *weight function*.

For a given TRWIS we define a *timed real-weighted model* (or a *model*) as a tuple $\mathcal{M} = (Act, S, \iota, T, \mathcal{V}, d)$, where:

- $Act = Act_1 \times \ldots \times Act_n \times Act_{\mathcal{E}}$ is the set of all the joint actions,
- $S = (L_1 \times \mathbb{N}^{|\mathcal{X}_1|}) \times \ldots \times (L_n \times \mathbb{N}^{|\mathcal{X}_n|)}) \times (L_{\mathcal{E}} \times \mathbb{N}^{|\mathcal{X}_{\mathcal{E}}|})$ is the set of all the *global states*,
- $\iota = (\iota_1 \times \{0\}^{|\mathcal{X}_1|}) \times \ldots \times (\iota_n \times \{0\}^{|\mathcal{X}_n|}) \times (\iota_{\mathcal{E}} \times (\{0\}^{|\mathcal{X}_{\mathcal{E}}|})$ is the set of all the *initial* global states,
- $\mathcal{V} : S \to 2^{\mathcal{PV}}$ is the valuation function defined as $\mathcal{V}(s) = \bigcup_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}} \mathcal{V}_{\mathbf{c}}(l_{\mathbf{c}}(s))$, $T \subseteq S \times (Act \cup \mathbb{N}) \times S$ is a transition relation defined by action and time transitions. For $a \in Act$ and $\delta \in \mathbb{N}$:

1. action transition: $(s,a,s') \in T$ (or $s \xrightarrow{a} s'$) iff for all $\mathbf{c} \in Ag \cup \mathcal{E}$, there exists a local transition $t_{\mathbf{c}}(l_{\mathbf{c}}(s), \varphi_{\mathbf{c}}, \mathcal{X}', a) = l_{\mathbf{c}}(s')$ such that $v_{\mathbf{c}}(s) \models \varphi_{\mathbf{c}} \wedge \mathcal{I}(l_{\mathbf{c}}(s))$ and $v'_{\mathbf{c}}(s') = v_{\mathbf{c}}(s)[\mathcal{X}' := 0]$ and $v'_{\mathbf{c}}(s') \models \mathcal{I}(l_{\mathbf{c}}(s'))$;
2. time transition $(s,\delta,s') \in T$ iff for all $\mathbf{c} \in Ag \cup \mathcal{E}$, $l_{\mathbf{c}}(s) = l_{\mathbf{c}}(s')$ and $v'_{\mathbf{c}}(s') = v_{\mathbf{c}}(s) + \delta$ and $v'_{\mathbf{c}}(s') \models \mathcal{I}(l_{\mathbf{c}}(s'))$.

- $d : Act \to \mathbb{R}$ is the "joint" weight function defined as follows: $d((a_1, \ldots, a_n, a_{\mathcal{E}})) = d_1(a_1) + \ldots + d_n(a_n) + d_{\mathcal{E}}(a_{\mathcal{E}})$.

Given a TRWIS, one can define for any agent $\mathbf{c}$ the indistinguishability relation $\sim_{\mathbf{c}} \subseteq S \times S$ as follows: $s \sim_{\mathbf{c}} s'$ iff $l_{\mathbf{c}}(s') = l_{\mathbf{c}}(s)$ and $v_{\mathbf{c}}(s') \simeq v_{\mathbf{c}}(s)$

A run in $\mathcal{M}$ is an infinite sequence $\rho = s_0 \xrightarrow{\delta_0,a_0} s_1 \xrightarrow{\delta_1,a_1} s_2 \xrightarrow{\delta_2,a_2} \ldots$ of global states such that the following conditions hold for all $i \in \mathbb{N} : s_i \in S, a_i \in Act, \delta_i \in \mathbb{N}_+$, and there exists $s'_i \in S$ such that $(s_i, \delta, s'_i) \in T$ and $(s_i, a, s_{i+1}) \in T$. Notice that the definition of a run does not permit two consecutive joint actions to be performed one after the other, i.e., between each two joint actions some time must pass; such a run is called *strongly monotonic*.

Abstract model. Let $\mathbb{D}_{\mathbf{c}} = \{0, \ldots, c_{\mathbf{c}} + 1\}$ with $c_{\mathbf{c}}$ be the largest constant appearing in any enabling condition or state invariants of agent $\mathbf{c}$ and $\mathbb{D} = \bigcup_{\mathbf{c} \in Ag \cup \mathcal{E}} \mathbb{D}_{\mathbf{c}}^{|\mathcal{X}_{\mathbf{c}}|}$. A tuple $\widehat{\mathcal{M}} = (Act, \widehat{S}, \widehat{\iota}, \widehat{T}, \widehat{\mathcal{V}}, d)$, is an *abstract model*, where $\widehat{\iota} = \prod_{\mathbf{c} \in Ag \cup \mathcal{E}} \iota_{\mathbf{c}} \times \{0\}^{|\mathcal{X}_{\mathbf{c}}|}$ is the set of all the initial global states, $\widehat{S} = \prod_{\mathbf{c} \in Ag \cup \mathcal{E}} L_{\mathbf{c}} \times \mathbb{D}_{\mathbf{c}}^{|\mathcal{X}_{\mathbf{c}}|}$ is the set of all the abstract global states. $\widehat{\mathcal{V}} : \widehat{S} \to 2^{\mathcal{PV}}$ is the valuation function such that: $p \in \widehat{\mathcal{V}}(s)$ iff $p \in \bigcup_{\mathbf{c} \in Ag \cup \mathcal{E}} \widehat{\mathcal{V}}_{\mathbf{c}}(l_{\mathbf{c}}(s))$ for all $p \in \mathcal{PV}$; and $\widehat{T} \subseteq \widehat{S} \times (Act \cup \tau) \times \widehat{S}$. Let $a \in Act$. Then,

1. Action transition: $(s,a,s') \in \widehat{T}$ iff $\forall_{\mathbf{c} \in Ag} \exists_{\phi_{\mathbf{c}} \in \mathcal{C}(\mathcal{X}_{\mathbf{c}})} \exists_{\mathcal{X}'_{\mathbf{c}} \subseteq \mathcal{X}_{\mathbf{c}}} (t_{\mathbf{c}}(l_{\mathbf{c}}(s), \phi_{\mathbf{c}}, \mathcal{X}'_{\mathbf{c}}, a) = l_{\mathbf{c}}(s')$ and $v_{\mathbf{c}} \models \phi_{\mathbf{c}} \wedge \mathcal{I}(l_{\mathbf{c}}(s))$ and $v'_{\mathbf{c}}(s') = v_{\mathbf{c}}(s)[\mathcal{X}'_{\mathbf{c}} := 0]$ and $v'_{\mathbf{c}}(s') \models \mathcal{I}(l_{\mathbf{c}}(s')))$
2. Time transition: $(s,\tau,s') \in \widehat{T}$ iff $\forall_{\mathbf{c} \in Ag \cup \mathcal{E}} (l_{\mathbf{c}}(s) = l_{\mathbf{c}}(s'))$ and $v_{\mathbf{c}}(s) \models \mathcal{I}(l_{\mathbf{c}}(s))$ and $succ(v_{\mathbf{c}}(s)) \models \mathcal{I}(l_{\mathbf{c}}(s)))$ and $\forall_{\mathbf{c} \in Ag} (v'_{\mathbf{c}}(s') = succ(v_{\mathbf{c}}(s')))$ and $(v'_{\mathcal{E}}(s') = succ(v_{\mathcal{E}}(s)))$.

Given an abstract model one can define for any agent $\mathbf{c}$ the indistinguishability relation $\sim_{\mathbf{c}} \subseteq \widehat{S} \times \widehat{S}$ as follows: $s \sim_{\mathbf{c}} s'$ iff $l_{\mathbf{c}}(s') = l_{\mathbf{c}}(s)$ and $v_{\mathbf{c}}(s') = v_{\mathbf{c}}(s)$. A path π in an abstract model is a sequence $s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} s_2 \xrightarrow{b_3} \ldots$ of transitions such that for each $i \leq 1$, $b_i \in Act \cup \{\tau\}$ and $b_1 = \tau$ and for each two consecutive transitions at least one of them is a time transition. Next, $\pi[j..m]$ denotes the finite sequence $s_j \xrightarrow{\delta_{j+1},a_{j+1}} s_{j+1} \xrightarrow{\delta_{j+2},a_{j+2}} \ldots s_m$ with $m - j$ transitions and $m - j + 1$ states, and $D\pi[j..m]$ denotes the (cumulative) weight of $\pi[j..m]$ that is defined as $d(a_{j+1}) + \ldots + d(a_m)$ (hence 0 when $j = m$). The set of all the paths starting at $s \in S$ is denoted by $\Pi(s)$, and the set of all the paths starting at an initial state is denoted by $\Pi = \bigcup_{s^0 \in \widehat{\iota}} \Pi(s^0)$.

WECTLK. The WECTLK has been defined in [7] as the existential fragment of the weighted CTLK with integer cost constraints on *all* temporal modalities. We extend WECTLK logic by adding non-negative real cost constraints. In the syntax of WECTLK we assume the following: $p \in \mathcal{PV}$ is an atomic proposition, $\mathbf{c} \in Ag$, $\Gamma \subseteq Ag$, I is an interval in $\mathbb{R}$ of the form: $[a, \infty)$ and $[a, b)$, for $a, b \in \mathbb{N}$ and $a \neq b$. Moreover, hereafter, $\mathbf{right}(I)$ denotes the right end of the interval I. The WECTLK formulae are defined by the following grammar:

$$\varphi ::= \mathbf{true} \,|\, \mathbf{false} \,|\, p \,|\, \neg p \,|\, \varphi \vee \varphi \,|\, \varphi \wedge \varphi \,|\, \mathbf{EX}_I \varphi \,|\, \mathbf{E}(\varphi \mathbf{U}_I \varphi) \,|\, \mathbf{EG}_I \varphi \,|\, \overline{\mathrm{K}}_{\mathbf{c}} \varphi.$$

A WECTLK formula φ is *true* in an abstract model $\widehat{\mathcal{M}}$ (in symbols $\widehat{\mathcal{M}} \models \varphi$) iff $\widehat{\mathcal{M}}, s^0 \models \varphi$ for some $s^0 \in \widehat{\iota}$ (i.e., φ is true at some initial state of the abstract model $\widehat{\mathcal{M}}$). For every $s \in \widehat{S}$ the relation $\models$ is defined inductively as follows:

- $\widehat{\mathcal{M}}, s \models \mathbf{true}$, $\widehat{\mathcal{M}}, s \not\models \mathbf{false}$, $\widehat{\mathcal{M}}, s \models p$ iff $p \in \widehat{\mathcal{V}}(s)$, $\widehat{\mathcal{M}}, s \models \neg p$ iff $p \notin \widehat{\mathcal{V}}(s)$,
- $\widehat{\mathcal{M}}, s \models \alpha \wedge \beta$ iff $\widehat{\mathcal{M}}, s \models \alpha$ and $\widehat{\mathcal{M}}, s \models \beta$, $\widehat{\mathcal{M}}, s \models \alpha \vee \beta$ iff $\widehat{\mathcal{M}}, s \models \alpha$ or $\widehat{\mathcal{M}}, s \models \beta$
- $\widehat{\mathcal{M}}, s \models \mathbf{EX}_I \alpha$ iff $(\exists \pi \in \Pi(s))(D\pi[0..1] \in I$ and $\widehat{\mathcal{M}}, \pi(1) \models \alpha)$,
- $\widehat{\mathcal{M}}, s \models \mathbf{EG}_I \alpha$ iff $(\exists \pi \in \Pi(s))(\forall i \geq 0)(D\pi[0..i] \in I$ implies $\widehat{\mathcal{M}}, \pi(i) \models \beta)$,
- $\widehat{\mathcal{M}}, s \models \mathbf{E}(\alpha \mathbf{U}_I \beta)$ iff $(\exists \pi \in \Pi(s))(\exists i \geq 0)(D\pi[0..i] \in I$ and $\widehat{\mathcal{M}}, \pi(i) \models \beta$ and $(\forall j < i)\widehat{\mathcal{M}}, \pi(j) \models \alpha)$,
- $\widehat{\mathcal{M}}, s \models \overline{\mathrm{K}}_{\mathbf{c}} \alpha$ iff $(\exists \pi \in \Pi)\ (\exists i \geq 0)(s \sim_{\mathbf{c}} \pi(i)$ and $\widehat{\mathcal{M}}, \pi(i) \models \alpha)$.

3 SMT-based Bounded Model Checking

In this section, we present an outline of the bounded semantics for WECTLK and define an SMT-based BMC method for WECTLK, which is based on the BMC encoding presented in [7]. As usual, we start by defining k-paths and $(k,l) - loops$. Next, we define a bounded semantics, which is used for the translation to SMT.

Bounded semantics. Let $\widehat{\mathcal{M}}$ be an abstract model, and $k \in \mathbb{N}$ a bound. A *k-path* π_k is a finite sequence $s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \ldots \xrightarrow{b_k} s_k$ of transitions such that for each $1 \leq i \leq k$, $b_i \in Act \cup \{\tau\}$ and $b_1 = \tau$ and for each two consecutive transitions at least one is a time transition. A k-path π_k is a *loop* if $l < k$ and $\pi(k) = \pi(l)$. Note that if a k-path π_k is a loop, then it represents the infinite path of the form uv^{ω}, where $u = (s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \ldots \xrightarrow{b_l} s_l)$ and $v = (s_{l+1} \xrightarrow{b_{l+2}} \ldots \xrightarrow{b_k} s_k)$. $\Pi_k(s)$ denotes the set of all the k-paths of $\widehat{\mathcal{M}}$ that start at s, and $\Pi_k = \bigcup_{s^0 \in \widehat{\iota}} \Pi_k(s^0)$.

The bounded satisfiability relation $\models_k$ which indicates k-truth of a WECTLK formula in the abstract model $\widehat{\mathcal{M}}$ at some state s of $\widehat{\mathcal{M}}$ is also defined in [7]. A WECTLK formula φ is *k-true* in the abstract model $\widehat{\mathcal{M}}$ (in symbols $\widehat{\mathcal{M}} \models_k \varphi$) iff φ is k-true at some initial state of the abstract model $\widehat{\mathcal{M}}$.

The *model checking problem* asks whether $\widehat{\mathcal{M}} \models \varphi$, but the *bounded model checking problem* asks whether there exists $k \in \mathbb{N}$ such that $\widehat{\mathcal{M}} \models_k \varphi$. The following theorem states that for a given abstract model and a WECTLK formula there exists a bound k such that the model checking problem ($\widehat{\mathcal{M}} \models \varphi$) can be reduced to the bounded model checking problem ($\widehat{\mathcal{M}} \models_k \varphi$).

▶ **Theorem 1.** *Let $\widehat{\mathcal{M}}$ be the abstract model and φ a WECTLK formula. Then, the following equivalence holds: $\widehat{\mathcal{M}} \models \varphi$ iff there exists $k \geq 0$ such that $\widehat{\mathcal{M}} \models_k \varphi$.*

Proof. The theorem can be proved by induction on the length of the formula φ (for details one can see [7]). ◀

Translation to SMT. Let $\widehat{\mathcal{M}}$ be an abstract model, φ a WECTLK formula, and $k \geq 0$ a bound. The presented SMT encoding of the BMC problem for WECTLK and for TRWIS is based on the SAT encoding of the same problem [10, 9], and it relies on defining the quantifier-free first-order formula: $[\widehat{\mathcal{M}}, \varphi]_k := [\widehat{\mathcal{M}}^{\varphi,\iota}]_k \wedge [\varphi]_{\widehat{\mathcal{M}},k}$ that is satisfiable if and only if $\widehat{\mathcal{M}} \models_k \varphi$ holds.

Let $\mathbf{c} \in Ag \cup \{\mathcal{E}\}$. The definition of the formula $[\widehat{\mathcal{M}}, \varphi]_k$ assumes that

- each global state $s \in \widehat{S}$ is represented by a valuation of a *symbolic state* $\overline{\mathbf{w}} = ((w_1, v_1), \ldots, (w_n, v_n), (w_{\mathcal{E}}, v_{\mathcal{E}}))$ that consists of *symbolic local states* and each symbolic local state $w_{\mathbf{c}}$ is a pair $(w_{\mathbf{c}}, v_{\mathbf{c}})$ of individual variables ranging over the natural numbers, in which the first element represents a local state of the agent $\mathbf{c}$, and the second represents a clock valuation;

- each joint action $a \in Act$ is represented by a valuation of a *symbolic action* $\overline{a} = (a_1, \ldots, a_n, a_{\mathcal{E}})$ that consists of *symbolic local actions* and each symbolic local action $a_{\mathbf{c}}$ is an individual variable ranging over the natural numbers;
- each sequence of weights associated with the joint action is represented by a valuation of a *symbolic weights* $\overline{d} = (d_1, \ldots, d_{n+1})$ that consists of *symbolic local weights* and each symbolic local weight $d_{\mathbf{c}}$ is an individual variable ranging over the natural numbers.

The formula $[\widehat{\mathcal{M}}^{\varphi,\widehat{\iota}}]_k$ encodes a rooted tree of $k-$paths of the abstract model $\widehat{\mathcal{M}}$. The number of branches of the tree depends on the value of $f_k : \text{WECTLK} \to \mathbb{N}$ which is an auxiliary function defined in [7]. The formula $[\widehat{\mathcal{M}}^{\varphi,\widehat{\iota}}]_k$ is defined over $(k+1) \cdot f_k(\varphi)$ different symbolic states, $k \cdot f_k(\varphi)$ different symbolic actions, and $k \cdot f_k(\varphi)$ different symbolic weights. Moreover, it uses the following auxiliary quantifier-free first-order formulae:

- $I_s(\overline{\mathbf{w}})$ – it encodes the state s of the abstract model $\widehat{\mathcal{M}}$;
- $H_{\mathbf{c}}(w_{\mathbf{c}}, w'_{\mathbf{c}})$ – it encodes equality of two local states, such that $w_{\mathbf{c}} = w'_{\mathbf{c}}$ for $\mathbf{c} \in Ag \cup \mathcal{E}$;
- $\mathcal{T}_{\mathbf{c}}(w_{\mathbf{c}}, ((\overline{a}, \overline{d}), \overline{\delta}), w'_{\mathbf{c}})$ – it encodes the local evolution function of agent $\mathbf{c}$;
- $\mathcal{A}(\overline{a})$ – it encodes that each symbolic local action $a_{\mathbf{c}}$ of $\overline{a}$ has to be executed by each agent in which it appears;
- $\mathcal{T}(\overline{\mathbf{w}}, ((\overline{a}, \overline{d}), \overline{\delta}), \overline{\mathbf{w}}') := \mathcal{A}(\overline{a}) \wedge \bigwedge_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}} \mathcal{T}_{\mathbf{c}}(w_{\mathbf{c}}, ((\overline{a}, \overline{d}), \overline{\delta}), w'_{\mathbf{c}})$;
- Let $\boldsymbol{\pi}_j$ denote the j-th *symbolic k-path*, i.e. the sequence of symbolic transitions: $\overline{\mathbf{w}}_{0,j} \xrightarrow{(\overline{a}_{1,j}, \overline{d}_{1,j}), \delta_{1,j}} \overline{\mathbf{w}}_{1,j} \xrightarrow{(\overline{a}_{2,j}, \overline{d}_{2,j}), \delta_{2,j}} \ldots \xrightarrow{(\overline{a}_{k,j}, \overline{d}_{k,j}), \delta_{k,j}} \overline{\mathbf{w}}_{k,j}$. Then, $\mathcal{D}^I_{a,b;c,d}(\boldsymbol{\pi}_n)$ for $a \leq b$ and $c \leq d$ is a formula that:
  - for $a < b$ and $c < d$ encodes that the weight represented by the sequences $\overline{d}_{a+1,n}, \ldots, \overline{d}_{b,n}$ and $\overline{d}_{c+1,n}, \ldots, \overline{d}_{d,n}$ belongs to the interval I,
  - for $a = b$ and $c < d$ encodes that the weight represented by the sequence $\overline{d}_{c+1,n}, \ldots, \overline{d}_{d,n}$ belongs to the interval I,
  - for $a < b$ and $c = d$ encodes that the weight represented by the sequence $\overline{d}_{a+1,n}, \ldots, \overline{d}_{b,n}$ belongs to the interval I,
  - for $a = b$ and $c = d$, the formula $\mathcal{D}^I_{a,b;c,d}(\boldsymbol{\pi}_n)$ is true iff $0 \in I$.

Thus, given the above, one can define the formula $[\widehat{\mathcal{M}}^{\varphi,\widehat{\iota}}]_k$ as follows:

$$[\widehat{\mathcal{M}}^{\varphi,\widehat{\iota}}]_k := \bigvee_{s \in \widehat{\iota}} I_s(\overline{\mathbf{w}}_{0,0}) \wedge \bigvee_{j=1}^{f_k(\varphi)} \overline{\mathbf{w}}_{0,0} = \overline{\mathbf{w}}_{0,j} \wedge \bigwedge_{j=1}^{f_k(\varphi)} \bigwedge_{i=0}^{k-1} \mathcal{T}(\overline{\mathbf{w}}_{i,j}, ((\overline{a}_{i,j}, \overline{d}_{i,j}), \overline{\delta}_{i,j}), \overline{\mathbf{w}}_{i+1,j})$$

where $\overline{\mathbf{w}}_{i,j}$, $\overline{a}_{i,j}$, and $\overline{d}_{i,j}$ are, respectively, symbolic states, symbolic actions, and symbolic weights for $0 \leq i \leq k$ and $1 \leq j \leq f_k(\varphi)$. Hereafter, by $\boldsymbol{\pi}_j$ we denote the j-th symbolic k-path of the above unfolding, i.e., the sequence of transitions: $\overline{\mathbf{w}}_{0,j} \xrightarrow{(\overline{a}_{1,j}, \overline{d}_{1,j}), \overline{\delta}_{1,j}} \overline{\mathbf{w}}_{1,j} \xrightarrow{(\overline{a}_{2,j}, \overline{d}_{2,j}), \overline{\delta}_{2,j}} \ldots \xrightarrow{(\overline{a}_{k,j}, \overline{d}_{k,j}), \overline{\delta}_{k,j}} \overline{\mathbf{w}}_{k,j}$.

The formula $[\varphi]_{\widehat{\mathcal{M}},k}$ encodes the bounded semantics of a WECTLK formula φ, and it is defined on the same sets of individual variables as the formula $[\widehat{\mathcal{M}}^{\varphi,\widehat{\iota}}]_k$. Moreover, it uses the auxiliary quantifier-free first-order formulae defined in [7].

Furthermore, following [7], our formula $[\varphi]_{\widehat{\mathcal{M}},k}$ uses the following auxiliary functions g_l, g_r, g_μ, $h_{\mathbf{U}}$, $h_{\mathbf{G}}$ that were introduced in [10], and which allow to divide the set $A \subseteq F_k(\varphi) = \{j \in \mathbb{N} \mid 1 \leq j \leq f_k(\varphi)\}$ into subsets needed for translating the subformulae of φ. Let $0 \leq n \leq f_k(\varphi)$, $m \leqslant k$, and $n' = min(A)$. The rest of translation is defined in the same way as in [7].

- $[\mathbf{true}]_k^{[m,n,A]} := \mathbf{true}$, $[\mathbf{false}]_k^{[m,n,A]} := \mathbf{false}$,
- $[p]_k^{[m,n,A]} := p(\overline{\mathbf{w}}_{m,n})$, $[\neg p]_k^{[m,n,A]} := \neg p(\overline{\mathbf{w}}_{m,n})$,
- $[\alpha \wedge \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \wedge [\beta]_k^{[m,n,g_r(A,f_k(\beta))]}$,

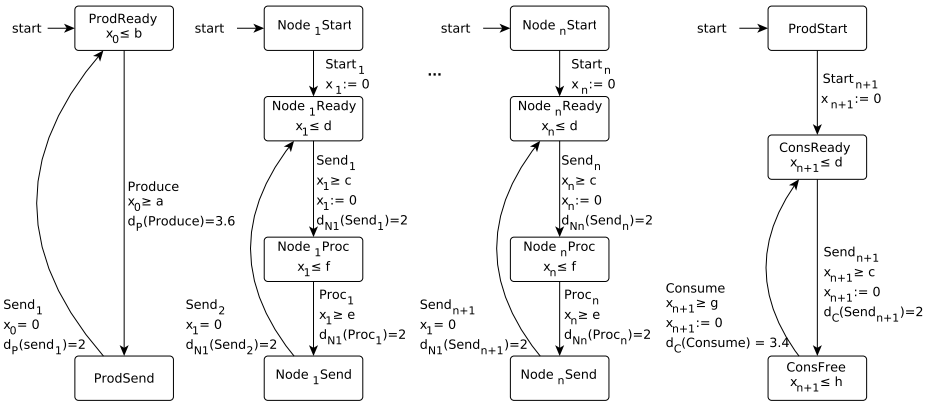


Figure 1 The TWGPP system.

- $[\alpha \vee \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \vee [\beta]_k^{[m,n,g_l(A,f_k(\beta))]}$,
- $[\mathbf{EX}_I\alpha]_k^{[m,n,A]} := \overline{\mathbf{w}}_{m,n} = \overline{\mathbf{w}}_{0,n'} \wedge (\overline{d}_{1,n'} \in I) \wedge [\alpha]_k^{[1,n',g_\mu(A)]}$, if $k > 0$; **false**, otherwise,
- $[\mathbf{E}(\alpha \mathbf{U}_I \beta)]_k^{[m,n,A]} := \overline{\mathbf{w}}_{m,n} = \overline{\mathbf{w}}_{0,n'} \wedge \bigvee_{i=0}^{k}([\beta]_k^{[i,n',h_{\mathbf{U}}(A,k,f_k(\beta))(j)]} \wedge (\sum_{j=1}^{i} \overline{d}_{j,n} \in I \wedge \bigwedge_{j=0}^{i-1}[\alpha]_k^{[j,n',h_{\mathbf{U}}(A,k,f_k(\beta))]})$,
- $[\mathbf{E}(\mathbf{G}_I\alpha)]_k^{[m,n,A]} := \overline{\mathbf{w}}_{m,n} = \overline{\mathbf{w}}_{0,n'} \wedge \big((\sum_{j=1}^{k} \overline{d}_{j,n} \geq \mathbf{right}(I) \wedge \bigwedge_{i=0}^{k}(\sum_{j=1}^{i} \overline{d}_{j,n} \notin I \vee [\alpha]_k^{[i,n',h_{\mathbf{G}}(A,k)(j)]})) \vee (\sum_{j=1}^{k} \overline{d}_{j,n} < \mathbf{right}(I) \wedge \bigwedge_{i=0}^{k}(\sum_{j=1}^{i} \overline{d}_{j,n} \notin I \vee [\alpha]_k^{[i,n',h_{\mathbf{G}}(A,k)(j)]}) \wedge \bigvee_{l=0}^{k-1}(\overline{\mathbf{w}}_{k,n'} = \overline{\mathbf{w}}_{l,n'} \wedge \bigwedge_{i=l}^{k-1}(\neg \mathcal{D}^I_{0,k;l,i+1}(\boldsymbol{\pi}_{n'}) \vee [\alpha]_k^{[i,n',h_{\mathbf{G}}(A,k)(j)]}))))\big)$,
- $[\overline{\mathrm{K}}_{\mathbf{c}}\alpha]_k^{[m,n,A]} := (\bigvee_{s \in \widehat{\iota}} I_s(\overline{\mathbf{w}}_{0,n'})) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,n',g_\mu(A)]} \wedge H_{\mathbf{c}}(\overline{\mathbf{w}}_{m,n}, \overline{\mathbf{w}}_{j,n'}))$,

The theorem below states the correctness and the completeness of the presented translation. It can be proved in a standard way, using induction on the complexity of the given WECTLK formula.

▶ **Theorem 2.** *Let $\widehat{\mathcal{M}}$ be an abstract model, and φ a WECTLK formula. For every $k \in \mathbb{N}$, $\widehat{\mathcal{M}} \models_k \varphi$ if, and only if, the quantifier-free first-order formula $[\widehat{\mathcal{M}}, \varphi]_k$ is satisfiable.*

4 Experimental Results

In this section, we experimentally evaluate the performance of our SMT-based BMC encoding for WECTLK over the TRWIS semantics.

The benchmark, we consider is the *timed weighted generic pipeline paradigm* (TWGPP) TRWIS abstract model [9]. The abstract model of TWGPP involves $n + 2$ agents: Producer producing data within the certain time interval ($[a, b]$) or being inactive, Consumer receiving data within the certain time interval ($[c, d]$) or being inactive within the certain time interval ($[g, h]$), a chain of n intermediate Nodes which can be ready for receiving data within the certain time interval ($[c, d]$), processing data within the certain time interval ($[e, f]$) or sending data. The weights are used to adjust the cost properties of Producer, Consumer, and of the intermediate Nodes.

Each agent of the scenario can be modelled by considering its local states, the local actions, the local protocol, the local evolution function, the local weight function, the local clocks, the clock constraints, the invariants, and the local valuation function. Fig. 1 shows

the local states, the possible actions, and the protocol, the clock constraints, invariants and weights for each agent. Null actions are omitted in the figure.

Given Fig. 1, the local evolution functions of TWGPP are straightforward to infer. Moreover, we assume the following set of propositional variables: $\mathcal{PV} = \{ProdReady, ProdSend, ConsReady, ConsFree\}$ with the following definitions of local valuation functions: $\widehat{\mathcal{V}}_P(ProdReady\text{-}0) = \{ProdReady\}$, $\widehat{\mathcal{V}}_P(ProdSend\text{-}1) = \{ProdSend\}$; $\widehat{\mathcal{V}}_C(ConsReady\text{-}0) = \{ConsReady\}$,
$\widehat{\mathcal{V}}_C(ConsFree\text{-}1) = \{ConsFree\}$.

Let $Act = Act_P \times \prod_{i=1}^{n} Act_{N_i} \times Act_C$, with $Act_P = \{Produce, Send_1\}$, $Act_C = \{Start_{n+1}, Consume, Send_{n+1}\}$, and $Act_{N_i} = \{Start_i, Send_i, Send_{i+1}, Proc_i\}$ defines the set of joint actions for the scenario. For $\widetilde{a} \in Act$ let $act_P(\widetilde{a})$ denotes an action of Producer, $act_C(\widetilde{a})$ denotes an action of Consumer, and $act_{N_i}(\widetilde{a})$ denotes an action of Node i. We assume the following local evolution functions: $t_P(ProdReady, x_0 \geq a, \emptyset, \widetilde{a}) = ProdSend$, if $act_P(\widetilde{a}) = Produce$; $t_P(ProdSend, true, \{x_0\}, \widetilde{a}) = ProdReady$, if $act_P(\widetilde{a}) = Send_1$ and $act_{N_i}(\widetilde{a}) = Send_1$; $t_C(ConsStart, true, \{x_{n+1}\}, \widetilde{a}) = ConsReady$, if $act_C(\widetilde{a}) = Start_{n+1}$; $t_C(ConsReady, x_{n+1} \geq c, \{x_{n+1}\}, \widetilde{a}) = ConsFree$, if $act_C(\widetilde{a}) = Send_{n+1}$ and $act_{N_n}(\widetilde{a}) = Send_{n+1}$; $t_C(ConsFree, x_{n+1} \geq g, \{x_{n+1}\}, \widetilde{a}) = ConsReady$, if $act_C(\widetilde{a}) = Consume$.

Finally, we assume the following two local weight functions for each agent:

- $d_P(Produce) = 4$, $d_P(send_1) = 2$, $d_C(Consume) = 4$, $d_C(send_{n+1}) = 2$, $d_{N_i}(send_i) = d_{Ni}(send_{i+1}) = d_{N_i}(Proc_i) = 2$,
- $d_P(Produce) = 4000$, $d_P(send_1) = 2000$, $d_C(Consume) = 4000$, $d_C(send_{n+1}) = 2000$, $d_{N_i}(send_i) = d_{Ni}(send_{i+1}) = d_{N_i}(Proc_i) = 2000$.

The set of all the global states $\widehat{S}$ for the scenario is defined as the product $(L_P \times \mathbb{N}) \times \prod_{i=1}^{n}(L_i \times \mathbb{N}) \times (L_C \times \mathbb{N})$. The set of the initial states is defined as $\widehat{\iota} = \{s^0\}$, where $s^0 = ((ProdReady\text{-}0, 0), (Node_1Ready\text{-}0, 0), \ldots, (Node_nReady\text{-}0, 0), (ConsReady\text{-}0, 0))$.

The system is scaled according to the number of its Nodes (agents), i.e., the problem parameter n is the number of Nodes. For any natural number $n \geq 0$, let $D(n) = \{1, 3, \ldots, n-1, n+1\}$ for an even n, and $D(n) = \{2, 4, \ldots, n-1, n+1\}$ for an odd n. Moreover, let $r(j) = d_P(Produce) + 2 \cdot \sum_{i=1}^{j} d_{N_i}(Send_i) + \sum_{i=1}^{j-1} \cdot d_{N_i}(proc_i)$. Then, we define $Right$ as follows: $Right = \sum_{j \in D(n)} r(j)$.

We consider the following formulae as specifications:

- $\varphi_1 = \overline{\mathrm{K}}_P(\mathbf{EF}_{[0,Right)}(ConsFree \wedge \mathbf{EG}(ProdSend \vee ConsFree)))$ - *it states that it is not true that Producer knows that there exists a path on which Consumer receives a data and the cost of receiving the data is less than Right and from that point there exists a path on which always either the Producer has sent a data or the Consumer has received a data.*
- $\varphi_2 = \overline{\mathrm{K}}_P(\mathbf{EF}_{[0,Right)}(ConsFree \wedge \overline{\mathrm{K}}_C\overline{\mathrm{K}}_P(\mathbf{EG}(ProdSend \vee ConsFree))))$- *it states that it is not true that Producer knows that there exists a path on which Consumer receives a data and the cost of receiving the data is less than Right and at that point it is not true that Consumer knows that it is not true that Producer knows that there exists a path on which always either the Producer has sent a data or Consumer has received a data.*

The number of the considered k-paths is equal to 3 for φ_1, and 5 for φ_2, respectively. The length of the witness is $2 \cdot n + 4$ if $n \in \{1, 2\}$ and, $2 \cdot n + 2$ if $n > 2$ for the formula φ_1, $2 \cdot n + 2$ for the formula φ_2, respectively.

Performance evaluation. We performed our experimental results on a computer equipped with I7-3770 processor, 32 GB of RAM, and the operating system Arch Linux with the kernel 3.19.2. The CPU time limit was set to 3600 seconds. Our SMT-based BMC algorithm was

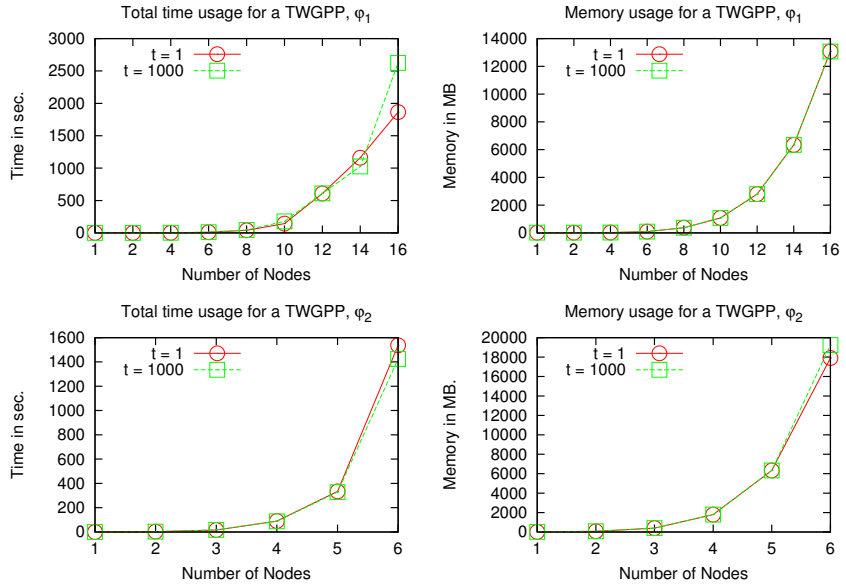


Figure 2 Formulae φ_1 and φ_2: Scaling up both the number of nodes and weights.

implemented as a standalone program written in the programming language `C++`. We used the state of the art SMT-solver Z3 [5].

For both properties φ_1 and φ_2 we scaled up both the number of nodes and the weights parameters. The results are summarised on charts in Fig. 2. One can observe that our SMT-based BMC is not sensitive to scaling up the weights, but it is sensitive to scaling up the size of benchmark.

For both the formulae, we obtained encouraging results. Namely, for φ_1 and for TWGPP with 16 nodes and the basic weights (`bw` for short) our method uses 13074.2 MB and 1864.4 seconds (13072.0 MB and 2624.5 seconds for `bw` multiplied by 1000); Next, for φ_2 and TWGPP with 6 nodes our method uses 17904.5 MB and 1536.9 second (19240.9 MB and 1424.4 seconds for `bw` multiplied by 1000).

5 Conclusions

We have proposed SMT-based BMC verification method for model checking WECTLK properties interpreted over the timed real-weighted interpreted systems. The preliminary experimental results show that the method is worth interest. In the future we are going to provide a comparison of our new method with the SAT- and BDD-based BMC methods. The module will be added to the model checker VerICS ([3]). All the benchmarks together with an instruction how to reproduce our results can be found at the webpage `http://www.ajd.czest.pl/~imi/agnieszkazbrzezny/modelchecking/`.

References

1 R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge, 1995.

2 P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of CAV'04*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.

3 M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Woźna, and A. Zbrzezny. VerICS 2007 – a model checker for knowledge and real-time. *Fundamenta Informaticae*, 85(1-4):313–328, 2008.

4 A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. In *Proceedings of CAV'09*, volume 5643 of *LNCS*, pages 682–688. Springer-Verlag, 2009.

5 L. De Moura and N. Bjørner. Z3: an efficient SMT solver. In *Proceedings of TACAS'08*, volume 4963 of *LNCS*, pages 337–340. Springer-Verlag, 2008.

6 M. Wooldridge. *An introduction to multi-agent systems – Second Edition.* John Wiley & Sons, 2009.

7 B. Woźna-Szcześniak. SAT-based bounded model checking for weighted deontic interpreted systems. In *Proceedings of EPIA'13*, volume 8154 of *LNAI*, pages 444–455. Springer-Verlag, 2013.

8 B. Woźna-Szcześniak. Checking EMTLK properties of timed interpreted systems via bounded model checking. In *Proceedings of AAMAS'14*, pages 1477–1478. IFAAMAS/ACM, 2014.

9 B. Woźna-Szcześniak, A. M. Zbrzezny, and A. Zbrzezny. SAT-based bounded model checking for weighted interpreted systems and weighted linear temporal logic. In *Proceedings of PRIMA'13*, volume 8291 of *LNAI*, pages 355–371. Springer-Verlag, 2013.

10 A. Zbrzezny. Improving the translation from ECTL to SAT. *Fundamenta Informaticae*, 85(1-4):513–531, 2008.

Automatic Transformation of Raw Clinical Data Into Clean Data Using Decision Tree Learning Combining with String Similarity Algorithm*

Jian Zhang

School of Computing, University of Dundee
Perth Road, Dundee, UK
j.s.zhang@dundee.ac.uk

Abstract

It is challenging to conduct statistical analyses of complex scientific datasets. It is a time-consuming process to find the relationships within data for whether a scientist or a statistician. The process involves preprocessing the raw data, the selection of appropriate statistics, performing analysis and providing correct interpretations, among which, the data pre-processing is tedious and a particular time drain. In a large amount of data provided for analysis, there is not a standard for recording the information, and some errors either of spelling, typing or transmission. Thus, there will be many expressions for the same meaning in the data, but it will be impossible for analysis system to automatically deal with these inaccuracies. What is needed is an automatic method for transforming the raw clinical data into data which it is possible to process automatically.In this paper we propose a method combining decision tree learning with the string similarity algorithm,which is fast and accuracy to clinical data cleaning. Experimental results show that it outperforms individual string similarity algorithms and traditional data cleaning process.

1998 ACM Subject Classification H.3.3 Information Search and Retrieval, I.2.6 Learning

Keywords and phrases Raw Clinical Data, Decision Tree Learning, String Similarity Algorithm

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.87

1 Introduction

Cancer research has become a greatly data rich environment. Several data analysis packages can be used for analyzing the data like SPSS (Statistical Product and Service Solutions) [1], Minitab [9]. However, before analyzing the data, it needs to be preprocessed to fix the errors, misspelling of the raw data and transform the raw data into an uniform data [15], making it fit-for-purpose. This process is both time-consuming and tedious. For example, a specific binary variable may require only the entries 'Yes' or 'No' in a large data column. However, 'Yes' may have been entered onto a spreadsheet as Yes, yes, Y, y, yES, 1, or been misspelled as Yed, yef, y es, y e s (note the inappropriate use of spacing), etc.. Clearly, there are an infinite number of possibilities of entering this 3-letter word incorrectly, and each of these entries is treated as separate entities by a computer program.

Obviously, we can see that there are only three possible answers in this data like the first column of the Table 1 – Yes, No or null. However, the errors include spelling, typing or transmission [2]. Besides, there is not a standard for scientists to collect data and record

* This work was partially supported by Henry Lester Trust.

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 87–94

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 The example of raw data.

Clean data	Raw data
yes	yes, Yed, yef, Y...
no	No, N, not...
null	don't know, waiting for lab...

them. Thus, the collected data will probably be the raw data which includes some errors like that in the second column of the Table 1. It is difficult for the system to directly deal with these inaccuracies for statistical analysis.

Currently statisticians will manually change these entries into a uniform string or a number allowing the system to do further analysis. Potter's Wheel [16], Google Refine [17, 5] are interactive data cleaning system, which can be used to clean the data up and transform the data from one form into another. Therefore, it will save time for statisticians if an automated method can be used instead of a manual operation [4]. During the transformation, the statisticians usually amend these data using their previous experience.

Decision tree learning is a common algorithm of machine learning [10]. It will set some rules via previous experience to build a tree, and then predict the result using the built tree. Its process is similar to the operation of transformation by statisticians mentioned above. According to this feature and the necessity of data classification, decision tree learning has been chosen to clean the raw data automatically.

Meanwhile,the comparison between the new entry and the previous data is another important method to transform the raw clinical data to clean data during the manual operation [6]. Thus, it is worth attempting the string similarity algorithm for this research.

In the raw clinical data, there is ordinal data and continuous data. The ordinal data hold larger percentage in the research data so this paper will explore the research in the ordinal data, especially two-category data.

In this paper, decision tree learning algorithms and string similarity algorithm will be separately investigated to transform the raw clinical data to clean data and their features will be explored. After that, the result of the two methods will be analyzed. Finally, the two method will be combined as their features for the raw clinical data.

2 Decision Tree Learning

In general,training, validation and comparison with training data set will be the process of making a prediction in machine learning. Decision tree learning as one of the machine learning algorithm is a method for approximating discrete-valued functions [10], which is one of the most popular inductive algorithms. Decision tree learning is a method to classify different values under different attributes. It can set the rules based on the past data to classify the new raw data to the clean data, whose process is the same as the manual data pre-processing.

2.1 Decision Tree Representation

Decision trees classify data through arranging the case from the tree root to a leaf. Every internal node (not leaf node) on the tree stands for a test of the value of the attribute; the branch represents the result of the test; every leaf represents a classification categories. In short, decision trees are similar to the tree structure of the workflow, adopting the top-down

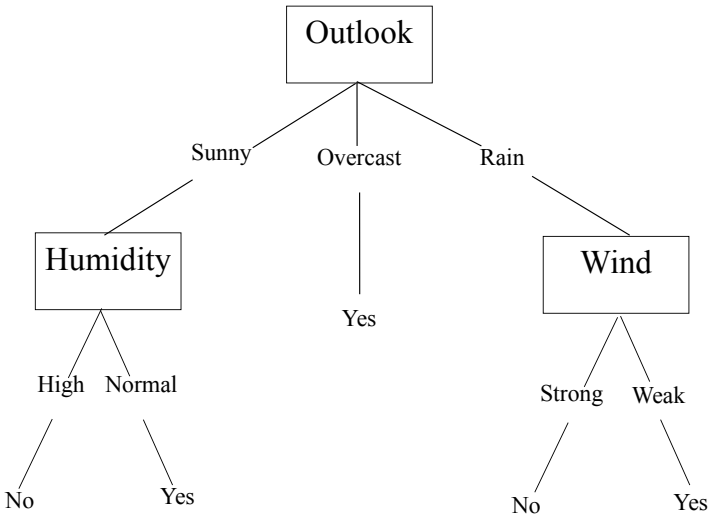


Figure 1 A simple decision tree example from [13].

inductive method. It begins from the root node, and then tests the value of the attribute in the internal nodes, next, confirms the matched branch based on the value of the attribute, finally, gets the result in the leaf node [10].

Figure 1 represents a typical learned decision tree. It describes the relationships between the weather and playing tennis [10]. In the figure, Yes equals play tennis, and No equals not to play tennis. There are three attributes, outlook, humidity and windy. Each attribute has their own value. For example, {outlook=rain; windy=strong; result =No} is a branch of the tree.

Generally, decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances [10]. Each path from the root to a leaf corresponds to these disjunctions of conjunctions. In order to classify the raw clinical data, it can test the attribute values based on the structure of the decision tree, from the tree root down to the child node, and finally to the leaf node. And the corresponding category of the leaf node will be the category of the data object.

2.2 Decision Tree Learning Algorithm

ID3 was proposed by Quinlan in 1986 [13]. It is a representation of a decision tree algorithm and most of the decision tree algorithms are achieved based on improvements to it. It adopts the divide-and-conquer strategy and uses information gain as the standard to choose the attribute at the different levels of the decision tree in order that it can collect the most information of the categories about the test records in the process of testing each non-leaf node.

And its demerits are that it can only process the discrete attribute, and sometimes it is not the best to process the attribute with lots of values.

As a result of some problems with the ID3 algorithms in the practical application, Quinlan proposed the C4.5 algorithm [12], strictly speaking, which is just an improved algorithm of ID3.C4.5 algorithm inherits the advantages of the ID3 algorithm, and improves several aspects of the ID3 algorithm. C4.5:

- It uses information gain ratio to choose an attribute so that it overcomes the weakness

Table 2 Probability of transforming the test data to clean data using different decision tree algorithm.

Decision Tree	Unknown Entry (%)	Total (%)	Time (ms)
ID3	4	75	109
C4.5	27	81	109

that the system tends to choose the attribute with more values when using information gain.

- It adds prune in the process of constructing the tree.
- It can complete the discretization of the continuous attributes.
- It can process the incomplete data [14].

2.3 Experiments and Results

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, which contains a collection of algorithms for data analysis and predictive modeling [18]. The algorithm ID3 and C4.5 are also coded in this software, and the detailed results including the possibility of the prediction will be calculated as well, so it will be adopted in testing the data.

The raw clinical data in this research is about breast cancer research from the website [3]. As mentioned in the introduction section, the two-category data will be undertaken in this experiment. The value of data is similar as the data in the Table 1. It will be divided into two parts, and one part is as training data set and the other part is as testing data set. In the testing data set, the data can be found in the training dataset, which is called the existing data; the other part is called the unknown data, which takes up 26%.

Table 2 shows that the probability of the correct transformation using the algorithm ID3 and C4.5 for the existing data, the unknown data and total. All the existing entries have been cleaned in the process. Compared with the probability of the correct transformation for the existing data, the percentage for the unknown data is quite low, which is 4% and 27% of all the unknown data respectively for ID3 and C4.5. Total results for correct transformation is 75% and 81% respectively.

All in all, decision tree learning have a good performance for the existing data transformation, but have a low performance on the unknown data [19]. The efficiency of the transformation mainly depends on the percentage of the existing data in the whole testing data set.

3 String Similarity Algorithm

3.1 String Similarity Algorithms

In this research, it is necessary to calculate the string similarity of the two strings when the entry is transformed from the raw data to processable data by the system. String-based similarity has a long history. Levenshtein proposed the edit distance, which is widely used for string similarity through calculating the minimum number of insertions, substitutions and deletions between two strings [8].The Levenshtein distance will be calculated once when add, delete or substitute have been done once during transformation. It provides a numeric

Table 3 Probability of transforming the test data to clean data using different string similarity algorithms.

Decision Tree	Unknown Entry (%)	Total (%)	Time (ms)
C4.5	27	81	109
Levenshtein Distance	58	89	1045
Needleman-Wunsch	73	89	846
Jaro-Winkler distance	73	91	907
N-W + Len	73	93	900

approach for transformation. For example, the Levenshtein distance between 'Yef' and 'Yes' is 1, because only one edit should be done, substitution of 's' for 'k'.

Needleman and Wunsch [11] extended the model to allow contiguous sequences of mismatched characters, or gaps, in the alignment of two strings, and described a general dynamic programming method for computing edit distance. For example, the score of Needleman and Wunsch between 'Yef' and 'Yes' is 2, since there are two matching letters 'Ye' between them.

Jaro-Winkler distance find the approximating string matching by means of calculating the number of matching characters and the number of transpositions. And a prefix scale is added in this method as well [7]. For example, the score between 'Yef' and 'Yes' is 0.82, and the score between 'Yef' and 'Tef' is 0.77. Even though the two group of string both have two matching letters, the first group have the prefix matching. Hence, 'Yef' is more similar as 'Yes' rather than 'Tef' based on this algorithm.

The Needleman-Wunsch algorithm is used to calculate the longest common substring (LCS). We can consider the effect of the length (Len) of the string in the N-W algorithm to improve it. For example,the score between 'Yef' and 'Yes' is 0.67, and the score between 'Yef' and 'Yest' is 0.57. Even though the two group of string both have two matching letters, the first group have the shorter string. Hence, 'Yef' is more similar as 'Yes' rather than 'Yest' based on this algorithm.

3.2 Experiments and Results

In this experiment, the training data set and the testing data set in the previous section will be used again. Because the algorithm C4.5 gets a higher performance and is suitable for the continuous data, which can be used in the future research, the result of C4.5 will be added to the result of this experiment for analysis.

Table 3 represents that the same testing elements as previous experiment. For all the existing entries, Levenshtein distance, Jaro-Winkler distance and improved N-W have the 100% as the same as the C4.5 gets. The Needleman-Wunsch algorithm gets 97%. The string similarity algorithms get a better results for the unknown data rather than the algorithm C4.5. The result for Levenshtein distance transforming the unknown is more than 2 times as the algorithm C4.5. The results for rest algorithm are much higher, which is 73% of the unknown data. Because of the larger improvements for the unknown data, the string similarity algorithm have a lot improvements for transforming the raw data to clean over the C4.5 algorithm.

Overall, the string similarity have a higher performance for the unknown data transformation. However, it is found that the time for running the string similarity algorithm is much slower than the algorithm C4.5 during the transformation.

Table 4 Probability of correct prediction for different string similarity algorithms combined with decision tree learning.

Decision Tree	Unknown Entry (%)	Total (%)	Time (ms)
C4.5	27	81	109
C4.5 + Lev	58	89	342
C4.5 + N-W	65	91	292
C4.5 + J-W	73	93	308
C4.5 + N-W + Len	73	93	305

4 Decision Tree Learning Combined with String Similarity Algorithm

According to the two previous experiments, the algorithms C4.5 have a low performance for the unknown data transformation but have fast process whilst the string similarity algorithm has a higher performance for the unknown data but is much slower. Thus, the combination of the two algorithms is worth exploring based on their features and performance.

The string similarity algorithm Levenshtein distance (Lev), Needleman-Wunsch (N-W), Jaro-Winkler distance (J-W) and a improved string similarity algorithm based on Needleman-Wunsch (NW-Len) will be undertaken to process the incorrect prediction in the following experiment combined with the algorithm C4.5.

This experiment will be undertaken the same training data set and testing data set as the previous experiment.

Table 4 shows that the probability of correct prediction for different combining algorithms and includes that for the algorithm C4.5 for comparison. The second row of the table shows the results of decision tree prediction. The other rows represent the results of each string similarity algorithms combining with the decision.The result for transforming the unknown data is the same as them without the combination. The improved NW-Len combined with the algorithm C4.5 share the highest percentage (73%) with the J-W distance combined with the algorithm C4.5. And they both get the highest percentage (93%) of the total data transformation. The result for the N-W algorithm and the Levenshtein distance combined with the algorithm C4.5 is 91% and 89% respectively, which is higher than the result of the algorithm C4.5 (81%).

To sum up, the results for transforming the raw clinical data to the processable data have improved significantly after combining the decision tree learning algorithm with string similarity algorithms, especially with the Jaro-Winkler distance and the modified Needleman-Wunsch algorithm to compare with decision tree learning.And the combining algorithm get much faster than the individual string similarity algorithm.

5 Conclusion and Future Work

This paper attempts to find an approach to reduce the manual operation for tedious and repeated data transformation work. To sufficiently introduce the performance of the decision tree learning algorithm combined with the string similarity algorithm, this paper firstly introduces the decision tree learning algorithm and tests its performance for transformation from the raw clinical data to clean data. The results represent that it has high performance for the existing entry but low performance for the unknown data. The testing process is fast. Secondly, this paper undertakes the string similarity algorithm to test in the same way. The

results demonstrate that it has a higher performance for the unknown data. However, the process is slower than the decision tree algorithm. Finally, this paper tests the efficiency of the combining algorithm. The results show that it has the high performance for the existing data as the algorithm C4.5 and the same performance for the unknown data as the string similarity algorithms. And the process is slower than the algorithm C4.5 but quite faster than the string similarity algorithm.

Even though the results demonstrate that the decision tree algorithm combined with the string similarity algorithm can to some extent, automatically transforming the raw clinical data into the clean data in order to reduce the manual operation, there is still plenty of challenge in the further research. Firstly, the time-consuming may be a problem for a very large data set transformation when the training data set is not rich. This paper investigates the transformation for two-category data. Secondly, it can be extended to the more than two category data. Thirdly, once the raw data set has been transformed by the combining algorithm, the correct transformed the unknown entry can be considered to update the training data set, and then a new decision tree will be built so that maybe the next transformation process will be faster. What's more, the process of transformation is not completely automated in this paper. There is still a lot space for improving.

References

1 Alan C Acock. Sas, stata, spss: A comparison. alan c. acock. *Journal of Marriage and Family*, 67(4):1093–1095, 2005.

2 Cyril N Alberga. String similarity and misspellings. *Communications of the ACM*, 10(5):302–313, 1967.

3 Clinical. `https://www.clinicalstudydatarequest.com/`. Accessed: 25 June 2015.

4 Mita K Dalal and Mukesh A Zaveri. Automatic text classification: a technical review. *International Journal of Computer Applications*, 28(2):37–40, 2011.

5 Google. `https://code.google.com/p/google-refine/`. Accessed: 25 June 2015.

6 D Gussfield. Algorithms on strings, trees, and sequences. *Computer Science and Computional Biology (Cambrigde, 1999)*, 1997.

7 Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

8 Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966.

9 Ruth Meyer and David Krueger. *Minitab guide to statistics*. Prentice-Hall, Inc., 1997.

10 Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

11 Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

12 J Quinlan. R.(1993) c4. 5: Programs for machine learning, 1993.

13 J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

14 J. Ross Quinlan. Improved use of continuous attributes in c4. 5. *Journal of artificial intelligence research*, pages 77–90, 1996.

15 Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

16 Vijayshankar Raman and Joseph M Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB*, volume 1, pages 381–390, 2001.

17 Seth Van Hooland, Ruben Verborgh, Max De Wilde, Johannes Hercher, Erik Mannens, and Rik Van de Walle. Free your metadata: Integrating cultural heritage collections through google refine reconciliation. *Pre-submission paper available on. http://freeyourmetadata.org/publications/freeyourmetadata. pdf*, 2011.

18 Ian H Witten, Eibe Frank, Leonard E Trigg, Mark A Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.

19 Jian Zhang, Karen Petrie, and Tingting Yu. Automatic transformation of raw clinical data into clean data using decision tree learning. *LMT*, 84(91):0–2344.