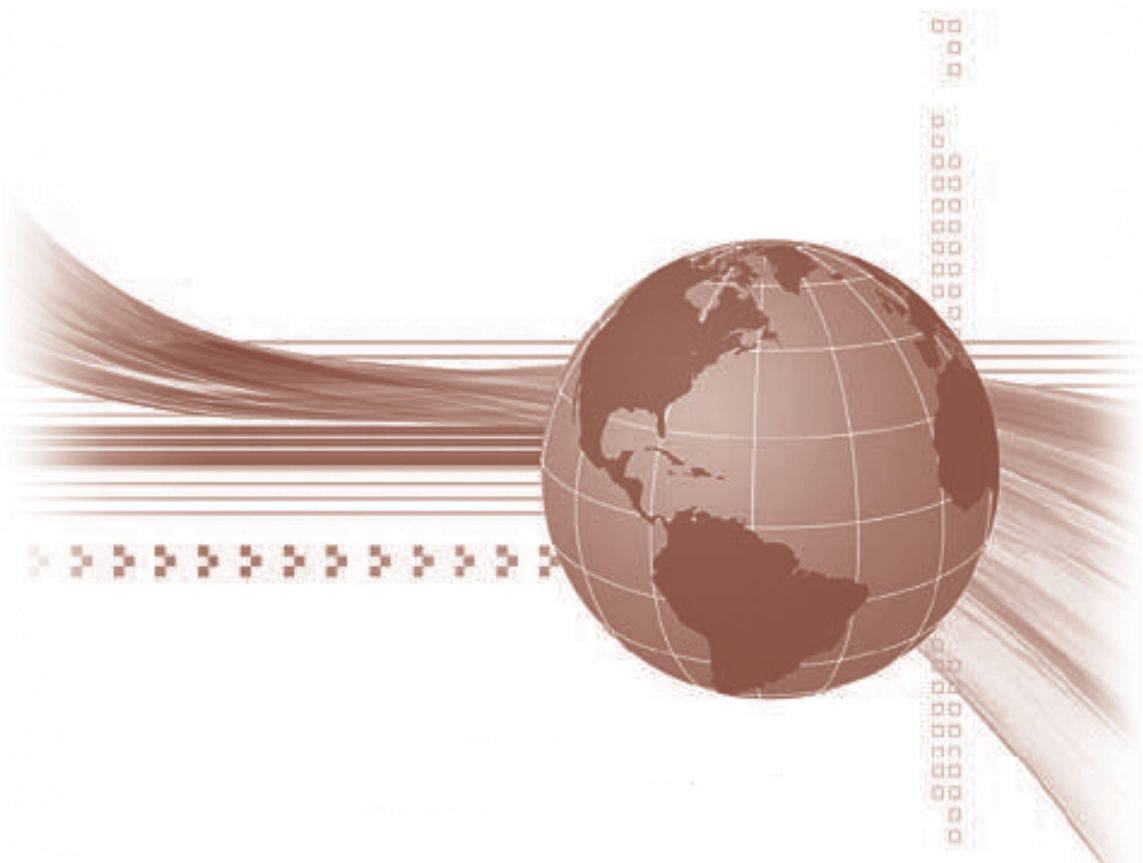
STUDIA UNIVERSITATIS
BABEŞ-BOLYAI



INFORMATICA

2/2013

STUDIA

UNIVERSITATIS BABEŞ-BOLYAI
INFORMATICA

No. 2/2013

April - June

This volume contains papers presented at the International Conference

KEPT2013

KNOWLEDGE ENGINEERING PRINCIPLES AND TECHNIQUES

The conference has been kindly sponsorted by

EDITORIAL BOARD

YEAR **Volume 58 (LVIII) 2013**
MONTH **JUNE**
ISSUE **2**

S T U D I A
UNIVERSITATIS BABEŞ-BOLYAI
INFORMATICA

2

EDITORIAL OFFICE: M. Kogălniceanu 1 • 400084 Cluj-Napoca • Tel: 0264.405300

SUMAR – CONTENTS – SOMMAIRE

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

KEPT2013: THE FOURTH INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING, PRINCIPLES AND TECHNIQUES

MILITON FRENŢIU, HORIA F. POP, AND SIMONA MOTOGNA

1. Introduction

The Faculty of Mathematics and Computer Science of the Babeş-Bolyai University in Cluj-Napoca is organizing the Fourth International Conference on Knowledge Engineering Principles and Techniques (KEPT2013), during July 5–7, 2013. This conference, organized on the platform of Knowledge Engineering, is a forum for intellectual, academic, scientific and industrial debate to promote research and knowledge in this key area, and to facilitate interdisciplinary and multidisciplinary approaches, more and more necessary and useful today. Knowledge engineering refers to the building, maintaining, and development of knowledge-based systems. It has a great deal in common with software engineering, and is related to many computer science domains such as artificial intelligence, databases, data mining, expert systems, decision support systems and geographic information systems. Knowledge engineering is also related to mathematical logic, as well as strongly involved in cognitive science and socio-cognitive engineering where the knowledge is produced by socio-cognitive aggregates (mainly humans) and is structured according to our understanding of how human reasoning and logic works. Since the mid-1980s, knowledge engineers have developed a number of principles, methods and tools that have considerably improved the process of knowledge acquisition and ordering. Some of the key issues include: there are different types of knowledge, and the right approach and technique should be used for the knowledge under study; there are different types of experts and ex- pertise, and methods should be chosen appropriately; there are different ways of representing knowledge, which can aid knowledge acquisition, validation and re-use; there are different ways of using knowledge, and the acquisition process can be goal-oriented; there are structured methods to increase the acquisition eficiency.

2. The content of KEPT2013

The Submissions were grouped into four traditional tracks in order to simplify the review process and Conference presentations. These sections are described downwards.

2.1. **Knowledge in Computational Linguistics (KCL).** The huge quantity of unstructured text documents stored on the web represents issues of the very hot researches in Computational Linguistics (or Natural Language Processing, NLP). As a part of Knowledge Engineering, Knowledge in Computational Linguistics includes the studies in Linguistic tools in Information retrieval and Information Extraction, in Text mining, Text entailment and Text summarization. The study of Discourse and Dialogue, of Machine learning for natural languages and of Linguistic components of information systems are also some very active fields in the present research. All these aspects of theoretical and application-oriented subjects related to NLP are subjects of debates in our section of Knowledge in Computational Linguistics.

2.2. **Knowledge Processing and Discovery (KPD).** The purpose of this track is to promote research in AI and scientific exchange among AI researchers, practitioners, scientists, and engineers in related disciplines. Topics include but are not limited to the following: Agent-based and multiagent systems; Cognitive modeling and human interaction; Commonsense reasoning; Computer vision; Computational Game Theory; Constraint satisfaction, search, and optimization; Game playing and interactive entertainment; Information retrieval, integration, and extraction; Knowledge acquisition and ontologies; Knowledge representation and reasoning; Learning models; Machine learning and data mining; Modelbased systems; Multidisciplinary AI; Natural computing: evolutionary computing, neural computing, DNA and membrane computing, etc.; Natural language processing; Planning and scheduling; Probabilistic reasoning; Robotics; Web and information systems.

2.3. **Knowledge in Software Engineering (KSE).** The main theme of this track is the interplay between software engineering and knowledge engineering, answering questions like: how knowledge engineering methods can be applied to software, knowledge-based systems, software and knowledge-ware maintenance and evolution, applications of knowledge engineering in various domains of interest.

2.4. **Knowledge in Distributed Computing (KDC).** For distributed computing and distributed systems, topics of interest include, but are not limited to, the following: System Architectures for Parallel Computing (including:

Cluster Computing, Grid and Cloud Computing); Distributed Computing (including: Cooperative and Collaborative Computing, Peer-to-peer Computing, Mobile and Ubiquitous Computing, Web Services and Internet Computing); Distributed Systems (including Distributed Systems Methodology and Networking, Software Agents and Multi-agent Systems, Distributed Software Components); Development of Basic Support Components (including Operating Systems for Distributed Systems, Middleware, Algorithms, Models and Formal Verification); Security in Parallel and Distributed Systems.

3. Invited lectures and accepted papers of KEPT2013

This fourth KEPT conference is honored by leading class keynote speakers, to present their invited lectures in two plenary sessions. This year, the lectures are presented by: Prof. Diana Inkpen (University of Ottawa, Canada), with a lecture on "Text Representation and General Topic Annotation based on Latent Dirichlet Allocation"; Prof. Prof. Attila Adamkó (University of Debrecen, Hungary), with a lecture on "Different approaches to MDWE: bridging the gap"; Prof. Zoltán Horváth (Eötvös Loránd University, Budapest, Hungary), with a lecture on "Workflow Description in Cyber-Physical Systems". The organisation of this conference reflects the following major areas of concern: Natural Language Processing, Knowledge Processing and Discovery, Software Engineering, and Knowledge in Distributed Computing. The 18 accepted papers (from 29 submitted) were organized in these four sections (2 to NLP, 8 to KPD, 3 to SE, and 5 to KDC). The participants submitted their works as peer-reviewed papers of 10–12 pages each. These full papers are published in this and the next issues, 2/2013 and 3/2013, of Studia Universitatis Babeş-Bolyai, Informatica journal.

4. Satellite workshops

Associated to the fourth KEPT conference, we organized three satellite workshops. Two of these workshops were organized in colaboration with partner companies, offering the opportunity to exchange ideas between academia and industry. The workshop on "Mobile development", organized in cooperation with the company Skobbler, took place on Friday, July 5, and the workshop on "Testing methodologies", organized in cooperation with the company Endava, took place on Saturday, July 6. As well, for the first time, we organized a satellite doctoral workshop, as an excellent opportunity for doctoral students to share their progress on doctoral work, exchange ideas, benefit from expert feed-back and defend their research reports.

5. Conclusions

We hope the Fourth International Conference on Knowledge Engineering Principles and Techniques (KEPT 2013) to be an exciting and useful experience and exchange of knowledge for our department. The possibility to communicate our most recent studies, and to compare with the results of other colleagues, the emulation of new ideas and research, all these mean a great gain of experience in our professional life. We hope that the next edition of KEPT (in 2015) will be even more successful and more enthusiastic than this one. We are taking the feedback of this Conference to improve the next editions, to atract more participants and to involve more personalities in the reviewing process.

Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: mfrentiu@cs.ubbcluj.ro

E-mail address: hfpop@cs.ubbcluj.ro

E-mail address: motogna@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

DIFFERENT APPROACHES TO MDWE: BRIDGING THE GAP

ATTILA ADAMKÓ AND LAJOS KOLLÁR

Abstract. This paper will give a short overview and comparison of Model-driven Web Engineering (MDWE) practices applied in both academy and industry with the aim of bridging the gap between those approaches. While Domain Specific Languages (DSLs) are used to express a high level outline of the imagined systems, the industrial approaches are focusing on a lower level and the distance of the two fields are mostly too wide to apply both in one development process. The goal of this paper is to propose a way to merge the two sides. DSLs can help to build prototypes in a very rapid manner. Based on those prototypes, common models can be derived that can serve as a basis for model-driven generation of Web applications using well-known production frameworks.

1. Introduction

Model-driven engineering has become more than a promising way of creating applications that are based on abstractions: MDE brings software development much closer to domain experts. It is the way how the MDA and MDD field could find its way to real life scenarios. However, these prominent ideas without fully realized key technology features cannot lead the way. Without executable modelling, meta-modelling, language engineering and proper tool support they are no more than interesting and idealistic research directions.

In the beginning of the 21st century there were several promising projects to support MDD and MDE. Nowadays, after a decade of the born of MDA and supporting technologies the list of possible tools with the mentioned key features are much more limited, only a few of them remain alive. We can

Received by the editors: June 1, 2013.

2010 *Mathematics Subject Classification.* 68U35, 68M11, 68N99.

1998 *CR Categories and Descriptors.* D.2.2 [**Software**]: Software Engineering – *Design Tools and Techniques*; D.2.10 [**Software**]: Software Engineering – *Design*.

Key words and phrases. MDWE, Web Applications, Model-driven development, Spring, Domain-Specific Languages.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

observe a landscape shift from general purpose languages to domain-specific ones. It opens the way for bridging the gap between stakeholders, domain experts and software developers.

2. Domain-specific languages and modeling

In software engineering, a domain-specific language (DSL) is a *"a computer programming language of limited expressiveness focused on a particular domain"* [6]. A DSL can be either a visual language, like the languages used by the Eclipse Modeling Framework, or textual languages.

A sound language description contains an abstract syntax, one or more concrete syntax descriptions, mappings between abstract and concrete syntaxes, and a description of the semantics. The abstract syntax of a language is often defined using a metamodel. The semantics can also be defined using a metamodel, but in most cases in practice the semantics are not explicitly defined but they have to be derived from the runtime behavior.

In model-driven engineering, many examples of domain-specific languages may be found, like OCL, a language for decorating models with assertions and constraints, or QVT, a domain-specific model transformation language. However, languages like UML are typically general purpose modeling languages.

Domain-specific languages have important design goals that contrast with those of general-purpose languages:

- domain-specific languages are less comprehensive;
- domain-specific languages are much more expressive in their domain;
- domain-specific languages should exhibit minimum redundancy.

3. Academical practices to MDWE

Modeling and systematic design methods are important and emerging fields in the Academic sector. Several new methodologies had born from this direction and resulted better software development practices. However, application of those methodologies are very time-consuming therefore it is mostly unacceptable for the industry. Several years are required before an approach becomes a well-functioning method that is ready for industrial use. The following sections describe some of the most interesting model-driven languages and methodologies (without attempting to be comprehensive). (The first one is an odd one out because it has both academic and industrial aspects.)

3.1. **WebML, WebRatio.** WebML [4] is a visual notation for specifying the content, composition, and navigation features of hypertext applications, building on ER and UML. Its not only for visualizing the models rather than a

methodology for designing complex data-intensive Web applications. It provides graphical, yet formal, specifications, embodied in a complete design process. Why it is an odd one out in this list that it has a commertial tool which can assist by visual design.

WebML enables designers to express the core features of a site at a high level, without committing to detailed architectural details. WebML concepts are associated with an intuitive graphic representation, which can be easily supported by CASE tools and effectively communicated to the non-technical members of the site development team. The specification of a site in WebML consists of four orthogonal perspectives:

- Structural Model
- Hypertext Model
  - Composition Model
  - Navigation Model
- Presentation Model
- Personalization Model

At first sight the business process modell is missing from the WebML methodology but its lying inside the composition model and navigation model, jointly named as hypertext model. Native BPMN is not supported by the tool, however there is a transformator which can transform a BPMN diagram into the WebML domain enriching the Application model.

Furthermore, WebML has been extended to cover a wider spectrum of front-end interfaces, thus resulting in the Interaction Flow Modeling Language (IFML), adopted as a standard by the Object Management Group (OMG) in 2013.

The WebML language started as a research project at Politecnico di Milano and later tool support has also been added. This software is called WebRatio that has become an industrial product so it can be considered as a success story for bridging the gap between academics and industry. In an ideal world, more ideas originating from the academic sector should reach that stage.

3.2. **UML-based Web Engineering.** UWE [8] applies the MDA pattern to the Web application domain from the from analysis to the generated implementation. Model transformations play an important role at every stage of the development process. The main reasons for using the extension mechanisms of the UML instead of a proprietary modelling technique are the acceptance of the UML in the field of software development and its extensibility with profiles. The UWE design approach for Web business processes consists of introducing specific process classes that are part of a separate process model with a defined interface to the navigation model.

Transformations at the platform independent level support the systematic development of models, like deriving a default presentation model from the navigation model. Then transformation rules that depend on a specific platform are used to translate the platform independent models describing the structural aspects of the Web application into models for the specific platform. Finally, these platform specific models are transformed to code by model-to-text transformations. Computer aided design using the UWE method is possible using MagicUWE, a plugin for MagicDraw.

3.3. **WebDSL.** WebDSL [11] is a domain-specific language for developing dynamic Web applications with a rich data model. The goal of WebDSL is to get rid of the boilerplate code you would have to write when building a Java application and raise the level of abstraction with simple, domain-specific sub-languages that allow a programmer to specify a certain aspect of the application and the WebDSL compiler would generate all the implementation code for that aspect. The main features of WebDSL are: Domain modeling, Presentation, Page-flow, Access control, Data validation, Workflow, Styling, Email.

Initially there were three sub-languages: a data modeling language, a user interface language and a simple action language to specify logic. WebDSL applications are translated to Java Web applications, and the code generator is implemented using Stratego/XT and SDF.

Although WebDSL is mainly a research project, a case study for domain-specific languages in the Web field [7], it can be usuable by anybody with some programming experience.

4. Industrial practices

OMG provides a key foundation for Model-Driven Architecture, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution. These concepts and directives have served as a basis for several solutions built by various companies. The only problem with these artifacts is the complexity. Complexity requires time and deep knowledge but in real life projects this factor is the most limited one. Companies are focusing on fast development time and choose frameworks supporting productivity rather than clear but time-consuming analysis phases. However, the following products could be used in an industrial environment because they have been proven to be working solutions while utilizing the OMG foundations.

4.1. **AndroMDA.** In short, AndroMDA is an open source MDA framework which works on UML models and utilizing plugins and components to generate

source code for a given programming language. Models are stored in XMI format produced from different CASE-tools. AndroMDA reads models into memory, making these object models available to its plugins. These plugins define exactly what AndroMDA will and will not generate. Each plugin is completely customizable to a project's specific needs.

It is mostly used by developers working with J2EE technologies and generate code for Hibernate, EJB, Spring and Web Services. AndroMDA allows customization of the templates used for code generation, therefore you can generate any additional code you want. We have seen AndroMDA in action and found it very promising in 2010 [1].

Currently, the only problem is its lost update cycle. The home page was last updated in 2011 and also the sourceforge repository's main branch seems to be stopped. However, there is a small sign of life because timestamps on several files in the SNAPSHOT branch show fresh (summer 2013) modification date. Because it was successfully applied on one of our industrial projects, it has the potential and possibility to became an alternative way for bridging the two world.

4.2. **openArchitectureWare (moved to Eclipse Modeling).** OpenArchitectureWare (oAW) was the second way for MDA around 2009. It was a modular MDA/MDD generator framework supporting arbitrary models and providing a language family to check and transform models as well as generate code based on them. OAW had strong support for EMF (Eclipse Modelling Framework) based models but could work with other models (e.g., UML2, XML or simple JavaBeans) too. The main power of oAW was the workflow engine which allowed to define generator/transformation workflows.

OAW was inherited by the Eclipse Modeling Framework and became a basis for it forming one big integrated family—Xpand (and Xtend), MWE (ant-like definition of transformations chains) and Xtext (textual DSL). It provides model-to-model and model-to-text transformation languages but they are not based on standards.

4.3. **Acceleo.** Acceleo is a pragmatic implementation of the Object Management Group (OMG) MOF Model to Text Language (MTL) standard. The creator was a French company, named Obeo. Nowadays, Acceleo is an Eclipse project mostly developed in Java and available under the Eclipse Public Licence (EPL) provided by the Eclipse Foundation. During the transition, the language used by Acceleo to define a code generator has been changed to use the new standard from the OMG for model-to-text transformation, MOFM2T. Acceleo is built on top of several key Eclipse technologies like EMF and, since the release of Acceleo 3, the Eclipse implementation of OCL (OMG's standard

language to navigate in models and to define constraints on the elements of a model). It has a very good tool support for productive coding.

4.4. **Spring.** One of the most powerful industrial solutions for Java EE application development is the Spring Framework. It is an open source application framework and Inversion of Control (IoC) container for the Java platform. The core features of the Spring Framework can be used by any Java applications, but there are extensions for building web applications on top of the Java EE platform. Although the Spring Framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBeans (EJB) model. The Spring Framework comprises several modules that provide a range of services including Inversion of Control container, Aspect-oriented programming, Data access, Transaction management, Model–View–Controller pattern, Remote access framework, Authentication and authorization, Messaging and Testing. These capabilities make it a strong candidate for industrial projects.

5. Closing the gap

A number of issues why the high majority of the industry is not committed to model-driven development has been identified in [5]. The authors' findings include that technical innovation does not go hand in hand with making profit; model-driven approaches are still believed to novel which means that they cannot be trusted enough for adoption; and MDD is considered to be heavyweight, complex and not mature enough to be used in a real-world development project. They also emphasize that *"simplicity is a key feature that helps to sell a technology"*, however, model-driven approaches can hardly be called simple.

Their additional observations are quite similar to those of [3] and [2]: successful application of model-driven techniques require a different approach to work (and a basic understanding and commitment) from project participants, revolutionary technologies require new forms of organization, tool support still does not reach the required level, etc. A very important observation states that *"the tools, training, and expectations of professionals under MDE are not as well developed and established as those under more traditional software development dynamics"* [2].

Despite of having some model-driven solutions that has become an industrial product (e.g., WebRatio), their range of application cannot be compared with those of the well-known and widely used frameworks like Spring.

Therefore, the existing gap between academical and industrial approaches can be filled, on the one hand, by developing better and better tools and teaching more and more people to think in models, or, on the other hand,

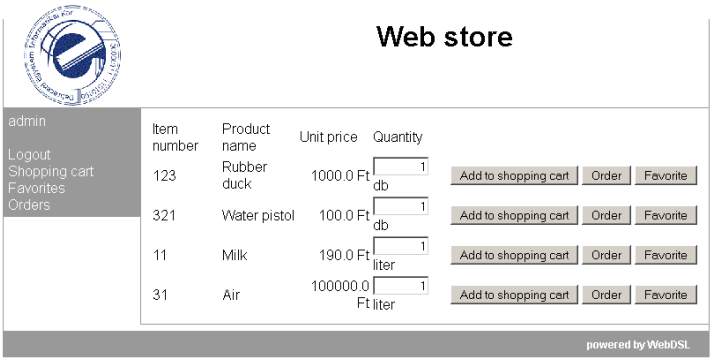


FIGURE 1. Web store implementation in WebDSL.

it might be filled by establishing model-based solutions which use the widespread production frameworks as a target platform. Our work is focused on that latter field: we propose a new model-driven generator that is able to generate source code and the necessary configuration files to the Spring framework having Hibernate as a persistence framework.

Since domain-specific languages (especially WebDSL) has been proved to be very effective in rapid prototyping, our starting point in development was to use it for easily and quickly define the concepts of the domain. This initial implementation might serve as a basis for building UML models that can later be transformed (using a generator developed for that purpose) to Spring. The reason for having UML models as intermediate artifacts is twofold: this way the development can be started by creating the appropriate UML models while it allows other DSL-to-UML mappings to be added later, extending the capabilities of the system this way.

5.1. **Demo application—Web store.** In order to demonstrate our ideas, a basic implementation for a Web store application has been created [10]. The project has been developed primarily for demonstrational purposes so it lacks much functionalities that a real web shop should implement.

This demo emphasizes the rapid prototyping capabilities of WebDSL: it is quite straightforward to create a base (but still useable) application in no more than 1,000 lines of code.

Figure 1 shows the developed application that has been transformed to Java code by the built-in generator included in the WebDSL Eclipse plugin.

5.1.1. *WebDSL implementation and problems.* The WebDSL implementation contains cca. 1,000 lines of code. It includes the definition of business entities (Product, User, Order, OrderItem, Cart, etc.), pages and navigation plus

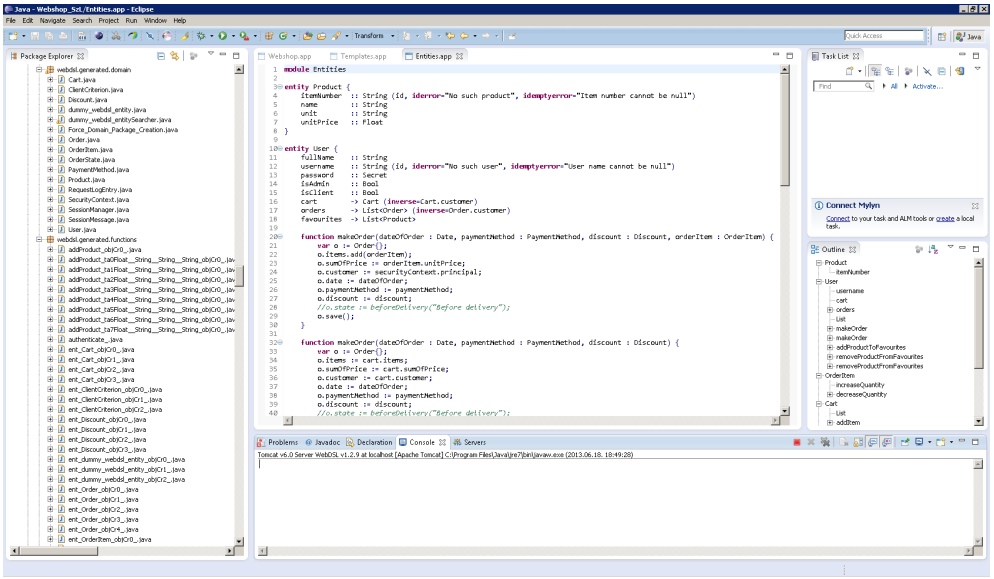


FIGURE 2. Excerpt of module Entities (middle) and the generated files (left).

some information on layout and access control. The WebDSL generator has generated 948 Java source files which is quite hard to understand and maintain. Figure 2 shows an excerpt of the module describing the business entities and Of course, a model-driven solution is not intended to modify the generated code (since you are required to modify the model and then re-generate), however, in practice, it is sometimes needed.

5.2. **WebDSL to UML transformation.** DSLs are backed with metamodels that capture the abstract syntax (i.e., the knowledge of the domain the DSL is aimed at). Based on that metamodel it is straightforward enough to transform the representation onto a model conforming the UML metamodel so due to lack of space we omit the details of the transformation.

5.3. **Generation of Spring implementation from UML model.** A generator that is able to generate Spring and Hibernate based implementation of a Web application from a stereotyped UML class diagram, has been developed [9]. The generator itself is built on top of Acceleo and is able to provide implementation for three layers: data layer, data access layer and business logic layer. For the data layer, it generates Hibernate entity classes, the data access layer will contain data access objects (DAOs), while the business logic layer contains POJIs (Plain Old Java Interfaces) for describing the services.

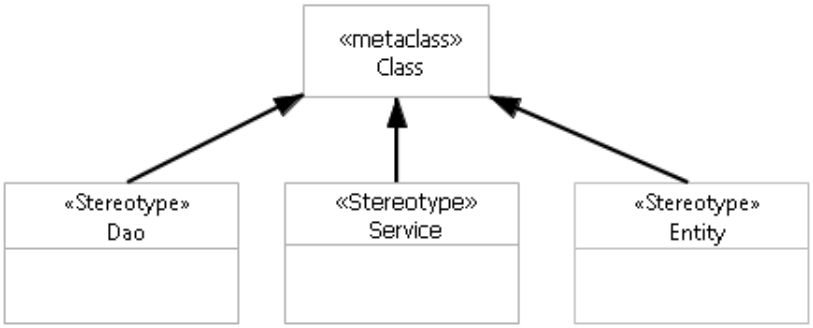


FIGURE 3. Stereotypes the generator understands.

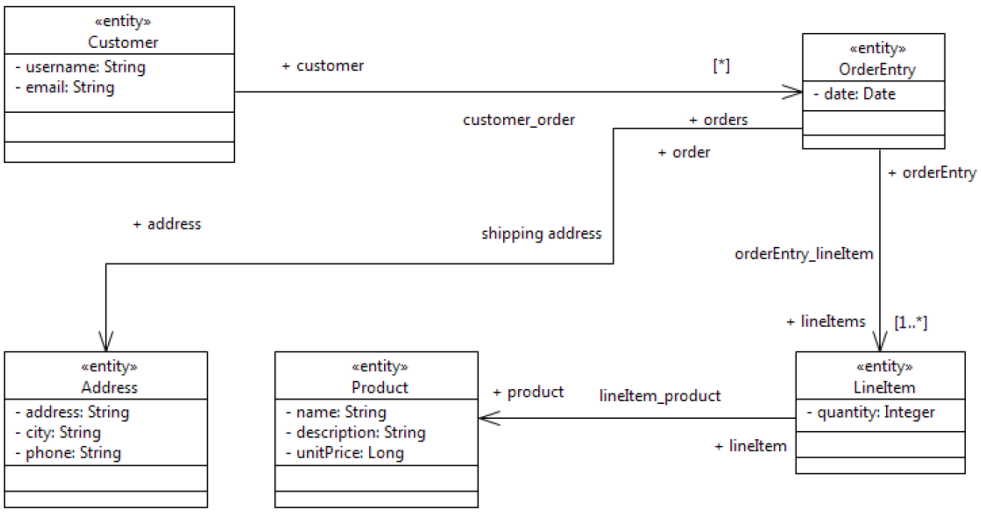


FIGURE 4. The domain model of the simplified Web store.

XML-based configuration files for both Spring and Hibernate are also generated.

The initial implementation of the generator deals only with class diagrams. Stereotypes are used to determine which layer the class belongs to. Figure 3 shows the stereotypes that are processed by the generator.

The domain model that has been created is shown in Figure 4. Its classes are stereotyped with <<entity>>.

Figure 5 demonstrates the Shopping Cart page of the generated Spring application.

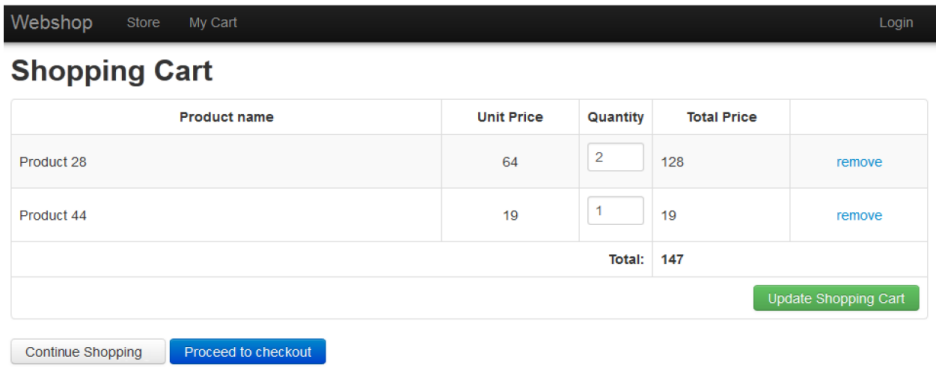


FIGURE 5. The Shopping Cart page of the generated Spring application.

6. CONCLUSION

For the MDE community, it is very important that new ideas and methods developed in the academia appear in industrial practice. Without it, model-driven practices will never reach their potential. This is the reason why we need increasing number of success stories like WebRatio. However, only a few of the proposed academic approaches receive broader attention from the industry.

In this paper, we gave a short (and, of course, highly incomplete) overview of the current state of model-driven engineering in academics and industry. In order to close the gap, we proposed a solution for generating applications for the Spring platform which is widely used across the industry. However, this work is not finished: by the time of this paper, our system is able to generate an application based on either a WebDSL descripiton or a UML model but generation of business logic is still an open question.

REFERENCES

[1] A. Adamkó and C. Bornemissza. Developing Web-Based Applications Using Model Driven Architecture and Domain Specific Languages. In *Proceedings of the 8th International Conference on Applied Informatics*, pages 287–293, 2010.

[2] J. Aranda, D. Damian, and A. Borici. Transition to model-driven engineering: what is revolutionary, what remains the same? In *Proceedings of the 15th international conference on Model Driven Engineering Languages and Systems*, MODELS'12, pages 692–708, Berlin, Heidelberg, 2012. Springer-Verlag.

[3] P. Baker, S. Loh, and F. Weil. Model-driven engineering in a large industrial context — motorola case study. In *Proceedings of the 8th international conference on Model Driven Engineering Languages and Systems*, MoDELS'05, pages 476–491, Berlin, Heidelberg, 2005. Springer-Verlag.

[4] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
[5] T. Clark and P.-A. Muller. Exploiting model driven technology: a tale of two startups. *Softw. Syst. Model.*, 11(4):481–493, Oct. 2012.
[6] M. Fowler. *Domain Specific Languages*. Addison-Wesley Professional, 1st edition, 2010.
[7] D. M. Groenewegen, Z. Hemel, and E. Visser. Separation of Concerns and Linguistic Integration in WebDSL. *IEEE Software*, 27(5):31–37, 2010.
[8] N. Koch and A. Kraus. Towards a common metamodel for the development of web applications. Cueva Lovelle, Juan Manuel (ed.) et al., Web engineering. International conference, ICWE 2003, Oviedo, Spain, July 14-18, 2003. Proceedings. Berlin: Springer. Lect. Notes Comput. Sci. 2722, 497-506 (2003)., 2003.
[9] A. Lőcsei. Webalkalmazások modell alapú készítése, 2013. Bachelor's thesis, University of Debrecen, Hungary, In Hungarian. Thesis supervisor: Lajos Kollár.
[10] L. Szarka. Modell-Vezérelt Webfejlesztési Megoldások. Master's thesis, University of Debrecen, Hungary, 2013. In Hungarian. Thesis supervisor: Attila Adamkó.
[11] E. Visser. WebDSL: A Case Study in Domain-Specific Language Engineering. In R. Lämmel, J. Visser, and J. a. Saraiva, editors, *Generative and Transformational Techniques in Software Engineering II*, pages 291–373. Springer-Verlag, Berlin, Heidelberg, 2008.

Department of Information Technology, Faculty of Informatics, University of Debrecen, H–4028 Debrecen, Kassai út 26., Hungary
E-mail address: adamkoa@inf.unideb.hu

Department of Information Technology, Faculty of Informatics, University of Debrecen, H–4028 Debrecen, Kassai út 26., Hungary
E-mail address: kollarl@inf.unideb.hu

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

WORKFLOW DESCRIPTION IN CYBER-PHYSICAL SYSTEMS

TAMÁS KOZSIK, ANDRÁS LŐRINCZ, DÁVID JUHÁSZ, LÁSZLÓ DOMOSZLAI, DÁNIEL HORPÁCSI, MELINDA TÓTH, AND ZOLTÁN HORVÁTH

Abstract. Cyber-physical systems (CPS) are networks of computational and physical processes, often containing human actors. In a CPS-setting, the computational processes collect information on their physical environment via sensors, and react upon via actuators in order to reach a desired state of the physical world.

In the approach presented in this paper a CPS application is implemented as a hierarchical workflow of mostly independent tasks, which are executed in a distributed environment, and satisfy timing constraints. In certain cases such workflows can be defined from natural language descriptions with the use of ontologies. The structure of a workflow, as well as the constraints put on the constituting tasks, are expressed in a domain-specific programming language.

1. Introduction

There is a growing need for complex controllable distributed systems. Some examples, selected randomly to illustrate the vast diversity of the application domains, are as follows: automated production lines, public transportation with driverless cars, infantry fighting vehicles, robotic surgery and internet-based multi-player augmented reality games. Cyber-physical systems (CPS) are networks of computational and physical processes, often containing

Received by the editors: June 1, 2013.

2010 *Mathematics Subject Classification.* 68N15, 68M14.

1998 *CR Categories and Descriptors.* D3.2 [**Programming Languages**]: Language Classifications – *Applicative (functional) languages, Concurrent, distributed, and parallel languages*.

Key words and phrases. cyber-physical system, task-oriented programming, workflow, timing constraint, domain specific language.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

human actors. In a CPS-setting, the computational processes collect information on their physical environment via sensors, and react upon via actuators in order to reach a desired state of the physical world.

There is a related emerging challenge for system and software development, which is concerned with the multi-faceted (physical, computational, psychological, cognitive), multi-layered (mobile, WLAN, backbone), multi-level (micro-to-macro), distributed computation and communication, from RF MEMS to cloud computing.

We suggest a novel route to address the challenge that we call *CPS Programming*, and which is concerned with the development of a domain specific language (DSL) for cyber-physical systems. The DSL is capable to describe distributed workflows, the timing constraints of the tasks involved, and fault-tolerance mechanisms.

In this paper, we propose a way for defining the computational parts of complex cyber-physical systems as workflows composed from hierarchical, independent building blocks. We have worked out a workflow system, where there are combinators specifically relevant to CPS Programming: time constraints, distribution, fault-tolerance and error-correction can be easily defined by means of them. We also describe the embedding of a domain-specific workflow language in Erlang.

The rest of the paper is structured as follows. Section 2 exposes the concepts for defining cyber-physical systems with workflows. Our workflow-system and its relevance to CPS Programming are revealed in Section 3. Finally, Section 4 concludes the paper.

2. The main concepts

The methodology proposed here splits up an application into modules, each implementing an (orchestrated set of) action(s) that can be executed, often in a reactive manner, in parallel with other actions. The main requirement is that interaction between modules should be minimized. There are two underlying reasons behind this constraint. Firstly, if interaction is complex, then time requirements of testing, as well as bandwidth requirements for real time error correction and reconfiguration will become unbearable. Secondly, unexpected instabilities might emerge because of the complexity and the accumulation of (small) errors in a controlled non-linear system.

We note that interaction with an ongoing process is slow, and the modification of a stable trajectory could be hard and may take time. Consider, for example, that in a given moment, say at time t, some deviation α from planned behavior is detected. If we were to choose a correction module that can correct α, we would ignore that the execution of the correction module

begins after a certain delay, the execution itself takes time, and the ongoing process may have side-effects on our correction module.

Hence, in order to meet our constraints, the progress of the application should be monitored, and whenever a failure, or some deviation from planned behavior is detected, a new correcting (sub)goal must be defined. In other words, we notice the deviation, predict its future dynamics, and make a plan to minimize the costs in the future on the top of the ongoing process. We have made the following assumptions: (i) we have a model, (ii) we can carry out model-based prediction, (iii) we can perform long-term cost optimization on the top of any ongoing process by means of modules, and (iv) these modules have minimal (or at least tolerable) side-effects on the ongoing process. This way we may (eventually) cope with the unavoidable delays apparent in a distributed and/or concurrent system.

2.1. **An illustrative example from nature.** Evolution teaches us for the relevance of the cost of interaction [1]. The brain has 10^{11} neurons, but only 10^{14} connections (instead of 10^{22}). Furthermore, the evolved system is built from robust (and complex) modules, but with minimized side-effects. Consider, for example, the mammalian control system [2], which has the following properties.

(1) Control space is divided according to high level tasks (eating, grasping, chewing, defense, manipulation in central space, climbing etc.), only a few may be concurrent at a time, but many (low complexity) combinations are executable in parallel.
(2) Each high level task is divided into sub-tasks; e.g., grasping is divided according to the discretization of the allo-centric 3D space within reach. It thus avoids combinatorial explosion of muscle space and is robust with respect to the huge dimension of body configuration space.

We conclude that module structure should be goal (task) driven and that the minimization of side-effects seems mandatory at least for the mammalian decision making and executive systems.

2.2. **Task and Test Driven Development.** A specific feature of CPS Programming is the software development methodology. The methodology must respect certain constraints, such as the use of a large number of units, stochastic behavior, delays in the execution of modules, system components originating from different sources, and humans-in-the-loop. Moreover, due to dependability requirements inherent in the CPS domain, the methodology should support both testing and verification. The main problem to solve here is the avoidance of combinatorial explosion both in the number of variables, and in the number of test cases.

The number of basic variables is typically large, and the full space scales with the number of variables in the exponent. Even evolution does not have the time to test all structures against one another in such a huge space. As opposed to evolution, our design serves certain tasks, so we are to test only those structures that have the promise of solving the task.

Note that tasks should be defined by decomposing goals, so they express a top-down approach. Testing, on the other hand, concerns the cases determined by the existing variables, so it is a bottom-up process. Unless we can limit the number of variables stepwise, we cannot test our solutions. This leads us to the well-known concept of side-effect free concurrent modules: if we test the individual modules at one level, then they may become our variables (in the sense that we may decide whether to include them) at the next level.

2.3. **Implementation aspect.** The above mentioned methodology is best supported by a domain specific language which is suitable to describe tasks, as well as task hierarchies and related timing constraints. Task-oriented programming [3] (TOP) provides the right paradigm for this DSL. "In TOP, a task is a specified piece of work aiming to produce a result of known type. When executed, tasks produce (temporary) results that can be observed in a controlled way. As work progresses it can be continuously monitored and controlled by other tasks. Tasks can either be fully automated, or can be performed by humans with computer support."

Complex workflows involving sensors, actuators, humans and communication in a distributed environment can be expressed as compositions of simpler tasks, using predefined and programmer-defined combinators. The DSL can facilitate the introduction of application-specific combinators in the form of higher-order functions. The description of timing constraints should be a central language feature in the DSL.

The technique of language embedding allows the use of a powerful host language in the embedded domain specific language. We are to start with Erlang; Erlang will be the host language. This choice is motivated by certain Erlang features: (i) Erlang is well-suited to programming distributed and concurrent systems and even more importantly, (ii) Erlang's concept of supervisor processes enables the easy implementation of fault-tolerance mechanisms.

We describe our framework, which is a good basis for bottom-up definition of workflows, in Section 3. The aim of a task-oriented programming DSL is to provide a syntax that resembles to natural language description of workflows, and focuses on the high level structures relevant to domain experts. Since its expressive power can lead to the definition of rather complex systems, and thus testing and verification might suffer from this complexity, the use

of the Task and Test Driven Development methodology is fostered. The top-down methodology can be improved by connecting with dialogue systems, and making the generation of workflows from application domain specific ontologies and natural language commands feasible.

3. The CPS Workflow System

In this section, a brief informal introduction to our framework is provided as follows. Section 3.1 classifies the entities of our system as special kinds of tasks. Section 3.2 presents a simple example that already utilizes the basic combinators. Finally, Section 3.3 gives a short description on embedding our DSL in Erlang.

3.1. **Everything is a task.** In a workflow system, tasks are first class citizens. This means that tasks can be arguments and results of other tasks, and, since we focus on a distributed execution environment, they can even be transmitted over the network. Tasks can be composed using "combinators", forming more complex tasks. In this approach, a complete workflow is a task as well.

Different kinds of tasks can be identified according to their function in a CPS workflow. The differentiation of tasks in our system can be seen in Figure 1.

A task consists of two parts: the description of its behavior, and the constraints on its execution. The behavior defines the computation the task performs when launched. The constraints part of a task specify spatial, timing and resource requirements, such as where to execute the task in a distributed environment, or what timeout triggers the cancellation of the task in the case of some failure.

We can distinguish two kinds of tasks. Primitive tasks are the simplest building blocks of workflows: interaction with a sensor or an actuator is a primitive task in a cyber-physical system. Moreover, any computation that is considered atomic according to the problem domain will be defined as a primitive task.

Combinators are tasks building new tasks from existing ones. They can be classified into two groups: a constructor combines the computational parts of its arguments, and a specificator establishes the constraints of a task. More details on combinators will be exposed through examples later on.

3.2. **Basic combinators.** As an illustration, let us consider a simple example, which contains already some of the main concepts of real-world cyber-physical systems, albeit in a small scale. We emphasize four issues regarding a CPS problem here: reading sensors, controlling actuators, operating under specified constraints, and using model-based predictions for correction modules.

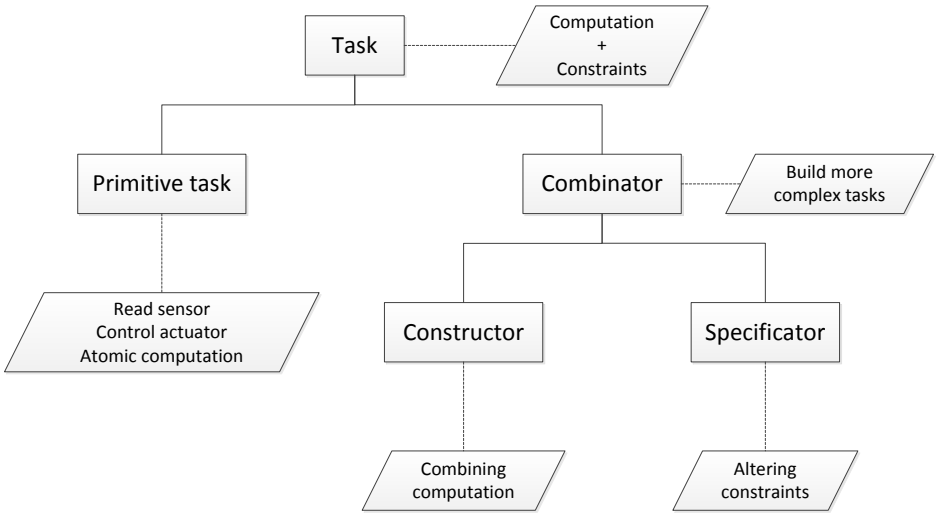


FIGURE 1. Everything is a task in a workflow system

In the example we want to bring water to boil in a kettle: we can switch on a coil to heat the water, we can check the water temperature regularly, and we can switch off the coil when the boiling point is reached. Constraints specify the location where the boiling water is needed, as well as the deadline when the water must reach 100^oC. Furthermore, we want to bring water to boil fault-tolerantly. For example, if the coil breaks down, we want to switch on another one in order to make sure that the water will eventually boil. We have a simple physical model: the temperature of the water is increasing continuously when the coil that heats the water is on. If we observe that this condition is not met, we conclude that either the coil or the thermometer is broken. By using more than one thermometers, and by cumulating sensory data, we can detect breakage of the thermometer. By using more than one coils, electricity to the broken coil can be cut off, and another coil can be switched on. A workflow implementing this functionality is depicted in Figure 2, and the source code of this workflow is presented in listing *kettle.wf*.

This workflow must be run a node connected to a kettle. Having the coil switched on, we start checking the temperature with three thermometers in parallel. We repeatedly read the thermometers, and the stream of sensory data is channeled to the model. We compute the average of the measured temperatures, and check whether the boiling point of water is reached. 97^oC is used due to sensor accuracy. The conditional control structure is provided by

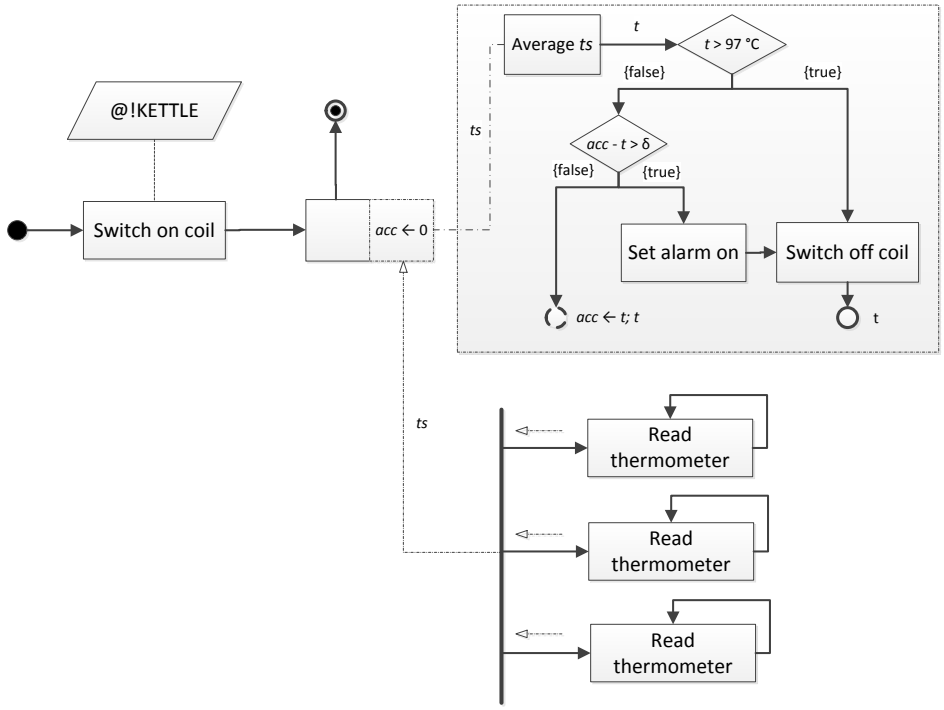


FIGURE 2. Bringing water to boil

the host language our DSL is embedded into. If the boiling point is reached, we switch off the coil, and stop the parallel tasks. If the water is not hot enough, we can compare the temperature against our physical model. If the temperature has not been raised since the last reading of the thermometers, we trigger an alarm and end the workflow. The state maintained by the model is "acc" (which stands for accumulator); it contains the previously read temperature data. Its initial value is 0, and the current temperature is saved into it at the end of each activation when the workflow is not stopped.

Now consider the building blocks of this simple workflow example. The primitive tasks here are the following: read a thermometer, switch on/off a coil and set the alarm. In the graphical representation two special symbols represent the starting and terminating points of the workflow.

What kind of combinators can we observe in this example? First of all, two tasks can be combined sequentially, which means that the second one will be launched when the first one ends. In some cases the result of the first task

```
-module(kettle).

-export(main/0).

main() ->
  Workflow =
    kettle_control:set_coil(on) >>|
    par([rec(kettle_control:read_thermometer(i) >>= fun(t) ->
          continue(t) end)
        | i <-[1,2,3]])
    controlled by fun(acc, ts) ->
      average(ts) >>= fun(t) ->
      if
        t > 97 ->
          kettle_control:set_coil(off) >>|
          return(t);
        acc - t > error_treshold ->
          kettle_control:set_alarm(on) >>|
          kettle_control:set_coil(off) >>|
          return({error, t})
        true ->
          continue({t, t})
      end
    end
  end
  with accumulator 0 @! [kettle],

  execute(Workflow).
```

. kettle.wf

is needed by the second. For instance, after computing the average of the values supplied by the three thermometers, we pass the result to the decision making task. (This is expressed by the `>>=` constructor in the DSL, which binds the value of the first task to a fresh variable – described as an "explicit `fun`-expression" in Erlang.) On the other hand, after switching the coil on, we can start reading the thermometers without passing any values from the first task to the second one. (This is expressed with the `>>|` constructor in the DSL.)

To describe parallel control flow, one can use the `par` constructor, which launches a number of tasks simultaneously. The compound task ends only if all of its components have ended, and its result is a list of the results of the components. Reading sensory data from the three thermometers is implemented with the parallel combinator in our example.

Workflows are often described in the style of reactive programming: the state of a subsystem can be monitored, and the workflow can react on changes of this state. According to our approach, we have a model which can plan an ideal trajectory in the problem space, and decide about error correction when deviation from that trajectory is detected.

Constraints can be associated to tasks, such as the spatial constraint `@!KETTLE` in the example. The specificator `@` can be used to impose requirements on the location where a task must be executed. In our example, we specify that the complete workflow must be launched on a node that is annotated with the "KETTLE" property. Specifying requirements on available resources as well as on user identities and roles is also possible in this way.

An interesting aspect of the `@` combinator is that it ensures implicit code transfer among nodes of a distributed system. Adaptivity of components in a CPS application and autonomy of tasks are fostered by allowing their code to be transfered by the workflow runtime in a transparent way. As an additional consequence, there is no need to deploy all the components of an application on all the nodes of the executing distributed execution environment, a minimal workflow runtime suffices: the code of the tasks can be transfered by the runtime to the appropriate node when needed, and hence the distributed execution environment can dynamically (re-)configure itself.

To make the workflow hierarchical, a special constructor, called a *controller* has been introduced. To understand its semantics, the concept of unstable values must be considered first. When a task ends, the value it results will never change: it is called a "stable value". In the case of sequences, when a task ends, its result is propagated forwards in the control flow as a stable value, but, in addition to this, it is also propagated backwards as an "unstable value". If we have a complex task, it may produce multiple unstable values before it reaches an end, and provides a stable value: its final result. As Figure 3 illustrates, the repeated read of a sensor (or any repeated execution of some task) will also yield a stream of unstable values.

Controllers provide a way to work with unstable values. The construct can be written as "`controlled by`" in the source code, and it binds a name to the unstable values observed. A controller has two tasks associated with it: a "controlled task" (of which the unstable values are observed by the controller), and a "plan". The plan is a task as well, with a very special meaning in the workflow. When a controller is launched, it launches the observed task. Every unstable value propagated from the observed task triggers the execution of the plan associated to the controller. Plans may maintain a state between different activations and can decide to stop the controlled task when a desired goal is reached, or in the case of faults/failures. Finally, when the controlled task ends, its stable value is propagated as the result of the controller.

FIGURE 3. Raising unstable values

Note that error detection and correction can be implemented in the plan of a controller. Therefore, controllers are the major means for model-based prediction and decision making in workflows. This is reflected in the kettle example as well.

3.3. **Behind the curtain – How the DSL is implemented.** As mentioned already in Section 2.3, workflow systems are described by using a software framework, which is in fact a simple programming language embedded into Erlang. It was a design decision that we implement distributed workflow systems in Erlang, since it is one of the most favored programming languages for implementing highly scalable, distributed, reliable and fault-tolerant software systems. In addition, we prefer this language also because it gives us the potential of employing the RefactorErl tool for statically analyzing and transforming the source code.

We chose Erlang despite the fact that it is not extensible, and it is certainly not suitable for DSL embedding. However, we found that the program transformation capabilities of RefactorErl can be easily turned into program translation capabilities; thus, we can extend the Erlang programming language with some key elements required for effective language embedding. The framework and the workflows are implemented in an extended version of the Erlang language, which is translated back to simple Erlang in one single step. The resulting program is compiled and run as any other Erlang application.

Since Erlang does not allow sending functions among different nodes of the network, computations cannot be handed over in an intuitive and effortless way. We developed support for so-called portable functions, realized as a compile-time transformation, which turns anonymous functions into complex data terms representing the computations along with their dependencies attached. On the other hand, the embedding of domain specific concepts into the language is mainly supported by the possibility of defining prefix and infix operators composed of natural language elements. Beside these two main components, we introduced some pieces of syntactic sugar that result in more natural workflow descriptions.

4. Conclusion

A programming methodology for cyber-physical systems has been proposed in this paper. The methodology emphasizes the introduction of modules with limited interaction, constraining the concurrent execution of interfering modules, and the need to avoid combinatorial explosion of test cases for validation.

A domain specific language for describing workflows can facilitate the development of CPS applications. The main concepts of such a language are tasks, combinators and constraints. In the CPS domain timing constraints are especially relevant.

We have presented a domain specific workflow language embedded into Erlang. This embedding provides the constructs of a powerful host language, as well as the superior distribution and fault tolerance capabilities of the Erlang programming model.

Acknowledgement

The research was carried out as part of the EITKIC_12-1-2012-0001 project, which is supported by the Hungarian Government, managed by the National Development Agency, financed by the Research and Technology Innovation Fund and was performed in cooperation with the EIT ICT Labs Budapest Associate Partner Group (www.ictlabs.elte.hu).

References

[1] J. Clune, J.B. Mouret, and H. Lipson. The evolutionary origins of modularity. *Proc. R. Soc. B*, 280(1755), 2013.

[2] M. S. A. Graziano. The organization of behavioral repertoire in motor cortex. *Annual Review of Neuroscience*, 29:105–134, 2006.

[3] B. Lijnse. *TOP to the rescue: Task-Oriented Programming for incident response*. PhD thesis, Radboud Universiteit Nijmegen, 2013. ISBN 978-90-820259-0-3, IPA Dissertation Series 2013-4.

Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

TEXT REPRESENTATION AND GENERAL TOPIC ANNOTATION BASED ON LATENT DIRICHLET ALLOCATION

DIANA INKPEN[(1)] AND AMIR H. RAZAVI[(2)]

Abstract. We propose a low-dimensional text representation method for topic classification. A Latent Dirichet Allocation (LDA) model is built on a large amount of unlabelled data, in order to extract potential topic clusters. Each document is represented as a distribution over these clusters. We experiment with two datasets. We collected the first dataset from the FriendFeed social network and we manually annotated part of it with 10 general classes. The second dataset is a standard text classification benchmark, Reuters 21578, the R8 subset (annotated with 8 classes). We show that classification based on the LDA representation leads to acceptable results, while combining a bag-of-words representation with the LDA representation leads to further improvements. We also propose a multi-level LDA representation that catches topic cluster distributions from generic ones to more specific ones.

1. Introduction

In order to improve the performance of text classification tasks, we always need informative and expressive methods to represent the texts [14] [16]. If we consider the words as the smallest informative unit of a text, there is a variety of well-known quantitative information measures that can be used to represent a text. Such methods have been used in a variety of information

Received by the editors: June 1, 2013.

2010 *Mathematics Subject Classification.* 62Fxx Parametric inference, 62Pxx Applications.

1998 *CR Categories and Descriptors.* code [**I.2.7 Natural Language Processing**]: Subtopic – *Text analisys* code [**H.3.1 Content Analysis and Indexing**]: Subtopic – *Linguistic processing*;

Key words and phrases. automatic text classification, topic detection, latent Dirichlet allocation.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

extraction projects, and in many cases have even outperformed some syntax-based methods. There are a variety of Vector Space Models (VSM) which have been well explained and compared, for example in [18]. However, these kinds of representations disregard valuable knowledge that could be inferred by considering the different types of relations between the words. These major relations are actually the essential components that, at a higher level, could express concepts or explain the main topic of a text. A representation method which could add some kind of relations and dependencies to the raw information items, and illustrate the characteristics of a text at different conceptual levels, could play an important role in knowledge extraction, concept analysis and sentiment analysis tasks.

In this paper, the main focus is on how we represent the topics of the texts. Thus, we select a LDA topic-based representation method. We also experiment with a multi-level LDA-based topic representation. Then, we run machine learning algorithms on each representation (or combinations), in order to explore the most discriminative representation for the task of text classification, for the two datasets that we selected.

2. Related work

In the most text classification tasks, the texts are represented as a set of independent units such as unigrams / bag of words (BOW), bigrams and/or multi-grams which construct the feature space, and the text is normally represented only by the assigned values (binary, frequency or term TF-IDF[1]) [17]. In this case, since most lexical features occur only a few times in each context, if at all, the representation vectors tend to be very sparse. This method has two disadvantages. First, very similar contexts may be represented by different features in the vector space. Second, in short instances, we will have too many zero features for machine learning algorithms, including supervised classification methods.

Blei, Ng and Jordan proposed the Latent Dirichlet Allocation (LDA) model and a Variational Expectation-Maximization algorithm for training their model. LDA is a generative probabilistic model of a corpus and the idea behind it is that the documents are represented as weighted relevancy vectors over latent topics, where a topic is characterized by a distribution over words. These topic models are a kind of hierarchical Bayesian models of a corpus [2]. The model can unveil the main themes of a corpus which can potentially be used to organize, search, and explore the documents of the corpus. In LDA models, a *topic* is a distribution over the feature space of the corpus and each document can be represented by several topics with different weights.

[1]term frequency / inverse document frequency

The number of topics (clusters) and the proportion of vocabulary that create each topic (the number of words in a cluster) are considered as two hidden variables of the model. The conditional distribution of these variables, given an observed set of documents, is regarded as the main challenge of the model.

Griffiths and Steyvers in 2004, applied a derivation of the Gibbs sampling algorithm for learning LDA models [9]. They showed that the extracted topics capture a meaningful structure of the data. The captured structure is consistent with the class labels assigned by the authors of the articles that composed the dataset. The paper presents further applications of this analysis, such as identifying *hot topics* by examining temporal dynamics and tagging some abstracts to help exploring the semantic content. Since then, the Gibbs sampling algorithm was shown as more efficient than other LDA training methods, e.g., variational EM and Expectation-Propagation [12]. This efficiency is attributed to a famous attribute of LDA namely, "the conjugacy between the Dirichlet distribution and the multinomial likelihood". This means that the conjugate prior is useful, since the posterior distribution is the same as the prior, and it makes inference feasible; therefore, when we are doing sampling, the posterior sampling become easier. Hence, the Gibbs sampling algorithms was applied for inference in a variety of models which extend LDA [19], [7], [4], [3], [11].

Recently, Mimno et al. presented a hybrid algorithm for Bayesian topic modeling in which the main effort is to combine the efficiency of sparse Gibbs sampling with the scalability of online stochastic inference [13]. They used their algorithm to analyze a corpus that included 1.2 million books (33 billion words) with thousands of topics. They showed that their approach reduces the bias of variational inference and can be generalized by many Bayesian hidden-variable models.

3. Datasets

The first dataset that we prepared for our experiments consists in threads from the FriendFeed social network. We collected main postings (12,450,658) and their corresponding comments (3,749,890) in order to obtain all the discussion threads (a thread consists in a message and its follow up comments). We filtered out the threads with less than three comments. We were left with 24,000 threads. From these, we used 4,000 randomly-selected threads as background source of data, in order to build the LDA model. We randomly selected 500 threads and manually annotated them with 10 general classes[2], to use as training and test data for the classification. The 10 classes are: *consumers,*

[2]We used only one annotator, but we had a second annotator check a small subset, in order to validate the quality of annotation. In future work, we plan to have a second annotator label all the 500 threads.

Class	No. of Training Docs	No. of Test Docs	Total
Acq	1596	696	2292
Earn	2840	1083	3923
Grain	41	10	51
Interest	190	81	271
Money-fx	206	87	293
Ship	108	36	144
Trade	251	75	326
Crude	253	121	374
Total	5485	2189	7674

TABLE 1. Class distribution of training and testing data for R8.

education, entertainment, life_stories, lifestyle, politics, relationships, religion, science, social_life and *technology.*

The second dataset that we chose for our experiments is the well-known R8 subset of the Reuters-21578 collection (excerpted from the UCI machine learning repository), a typical text classification benchmark. The data includes the 8 most frequent classes of Reuteres-21578; hence the topics that will be considered as class labels in our experiments are *acq, crude, earn, grain, interest, money, ship* and *trade.*

In order to follow the Sebastiani's convention [16], we also call the dataset R8. Note that there is also a R10 dataset, and the substantial difference between R10 and R8 is that the classes *corn* and *wheat*, which are closely related to the class *grain*, were removed. The distribution of documents per class and the split into training and test data for the R8 subset is shown in Table 1.

4. METHOD

We trained LDA models for each of the two datasets: one model on 4000 threads from FriendFeed and one model on all the R8 text data. LDA models have two parameters whose values need to be chosen experimentally: the number of topic clusters and the number of words in each cluster. We experimented with various parameter values of the LDA models.

For the first dataset, the best classification results were obtained by setting the number of cluster topics to 50, and the number of words in each cluster to maximum 15.

In LDA models, polysemous words can be member of more than one topical cluster, while synonymous words are normally gathered in the same topics. An example of LDA topic cluster for the first model is: "Google", "email",

"search", "work", "site", "services", "image", "click", "page", "create", "contact", "connect", "buzz", "Gmail", "mail". This could be labeled as *Internet*.

As mentioned, our 500 threads were manually annotated with the 10 generic classes. These classes, enumerated in section 3, are a manually generalized version of the 50 LDA clusters into the 10 generic categories. For the above example, the annotator placed it under the *technology* and *social_life* categories. The classification task is therefore multi-class, since a thread can be in more than one class. We trained binary classifiers from Weka [20] for each class, and averaged the results over all classes.

The manual mapping of LDA clusters into generic classes would allow us to automatically annotate more training data form the FriendFeed dataset, in our future work. Since each document has LDA clusters that were associated to it during the Gibbs sampling process, the generic classes for these clusters can be obtained, and one or more labels can be assigned to the document. Only the labels with high LDA weights will be retained. If the weights are low for all labels, the document would not be added to the training data. If more than one label has high weight, the document would have multiple labels. This process would allow us to add a large amount of training data, perhaps with some noise. For more details see [15].

For the classification task, we chose several classifiers from Weka: Naive Bayes (NB) because it is fast and works well with text, SVM since it is known to obtain high performance on many tasks, and decision trees because we can manually inspect the learned tree.

We applied these classifies on simple bag-of-words (BOW) representation, on LDA-based representations of different granularities, and on an integrated representation concatenating the BOW features and the LDA features. The values of the LDA-based features for each document are the weights of the clusters associated to the document by the LDA model (probability distributions).

5. Experiments and Results

The results on the first dataset are presented in Table 2. After stopword removal and stemming, the bag-of-words (BOW) representation contained 6573 words as features (TF-IDF values). The lower-dimensional representation based on LDA contained 50 features, whose values are the weights corresponding to the topic clusters. For the combined representation (BOW integrated with the LDA topics) the number of features was 6623.

We observed that the 10 class labels (general topics) are distributed unevenly over the dataset of 500 threads, in which we had 21 threads for the class *consumers*, 10 threads for *education*, 92 threads for *entertainment*, 28 threads

Representation / Classifier	Accuracy
BOW(TF-IDF)/ CompNB	77.22%
LDA Topics / Adaboost (j48)	69.32%
BOW(TF-IDF)+LDA / SVM(SMO)	80.00%

TABLE 2. Results on the FriendFeed dataset.

for *incidents*, 90 threads for *lifestyle*, 27 threads for *politics*, 58 threads for *relationships*, 31 threads for *science*, 49 threads for *social_activities*, and 94 threads for *technology*. Thus, the baseline of any classification experiment over this dataset may be considered as 18.8%, for a trivial classifier that puts everything in the most frequent class, *technology*. However, after balancing the above distribution through over/under sampling techniques, the classification baseline lowered to 10%.

On this dataset, we conducted the classification evaluations using stratified 10-fold cross-validations (this means that the classifier is trained on nine parts of the data and tested on the remaining part, then this is repeated 10 times for different splits, and the results are averaged over the 10 folds). We performed several experiments on a range of classifiers and parameters for each representation, to check the stability of a classifier's performance. We changed the *seed*, a randomization parameter of the 10-fold cross-validation, in order to avoid the accidental over-fitting.

For the BOW representation, the best classifier was Complement Naive Bayes (a version of NB that compensates for data imbalance), with an accuracy of 77.22%. Using the low-dimensional LDA representation, the accuracy goes down, but it has the advantage that the classifiers are faster and other classifiers could be used (that do not usually run on high-dimensional data). Combining the two representations achieved the best results, 80% accuracy.

The results on the second dataset, R8, are shown in Table 3. We experimented with several parameters for the LDA model: 8, 16, 32, 64, 128, and 256 for the number of clusters (therefore we build 6 models). We chose 20 words in each cluster. The reason we started with 8 clusters is that there are 8 classes in the annotated data. We experimented with combinations of the models in the feature representation (a multi-leveal LDA-based representation), leaving up to the classifier to choose an appropriate level of generalization.

After stopword removal and stemming, the BOW representation (TF-IDF values) contained 17387 words as the feature space. We experimented with each LDA representation separately, without good results; therefore we chose a combined 6-level representation (corresponding to the LDA models with 256, 128, 64, 32, 16, 8 clusters). For the integrated representation BOW with LDA

Representation / Classifier	Accuracy
BOW / SVM	93.33%
LDA Topics / SVM	95.89%
LDA+BOW / SVM	97.03%
BOW / NB	95.20%
LDA Topics / NB	94.61%
LDA+BOW / NB	95.52%
BOW / DT	91.54%
LDA Topics / DT	91.78%

TABLE 3. Results on the R8 dataset.

topics we had 17891 features (256 + 128 + 64 + 32 + 16 + 8 = 504, plus the 17387 words).

The average classification accuracy is very high, compared to a baseline of 51% (of a simplistic 8-way classifier that always chooses the most frequent class, *earn* in this dataset). The SVM and NB classifiers achieved the best results. These values are in line with state-of-the art results reports in the literature. We can compare our results with other reported classification results of the same dataset. According to the best of our knowledge, the accuracy of our integrated representation method on the Reuters R8 dataset, 97%, is higher than any simple and combinatory representation method from related work, which reports accuracies of 88%–95% [6], [1], [5], while 96% was reached with SVM on a complex representation method based on kernel functions and Latent Semantic Indexing [21].

For SVM, the LDA-based representation achieved better accuracy (95.89%) than the BOW representation (93.33%). This is due to the multi-level representation. When we experimented with each level separately, the accuracies dropped considerably. The best results over all the experiments were for SVM with the combined BOW and LDA-based representation.

6. Conclusions and Future Work

As our experimental results show, we can achieve good classification results by using a low-dimensional representation based on LDA. This representation has the advantage that allows the use of classifiers or clustering algorithms that cannot run on high-dimensional feature spaces. By using a multi-level representation (different generalization levels) we achieved better results than the BOW representation on the second dataset. In future work, we plan to test the multi-level representation on the first dataset, to confirm our hypothesis that it is better to let classifiers choose the appropriate level of generalization.

The combined BOW and LDA features representation achieved the best classification performance, and it can be used when there memory is not a concern, for classifiers that are able to cope with the large vector spaces.

Our results show that the first dataset is more difficult to classify than the second dataset. The reason is that it consists in social media texts, which are very noisy. In future work, we plan to experiment with more training data for the FriendFeed dataset (automatically annotated via the mapping of LDA clusters into the 10 classes), and to design representation and classification methods that are more appropriate for this kind of data.

Acknowledgments

The authors wish to thank Ontario Centres of Excellence and to The Natural Sciences and Engineering Research Council of Canada for the financial support. We thank Lana Bogouslavski for annotating the FriendFeed data and Dmitry Brusilovsky for his insights in the project.

References

[1] Charu C. Aggarwal and Peixiang Zhao. 2012. Towards graphical models for text processing; Knowledge Information Systems, DOI 10.1007/s10115-012-0552-3; Springer-Verlag London.

[2] David M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In Proceedings of the conference on Neutral Processing Information Systems NIPS 2003.

[3] David M. Blei and J. McAulie. 2007. Supervised topic models. In Proceedings of the conference on Neutral Processing Information Systems NIPS 2007.

[4] David M. Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. 2003. Journal of Machine Learning Research, 3: 993–1022.

[5] Ana Cardoso-Cachopo, Arlindo L. Oliveira. 2007. Combining LSI with other Classifiers to Improve Accuracy of Single-label Text Categorization. In Proceedings of the first European Workshop on Latent Semantic Analysis in Technology Enhanced Learning, EWLSATEL 2007.

[6] Yen-Liang Chen and Tung-Lin Yu. 2011. News Classification based on experts' work knowledge. In Proceedings of the 2nd International Conference on Networking and Information Technology IPCSIT 2011, vol.17 ; IACSIT Press, Singapore.

[7] Andrew McCallum and X. Wang. 2005. Topic and role discovery in social networks. In Proceedings of IJCAI 2005.

[8] J. R. Firth et al. 1957. Studies in Linguistic Analysis. A synopsis of linguistic theory, 1930–1955. Special volume of the Philological Society. Oxford: Blackwell.

[9] Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. In Proceedings of the National Academy of Sciences, 101 (Suppl 1), 5228–5235.

[10] Gregor Heinrich. 2004. Parameter estimation for text analysis, Technical Report (For further information please refer to JGibbLDA at the following link: http://jgibblda.sourceforge.net/)

[11] Wei Li and Andrew McCallum. 2006. Pachinko allocation: Dag-structured mixture models of topic correlations. In Proceedings of ICML 2006.

[12] Thomas Minka and John Lafferty. 2002. Expectation propagation for the generative aspect model. In Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence UAI 2002. https://research.microsoft.com/minka/papers/aspect/minka-aspect.pdf.
[13] David Mimno, M. Hoffman, and David M. Blei. 2012. Sparse stochastic inference for latent Dirichlet allocation. In Proceedings of International Conference on Machine Learning ICML 2012.
[14] Xiaoshan Pan and Hisham Assal. 2003. Providing context for free text interpretation. In Proceedings of Natural Language Processing and Knowledge Engineering, 704–709.
[15] Amir H. Razavi and Diana Inkpen. 2013. General Topic Annotation in Social Networks: A Latent Dirichlet Allocation Approach. In Proceedings of the 26th Canadian Conference on Artificial Intelligence (AI 2013), Regina, SK, Canada.
[16] Fabrizio Sebastiani. 2006. Classification of text, automatic. In Keith Brown (ed.), The Encyclopedia of Language and Linguistics, Volume 14, 2nd Edition, Elsevier Science Publishers, Amsterdam, NL, 457–462.
[17] Karin Spark Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28 (1): 11–21.
[18] Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics, Journal of Artificial Intelligence Research (JAIR), 37, 141–188.
[19] Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-Markov continuous-time model of topical trends. In Proceedings of ACM SIGKDD conference on Knowledge Discovery and Data Mining KDD 2006.
[20] Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, Morgan Kaufmann, San Francisco.
[21] Man Yuan, Yuan Xin Ouyang and Zhang Xiong. 2013. A Text Categorization Method using Extended Vector Space Model by Frequent Term Sets. Journal of Information Science and Engineering 29, 99–114.

(1) University of Ottawa, School of Electrical Engineering and Computer Science, Ottawa, ON, Canada, K1N 6N5
E-mail address: Diana.Inkpen@uottawa.ca

(2) University of Ottawa, School of Electrical Engineering and Computer Science, Ottawa, ON, Canada, K1N 6N5
E-mail address: araza082@eecs.uottawa.ca

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

HREBS AND COHESION CHAINS AS SIMILAR TOOLS FOR SEMANTIC TEXT PROPERTIES RESEARCH

DOINA TATAR[(1)], MIHAIELA LUPEA[(1)], AND EPAMINONDAS KAPETANIOS[(2)]

Abstract. In this study it is proven that the Hrebs used in Denotation analysis of texts and Cohesion Chains (defined as a fusion between Lexical Chains and Coreference Chains) represent similar linguistic tools. This result gives us the possibility to extend to Cohesion Chains (CCs) some important indicators as, for example the Kernel of CCs, the topicality of a CC, text concentration, CC-diffuseness and mean diffuseness of the text. Let us mention that nowhere in the Lexical Chains or Coreference Chains literature these kinds of indicators are introduced and used since now. Similarly, some applications of CCs in the study of a text (as for example segmentation or summarization of a text) could be realized starting from hrebs. As an illustration of the similarity between Hrebs and CCs a detailed analysis of the poem "Lacul" by Mihai Eminescu is given.

1. Introduction

Denotation analysis is a complex discipline concerned with the mutual relationships of sentences. An important tool used in Denotation analysis is the concept of *hreb* defined in [10] as a discontinuous text unit that can be presented in a set form or a list form, when the order is important. A hreb contains all entities denoting the same real entity or referring to one another in the text. This basic concept is baptized in this way in honor of L. Hebek ([3]) who introduced measurement in the domain of Denotation analysis, as it is known in Quantitative Linguistics. As we will show, the concepts as Lexical Chain or Coreference Chain (as in Computational Linguistics) subsume the notion of hrebs in the variant of word-hrebs. In fact, we are interested in this paper only in the notion of word-hrebs (for other kinds of hrebs: morpheme-hrebs, phrase-hrebs and sentence-hrebs see [10], [15]).

Received by the editors: March 28, 2013.

2010 *Mathematics Subject Classification.* 68T50,03H65.

Key words and phrases. Lexical Chains, Coreference Chains, Hrebs, Text segmentation, Text summarization.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

We will operate with the concept of Cohesion Chain (CC), defined as a Lexical Chain **or** a Coreference Chain, and we will show the relationship between CCs and hrebs (more exactly a slightly modified kind of word-hrebs, quasi-hrebs). Due to this relation, some denotational properties of a text defined using hrebs could be translated to CCs, in the benefit of the last ones. Similarly, some applications of CCs in the study of a text (as for example segmentation or summarization of a text) could be realized starting from quasi-hrebs.

The structure of the paper is as follows: Section 2 presents the concept of hreb and some indicators of a text connected with it. In Section 3 the Lexical Chains, the Coreference Chains, and their use in segmentation and summarization are introduced. In Section 4 we analyze a poem by Eminescu from the point of view of word-hrebs (as in [10]) and CCs. The paper ends with some conclusions and further work proposal.

2. Hrebs

A word-hreb contains all the words which are synonyms or refer to one of the synonyms. The hrebs usually are constructed using some rules such that a word belongs to one or more hrebs [10]. For example a verb with personal ending (1st and 2nd person) belongs to both the given verb and the person (subject) it overtly refers to. We will slightly modify the definition of a hreb eliminating the above syntactical constraint and will denote the new concept by *quasi-hreb*. Namely, for us verbs with personal ending (1st and 2nd person) belong to the given verb and don't have any connection with the hreb representing the subject of these verbs. In this way, a word belongs to only one *quasi-hreb*, similarly with the property that a word belongs to only one Lexical Chain or Reference Chain (Coherence Chain). The rest of the properties of hrebs mentioned in [10] are unmodified for *quasi-hrebs*: references belong to the quasi-hreb of the word they refer to, e.g. pronouns and Named Entities belong to the basic word; synonyms constitute a common quasi-hreb; articles and prepositions are not considered; adverbs may coincide with adjectives, and may belong to the same quasi-hreb.

According to the information and ordering of entities, [15] defines five kinds of hrebs:

(1) Data-hreb containing the raw data, e.g. words, and the position of each unit in text.

(2) List-hreb containing the data but without the positions of the units in the text.

(3) Set-hreb being the set containing only the lemmas (for word-hrebs).

(4) Ordered set-hreb is identical with (3) but the units are ordered according to a certain principle, e.g. alphabetically, or according to length, frequency, etc.

(5) Ordered position-hreb containing only the positions of units in the given text.

In our example in Section 4 we will use only the cases 1, 2 and 3.

Complete word-hreb analyses of several texts can be found in [15].

2.1. **Denotational analysis with hrebs.** Creating hrebs means a reduction of the text to its fundamental semantic components. Having defined them one can make statements both about the text and the hrebs themselves and obtain new indicators. A short introduction in these indicators is given below (for a complete presentation see [10]):

1. By lemmatizing the words occurring in a List-hreb, and eliminating the duplicates, the corresponding Set-hreb is obtained. If in a Set-hreb there are at least two words (different lemmas), then the hreb belongs to the $Kernel$ (core) of the text, i.e. if $|hreb_i| \geq 2$ then $hreb_i \in Kernel$. The hrebs of a $Kernel$ will be called *kernel hrebs*.

2. An important indicator of a text is the size of the $Kernel$, denoted by $|Kernel|$.

3. Topicality of a set-*kernel hreb* H_i, is calculated as:

$$T(H_i) = \frac{|H_i|}{|Kernel|}$$

4. Kernel concentration is defined as the size of the kernel divided by the total number n of hrebs in the text:

$$KC = \frac{|Kernel|}{n}$$

5. Text concentration is calculated based on the List-hrebs. If H_i is a List-hreb (containing all word-forms, not only lemmas) and L is the number of tokens in the text, then $p_i = |H_i|/L$ is the relative frequency of the List-hreb H_i. Text concentration TC is given as:

$$TC = \sum_{i=1}^{n} p_i^2$$

Relative text concentration, TC_{rel} is defined as:

$$TC_{rel} = \frac{1 - \sqrt{TC}}{1 - 1/\sqrt{n}}$$

6. Hreb diffuseness

The diffuseness D_H of a given hreb H with n_H elements, where the positions of tokens are (in an ascending order) $P = \{pos_1, ..., pos_{n_H}\}$, is defined using the maximal and minimal position of tokens occurring in it:

$$D_H = \frac{pos_{n_H} - pos_1}{n_H}$$

i.e. the difference of the last and the first position divided by the cardinal number of the hreb.

7. Mean diffuseness of the text is:

$$D_{Text} = \frac{1}{K} \sum_{j=1}^{K} D_{H_j}$$

where K is the number of kernel-hrebs ($|Kernel|$) in $Text$.

8. Finally, text compactness is defined as:

$$C = \frac{1 - n/L}{1 - 1/L}$$

where n is the number of hrebs in the text and L is the number of (word-)tokens.

3. Cohesion Chains

3.1. **Lexical Chains.** Lexical Chains (LCs) are sequences of words which are in a lexical cohesion relation with each other and they tend to indicate portions of a text that form semantic units ([8], [11], [5]). The most frequent lexical cohesion relations are the synonymy and the repetition, but could be also hypernyms, hyponyms, etc.. Lexical cohesion relationships between the words of LCs are established using an auxiliary knowledge source such as a dictionary or a thesaurus.

A Lexical Chain could be formalized as:

$$LC_i : [LC_i^1\ (Token_j), \cdots, LC_i^m\ (Token_k)]$$

where the first element of the chain LC_i is the word LC_i^1, representing the token with the number j in the text, the last element of the chain LC_i is the word LC_i^m, representing the token with the number k in the text (where $j < k$), the length of the chain LC_i is m. Because the analysis is made at the level of sentences, usually the sentences where the words occur are indicated. The representation in this case is:

$$LC_i : [LC_i^1\ (S_j), \cdots, LC_i^m\ (S_k)]$$

The first element of the chain LC_i is the word LC_i^1, and occurs in the sentence S_j , the last element of the chain LC_i is the word LC_i^m, and occurs in the sentence S_k of the text (where $j < k$).

LCs could further serve as a basis for Text segmentation and Text summarization (see [4]). The first paper which used LCs (manually built) to indicate the structure of a text was that of Morris and Hirst ([7]), and it relies on the hierarchical structure of Roget's thesaurus to find semantic relations between words. Since the chains are used to structure the text according to the attentional/intentional theory of Grosz and Sidner theory, ([1]), their algorithm divides texts into segments which form hierarchical structures (each segment is represented by the span of a LC). Some algorithms for linear segmentation (as opposite to hierarchical segmentation) are given in [12], [13], [14]. In all these algorithms it is applied the following remark of Hearst 1997 [2]: *There are certain points at which there may be radical changes in space, time, character configuration, event structure(...). At points where all of these change in a maximal way, an episode boundary is strongly present.* The algorithms are based on different ways of scoring the sentences of a text and then observing the graph of the score function. In this paper we introduce two new scoring functions for sentences (in the next subsection).

Let us remark that linear segmentation and the (extractive) summarization are two interdependent goals: good segmentation of a text could improve the summarization ([4]). Moreover, the rule of extracting sentences from the segments is decisive for the quality of the summary. Some largely applied strategies (rules) are ([12]):

1. The first sentence of a segment is selected.

2. For each segment the sentence with a maximal score is considered the most important for this segment, and hence it is selected (for example, the minima in the graph of the below $Score^1$ and $Score^2$ functions represent the sentences candidates for boundaries between segments of a text).

3. From each segment the most informative sentence (the least similar) relative to the previously selected sentences is picked up.

Thus, one can say that determining a segmentation of a text and using a strategy (1, 2 or 3), a summary of the text can be obtained, as well.

3.2. **Coreference Chains.** Coreference Chains are *chains of antecedents-anaphors* of a text. A complete study of Coreference Chains is the textbook [6]. A Coreference Chain contains the occurrences of the entities identified as antecedents for a given anaphor and also the occurrences of this anaphor.

The formalization of a Coreference Chain is as follows:

$$CR_i : [CR_i^1\ (Token_j), \cdots, CR_i^m\ (Token_k)], (where\, j < k\,)$$

or

$$CR_i : [CR_i^1\ (S_j), \cdots, CR_i^m\ (S_k)], (where\, j \leq k\,)$$

depending on the marks (tokens or sentences) picked out.

In the same way as the Lexical Chains, Coreference Chains express the cohesion of a text. The algorithms of segmentation (and summarization) of a text based on Lexical Chains could be adapted for Coreference Chains. In this paper we refer to both Lexical Chains and Coreference Chains by the name of Cohesion Chains.

3.3. **Scoring the sentences by Cohesion Chains.** Cohesion Chains (CCs) defined as in the above sections could be used to score the sentences such that when this score is low, cohesion is low, and thus the sentence is a candidate for a boundary between segments; similarly for a high score (a high cohesion) and the non-boundary feature of a sentence. In this paper we propose the following two new functions of score for sentences:

$$Score^1(S_i) = \frac{the\, number\, of\, tokens\, in\, S_i\, contained\, in\, at\, least\, one\, CC}{the\, number\, of\, tokens\, in\, S_i.}$$

Let us remark that $0 \leq Score^1(S_i) \leq 1$. When $Score^1(S_i) = 0$ (or close to 0), S_i is a candidate for a boundary between segments because S_i has a low connection with other sentences. When $Score^1(S_i) = 1$ (or close to 1), S_i is "very" internal for a segment. So, observing the graph of the function $Score^1(S_i)$ we could determine the segments of a text.

The second proposed scoring function is:

$$Score^2(S_i) = \frac{the\, number\, of\, CCs\, which\, traverse\, S_i}{the\, total\, number\, of\, CCs\, in\, the\, text}$$

Again $0 \leq Score^2(S_i) \leq 1$ and the above remarks remain valid: when $Score^2(S_i)$ is 0 (or close to 0), S_i is a candidate for a boundary between segments because S_i has a low connection with the other sentences. When $Score^2(S_i)$ is 1 (or close to 1), S_i is "very" internal for a segment.

As a final remark, let us observe that the hrebs (quasi-hrebs) could be used exactly in the same way to score the sentences: it is enough to put *quasi-hrebs* instead of *CCs* in the definitions for $Score^1(S_i)$ and $Score^2(S_i)$. Thus, hrebs (quasi-hrebs) could serve to segment and/or summarize texts.

In the same way, the indicators 1-8 used in Denotational analysis with hrebs could be extended to CCs. Let us remark that quasi-hrebs (and thus CCs) are defined in the Data-hrebs format. This is accordingly with the definition of Lexical Chains where the most important (frequent) lexical relation which is present in a Lexical Chain is the repetition [11]. The more frequently a word is repeated in a Lexical Chain, the more important this Lexical Chain

is. Obtaining CCs from Data-hrebs (duplicates are not eliminated), we will impose the condition to a kernel CC to have at least a given number of elements. In other words, a kernel CC must have a size bigger than a minimal one. Further, the topicality of a kernel CC, text concentration, CC-diffuseness and mean diffuseness of the text could be defined.

Let us mention that nowhere in the Lexical Chains or Coreference Chains literature these kinds of indicators are introduced up to now.

4. Example in Romanian

For the Eminescu's poem "Lacul" we will exemplify **hrebs, quasi-hrebs and CCs**, and the relationships between them. We will begin with the **Rules** for hrebs formation in Romanian language [10].

Rules of hrebs formation for the Romanian language

The Rules for hrebs are of the form: $a \in B$. Here a is an expression containing a special element called *pos* indicator which is written in italic (*pos* is for *part of speech*). Particularly, a could be formed only from the *pos* indicator. B is a (name for a) given hreb written with capital letters. More exactly, the Rule $a \in B$ means: a (or *pos* indicator of a) is an element of the hreb B . The connection between a and B will result from the word used for *pos* indicator. As a word-form could be contained in more than one hreb, in the application of rules it is possible to obtain a result as: $a \in B, C, \cdots$ meaning: a is an element of hreb B and hreb C and $\cdots$. The rules are valid only for the *pos* of a being *noun*, *verb*, *adjective*, *adverb*, *pronoun*.

RULES:

R1. $verb \in VERB$

R2. personal ending of a *verb*, which could be a *noun* or a *pronoun*, $\in NOUN\, or\, PRONOUN$

R3. synonym of a $verb \in VERB$

R4. *pronoun* referring to a $noun \in NOUN$

R5. *pronoun* referring to a Named Entity $\in NAMED\ ENTITY$.

R6. synonym of a Named Entity $\in NAMED\ ENTITY$.

R7. non-referring $pronoun \in PRONOUN$

R8. $noun \in NOUN$

R9. synonym of a $noun \in NOUN$

R10. $adjective \in ADJECTIVE$

R11. synonym of an $adjective \in ADJECTIVE$

R12. $adverb \in ADVERB$

R13.synonym of an $adverb \in ADVERB$

The Rules 1-13 could be summarized as follows: a noun, its synonyms, referring pronouns and personal endings in a verb belong all to the given noun; a Named Entity, its synonyms, referring pronouns and personal endings in a verb belong all to the given Named Entity; a verb in all its forms, its synonyms, belong to the given verb, however, the personal endings belong also to the respective noun; an adjective (adverb) and its synonyms belong all to the given adjective (adverb).

We illustrate the rules as applied to the poem "Lacul". Namely, we will make a denotation of tokens in the poem, then we will extract:

- **A. Hrebs** (Table 1),
- **B. Quasi-hrebs** (Table 2)
- **C. Cohesion Chains** (Table 3).

The tokens numbered are only nouns, verbs, adjectives, adverbs, and pronouns (in this poem do not exist Named Entities).

LACUL (denotation of tokens)

(S1) Lacul (1) codrilor (2) albastru (3)
Nuferi (4) galbeni (5) îl (6) încarcă (7).
(S2) Tresărind (8) în cercuri (9) albe (10)
El (11) cutremură (12) o barcă (13).
(S3) Şi eu (14) trec (15) de-a lung (16) de maluri (17),
Parc-ascult (18) şi parc-aştept (19)
Ea (20) din trestii (21) să răsară (22)
Şi să-mi (23) cadă (24) lin (25) pe piept (26).
(S4) Să sărim (27) în luntrea (28) mică (29) ,
Îngînaţi (30) de glas (31) de ape (32),
Şi să scap (33) din mână (34) cârma (35),
Şi lopeţile (36) să-mi (37) scape (38).
(S5) Să plutim (39) cuprinşi (40) de farmec (41)
Sub lumina (42) blîndei (43) lune (44).
(S6)Vîntu-n (45) trestii (46) lin (47) foşnească (48),
Unduioasa (49) apă (50) sune (51)!
(S7) Dar nu vine (52)... **(S8)**Singuratic (53)
În zadar (54) suspin (55) şi sufăr (56)
Lîngă lacul (57) cel albastru (58)
Încărcat (59) cu flori (60) de nufăr (61).

Hreb	Elements of Data-hreb	SDH	SSH
EU	(eu 14, **trec 15, -ascult 18, -aştept 19, scap 33**, -mi 23,		
(EU cont)	-mi 37 , **suspin 55, sufăr 56**)	9	8
LAC	(lacul 1, il 6, **tresărind 8**, el 11, **cutremură 12**, lacul 57)	6	5
EA	(ea 20, **răsară 22, cadă 24, vine 52**)	4	4
NUFĂR	(nuferi 4, **incarcă** 7, nufăr 61)	3	2
APĂ	(ape 32, apă 50, **sune** 51)	3	2
NOI	(**sărim 27, plutim 39**)	2	2
BARCĂ	(barca 13, luntrea 28)	2	2
TRESTIE	(trestii 21, trestii 46)	2	1
ALBASTRU	(albastru 3, albastru 58)	2	1
A PĂREA	(parc- 18, parc- 19)	2	1
LIN	(lin 25, lin 47)	2	1
A SCĂPA	(scap 33, scape 38)	2	1

TABLE 1. **A.** The hrebs with the size bigger than 1 (extracted from the poem **Lacul**)

4.1. **From Hrebs to Cohesion Chains.** By the application of the above mentioned rules a total number of 51 hrebs are obtained. From all these, only 12 hrebs presented in Table 1 contain more than one element. In Table 1 the hrebs are constituted as Data-hrebs, where SDH means "Size of Data-hreb" and SSH means "Size of Set-hreb".

The names of all 51 hrebs are as follows:

A ASCULTA, A AŞTEPTA, A ÎNCĂRCA, A CĂDEA, A CUTREMURA, A FOŞNI, A PĂREA, A PLUTI, A RĂSĂRI, A SĂRI, A SCĂPA, A SUFERI, A SUNA, A SUSPINA, A TRECE, A TRESĂRI, A VENI, ALB, ALBASTRU, APĂ, BARCĂ, BLÂNDĂ, CÂRMĂ, CERC, CODRU, CUPRINS, EU, EA, FARMEC, FLOARE, GALBEN, GLAS, ÎNCĂRCAT, ÎNGÂNAT, LAC, LIN, LOPATĂ, LUMINĂ, LUNĂ, LUNG, MAL, MÂNĂ, MIC, NOI, NUFĂR, PIEPT, SINGURATIC, TRESTIE, UNDUIOASĂ, VÂNT, ZADAR.

From the set of Rules R1-R13, the Rule R2 makes the difference when the quasi-hrebs are calculated. This rule is reproduced here:

R2. personal ending of a *verb*, which could be a *noun* or a *pronoun*, $\in NOUN\, or\, PRONOUN$

In Table 1 are bold marked all the verbs which are contained in a NOUN or PRONOUN hreb due to the Rule R2. All these verbs are not present in Table

Quasi-hreb	Elements of Data-hreb	SDH	SSH
EU	(eu 14, , -mi 23, -mi 37)	3	2
LAC	(lacul 1, il 6, , el 11, , lacul 57)	4	3
EA	(ea 20)	1	1
NUFĂR	(nuferi 4, nufăr 61)	2	1
APĂ	(ape 32, apă 50)	2	1
BARCĂ	(barca 13, luntrea 28)	2	2
TRESTIE	(trestii 21, trestii 46)	2	1
ALBASTRU	(albastru 3, albastru 58)	2	1
A PĂREA	(parc- 18, parc- 19)	2	1
LIN	(lin 25, lin 47)	2	1
A SCĂPA	(scap 33, scape 38)	2	1

TABLE 2. **B.** The quasi-hrebs extracted from the poem **Lacul**

2, the table of quasi-hrebs. As a remark, the hreb "NOI" is not a quasi-hreb, because both its elements (sărim 27, plutim 39) are obtained by Rule R2.

Let us remember that Lexical Chains are sequences of words which are in a lexical cohesion relation (synonymy, repetition, hypernymy, hyponymy, etc) with each other. Coreference Chains are chains of antecedents-anaphors of a text. Examining Table 2 of quasi-hrebs, we observe that: the quasi-hreb EU corresponds to a Coreference Chain (eu 14, -mi 23, -mi 37), the quasi-hreb LAC to a Coreference Chain (lacul 1, il 6, el 11, lacul 57). The quasi-hreb EA is not a chain (it has only one element). The rest of quasi-hrebs represent Lexical Chains: (nuferi 4, nufăr 61), (ape 32, apă 50), (barca 13, luntrea 28), (trestii 21, trestii 46), (albastru 3, albastru 58), (parc- 18, parc- 19), (lin 25, lin 47), (scap 33, scape 38). Table 3 contains the Cohesion Chains denoted as we will use further. We obtained CCs from the Data-hrebs, and the length of a Cohesion Chain is given in the SDH column, because the duplicates are not eliminated (as in SSH column).

Calculating the scores $Score^1$ for each sentence are obtained the following results:

$$Score^1(S_1) = 4/7 = 0.57$$
$$Score^1(S_2) = 2/6 = 0.33$$
$$Score^1(S_3) = 6/13 = 0.46$$
$$Score^1(S_4) = 5/12 = 0.42$$
$$Score^1(S_5) = 0/6 = 0.$$
$$Score^1(S_6) = 3/7 = 0.43$$
$$Score^1(S_7) = 0/1 = 0.$$

Denotation of CC	Elements of CC	Length of CC
CC1	(eu 14, -mi 23, -mi 37)	3
CC2	(lacul 1, il 6, el 11, , lacul 57)	4
CC3	(nuferi 4, nufăr 61)	2
CC4	(ape 32, apă 50)	2
CC5	(barca 13, luntrea 28)	2
CC6	(trestii 21, trestii 46)	2
CC7	(albastru 3, albastru 58)	2
CC8	(parc- 18, parc- 19)	2
CC9	(lin 25, lin 47)	2
CC10	(scap 33, scape 38)	2

TABLE 3. **C.** Cohesion Chains extracted from the poem **Lacul**

$Score^1(S_8) = 3/9 = 0.33$

Taking as segment boundaries the sentences with minimal score, the text is divided in 4 segments: $Seg1 = [S_1, S_2]; Seg2 = [S_3, S_5]; Seg3 = [S_6, S_7]; Seg4 = [S_8]$ or 3 segments: $Seg1 = [S_1, S_2]; Seg2 = [S_3, S_5]; Seg3 = [S_6, S_8]$ if mono-sentence segments are not permitted.

Scoring with $Score^2$ formula, the results are as following:

$Score^2(S_1) = 3/10 = 0.30$
$Score^2(S_2) = 4/10 = 0.40$
$Score^2(S_3) = 8/10 = 0.80$
$Score^2(S_4) = 9/10 = 0.90$
$Score^2(S_5) = 6/10 = 0.60$
$Score^2(S_6) = 6/10 = 0.60$
$Score^2(S_7) = 3/10 = 0.30$
$Score^2(S_8) = 3/10 = 0.30$

The text has only one segment $[S_1, S_8]$, with the most "internal" sentence S_4. A summary of the poem using $Score^1$ is formed by the sentences: S_1, S_3, S_6 and using $Score^2$, by the sentence S_1. In both cases the rule one (Section 3.1) has been applied.

4.2. **Indicators of Cohesion Chains.** Let us suggest how the indicators in Section 2.1 could be defined for the Cohesion Chains CC1 to CC10.

- Kernel CCs : Considering the minimal size of a kernel CC being 2, all CCs are in $Kernel$. Considering the minimal size of a kernel CC

being 3, only CC1 and CC2 are in $Kernel$. The last supposition is more realistic, since a CC has always at least 2 elements;

- The size of the $Kernel$ is 2, in the last above case;
- Topicality of the kernel CC denoted by CC1 is $3/2 = 1.5$ and topicality of CC2 is $4/2 = 2$;
- Kernel concentration is $KC = 2/10 = 0.2$;
- $p_1 = 3/61; p_2 = 4/61; p_i = 2/61$, $i = 3\,to\,10$. Text concentration is $TC = 0.0151$ and Relative Text concentration is $TC_{Rel} = 1.2830$;
- Diffuseness for each CC is as follows:
  $D_{CC1} = (37-14)/3 = 7.66; D_{CC2} = (57-1)/4 = 14; D_{CC3} = (61-4)/2 = 28.5; D_{CC4} = (50-32)/2 = 9; D_{CC5} = (28-13)/2 = 7.5; D_{CC6} = (46-21)/2 = 12.5; D_{CC7} = (58-3)/2 = 27.5; D_{CC8} = (19-18)/2 = 0.5; D_{CC9} = (47-25)/2 = 11; D_{CC10} = (38-33)/2 = 2.5$
- Mean diffuseness of the text is $D_{Text} = 10.75$;
- Text compactness is $C = (1-10/61)/(1-1/61) = 0.8505$.

The above indicators could make differences between CCs, such that some of them are kernel CCs, or have a higher topicality and/or diffuseness.

5. Conclusions and Further work

Lexical Chains and Coreference Chains (CCs) are intensively studied, but few indicators are standard for them. The indicators inspired from the hrebs must be studied and adopted for CCs. These indicators, in the context of some applications using CCs, could become instruments for the evaluation of these applications and for improving them. For example, there is a large debate about how to select CCs to construct the summaries of a text: selecting long or short CCs is one of the questions. Using only kernel CCs, or kernel CCs with a high topicality and /or high diffuseness could be a solution.

As a general remark, Quantitative Linguistics and Computational Linguistics are considered two distinct fields with their own journals, techniques and specialists. It is important to identify those parts they have in common, and to try to extract the advantage from this commonality. This paper is a step toward this desirable aim.

References

[1] B. Grosz and C. Sidner. 1986. "*Attention, Intentions and the Structure of Discourse*". Computational Linguistics 12: 175 - 204.

[2] M. Hearst 1997. "*TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages*". Computational Linguistics 23: 33 - 76.

[3] L. Hrebicek 1997. "*Lectures on Text Theory*". Prague: Oriental Institute.

[4] E. Kapetanios, D. Tatar and C. Sacarea. 2013. "*Natural Language Processing: semantic aspects*", Science Publishers, (to appear).

[5] A. Labadie and V. Prince. 2008. "*Finding text boundaries and finding topic boundaries: two different tasks?*" Proceedings of GoTAL08.
[6] R. Mitkov 2002. "*Anaphora Resolution*", Pearson Education, Longman.
[7] J. Morris and G. Hirst. 1991. "*Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text*". Computational Linguistics 17: 21 - 48.
[8] M. Okumura and T. Honda. 1994. "*WSD and text segmentation based on lexical cohesion*", 755 - 761. Proceedings of COLING-94.
[9] I.I.Popescu, J. Macutek, E. Kelih, R. Cech, K. H. Best, and G. Altmann. 2010. "*Vectors and Codes of Text*". Studies in Quantitative Linguistics 8, RAM Verlag.
[10] I.I. Popescu, M. Lupea, D. Tatar, and G. Altmann. 2013. "*Quantitative analysis of poetry*", Ed. Mouton de Gruyter, to appear.
[11] N. Stokes, J. Carthy and A.F. Smeaton. 2004. "*Select: a lexical cohesion based news story segmentation system*". AI Communications, 17(1): 3 - 12.
[12] D. Tatar, A. Mihis and D. Lupsa. 2008. "*Text Entailment for Logical Segmentation and Summarization*", 233 – 244. In Kapetanios, E., Sugumaran, V., Spiliopoulou, M. [eds.] Proceedings of 13th International Conference on Applications of Natural Language to Information Systems, London, UK. (LNCS 5039).
[13] D. Tatar, E. Tamaianu-Morita and G. Serban-Czibula. 2009. "*Segmenting text by lexical chains distribution*", Proceedings of Knowledge Engineering Principles and Techniques (KEPT 2009), University Press, Cluj-Napoca, Romania.
[14] D. Tatar, M. Lupea and Z. Marian. 2011. "*Text summarization by Formal Concept Analysis approach*". Proceedings of Knowledge Engineering Principles and Techniques (KEPT 2011), Cluj-Napoca, Romania.
[15] A. Ziegler and G. Altmann. 2002. "*Denotative Textanalyse*", Wien, Praesens.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania
E-mail address: dtatar@cs.ubbcluj.ro

E-mail address: lupea@cs.ubbcluj.ro

(2) University of Westminster, UK
E-mail address: E.Kapetanios@westminster.ac.uk

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

ON THE STUDY OF REDUCING THE LEXICAL DIFFERENCES BETWEEN SOCIAL KNOWLEDGE SOURCES AND TWITTER FOR TOPIC CLASSIFICATION

ANDREA VARGA[(1)], AMPARO CANO[(2)], FABIO CIRAVEGNA[(1)], AND YULAN HE[(2)]

Abstract. State-of-the-art approaches on *cross-source topic classification* (TC) of Tweets rely on building a supervised machine learning classifier on *Social Knowledge Sources* (KSs) (such as DBpedia and Freebase) for detecting topics of Tweets. These approaches typically employ various lexical, syntactical or semantic features derived from the content of these documents or Tweets, often ignoring other indicators to external data sources (e.g. URL), which can provide additional background information for cross-source TC. In order to address these limitations, in this paper we analyse various such indicators, and evaluate their impact on cross-source TC. Our experiments, evaluating the proposed TC in the context of Violence Detection (VD) and Emergency Response (ER) tasks, indicate that the *Twitter specific information (indicators)* contain valuable information; and thus incorporating them into a TC can improve the performance over previous approaches not considering them.

1. Introduction

Topic classification (TC) of Tweets has only started to gain attention very recently. It provides an efficient and effective way of organising and searching Tweets, which can then be useful for various tasks e.g. *relating topics to events* (such as an Airplane crash, Egypt revolution, Mexican drug war, etc.) ([11]), *summarisation* ([12]), *question answering* ([6]), *content filtering* ([16]) etc. State-of-the-art approaches on *cross-source topic classification* (TC) of Tweets rely on building a supervised machine learning classifier on *Social Knowledge*

Received by the editors: April 15, 2013.

2010 *Mathematics Subject Classification.* 68T50, 03H65.

1998 *CR Categories and Descriptors.* I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text Analysis.*

Key words and phrases. cross-source topic classification, linked knowledge sources, violence detection, emergency response.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

Sources (KSs) (such as *DBpedia* and *Freebase*) for detecting topics of Tweets [1, 7, 17]. These approaches typically employ various lexical (BoW) and entity-based features (BoE) derived from the sole content of these documents or Tweets, often ignoring other features which can act as indicators of specific external data sources (e.g. URL, hashtags), providing additional background information for cross-source TC. In order to address these limitations, in this paper we analyse various such *external data sources' indicators*, and evaluate their impact on cross-source TC.

To better understand how the Twitter specific information impacts a cross-source TC, we conducted an in-depth analysis of Tweets collected over a period of three months belonging to Violence Detection (VD) and Emergency Response (ER) situations, and answered the following research question: *Do information derived from external data sources' indicators play an important role in TC of Tweets?*

The main contribution of our work are thus as follows: i) *we investigate the impact of enhancing the content of a Tweet by leveraging external data-sources obtained from Twitter content indicators, namely URLs and hashtags* ii) *we provide a detailed analysis and comparison on three topics (i.e. War, Disaster and Accident, and Law and Crime) related to the VD & ER scenarios.*

Before studying the above research questions, in Section 2 we review related work on TC; in Section 3 we introduce and describe the main methodology we followed to enrich KSs and Tweets with additional background information. The experimental results are described in Section 4, and the main challenges that we faced are presented in Section 5. Conclusions are then drawn in Section 6.

2. Related Work

Existing approaches to topic classification of Tweets can be divided into two main strands: approaches utilising a single data source (single source TC) (e.g. data from Twitter or blogs) for TC and approaches utilising *social knowledge sources* (multi-source or cross-source TC) (such as DBpedia or Freebase) for TC.

In the former case, Genc et al. [3] proposed a latent semantic topic modelling approach, which mapped each Tweet to the most similar Wikipedia articles based on lexical features extracted from Tweets' content only. Song et al. [13] mapped a Tweet's terms to the most likely resources in the Probbase KS. These resources were used as additional features in a clustering algorithm which outperformed the simple BoW approach. Munoz et al. [10] proposed an unsupervised vector space model for detecting topics in Tweets in Spanish.

They used syntactical features derived from PoS (part-of-speech) tagging, extracted entities using the Sem4Tags tagger ([2]) and assigned a DBpedia URI for those entities by considering the words appearing in the context of the entity inside the Tweets. Vitale et al. [18] proposed a clustering based approach which augmented the BoW features with BoE features extracted using the Tagme system, which enriches a short text with Wikipedia links by pruning n-grams unrelated to the input text, showing significant improvement over the BoW features. Tao et al. [14] studied various Twitter dataset specific features (including whether a tweet contains a hashtag or a URL) for identifying whether a tweet is relevant to a topic, and showed that incorporating these features can help TC.

Considering the approaches exploiting data from KSs ; Michelson et al. [8] proposed an approach for discovering Twitter user' topics of interest by first extracting and disambiguating the entities mentioned in a Tweet. Then a subtree of Wikipedia category containing the disambiguation entity is retrieved and the most likely topic is assigned. Milne et al. [9] also assigned resources to Tweets. In their approach they make use of Wikipedia as a knowledge source, and consider a Wikipedia article as a concept, their task then is to assign relevant Wikipedia article links to a Tweet. They proposed a machine learning approach, which makes use of Wikipedia n-gram and Wikipedia link-based features. Xu et al. [19] proposed a clustering based approach which linked terms inside Tweets to Wikipedia articles, by leveraging Wikipedia's linking history and the terms' textual context information to disambiguate the terms meaning. In Varga et al. [17], we studied the similarity between KSs and Twitter using both BoW and BoE features, showing that DBpedia and Freebase KSs contain complementary information for TC of Tweets, with the lexical features achieving the best performance. More recently, in Cano et al. ([1]) we demonstrated that exploiting the semantic information about entities from DBpedia and Freebase is beneficial, and incorporating additional semantic information about entities in terms of properties and concepts can furthermore improve the performance of TC against the sole Twitter data approach. Consequent work, classifying blog posts into topics ([5]) has also demonstrated that selecting data from Freebase using distant supervision in addition to incorporating features about named entities is beneficial for TC.

Whilst previous work already focused on incorporating lexical and semantic features into TCs, these features were extracted from the sole content of Tweets. However, due to the length constraints of Twitter messages, these short messages often contain various other information (e.g. URLs or hashtags), which can further help the understanding of the content of the messages. The usefulness of Twitter specific features (such as "has_URL", "has_ hashtag") has already been shown to be beneficial for single source TC case ([14]).

However, to date no study has been conducted to validate whether these data-source specific indicators are also useful in cross-source TC scenarios as well.

3. Enriching KSs and Twitter with additional background information

In previous work [1, 17] we have investigated the use of *Social Knowledge Sources* (e.g. DBpedia, Freebase) for building cross-source topic classifiers, which can aid in the topic classification of Tweets. In such approaches we leverage the entities appearing in both KS and Twitter content for deriving semantic features, enabling to reduce the distributional differences across datasets. One of the main reasons for the distributional differences lies in the variation in vocabulary, writing style and format of the documents across data sources.

In this paper, however, we introduce a novel approach which leverages two main type of external source indicators for reducing the differences for both lexical and entity features across datasources. Figure 1, presents a tweet highlighting entities, links and hashtags.

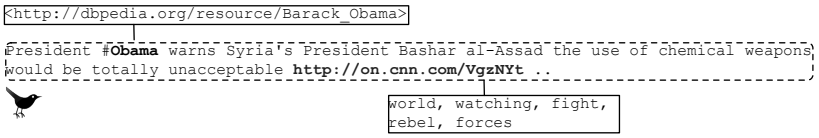


Figure 1. Enriching tweet content by using hashtags and links as indicators of external sources.

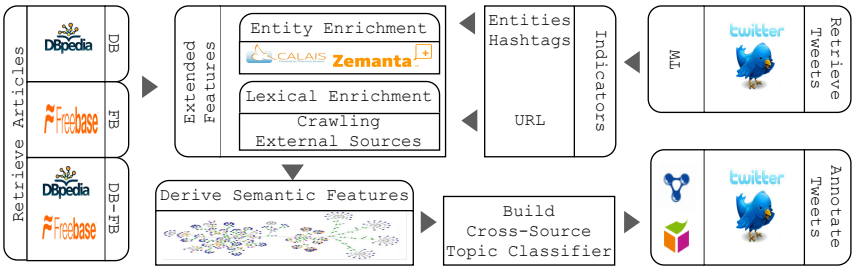


Figure 2. Architecture of cross-source TC based on lexical and entity-based feature enrichment derived from specific indicators (e.g. hashtags, links and entities).

In our approach we propose to reduce the lexical and entity differences between KSs and Twitter by: i) incorporating lexical features derived from external sources pointed out by *links indicators*; and by ii) incorporating KS entity-based features derived from resources embedded in *hashtags indicators* (e.g. #Egypt, #Obama).

We propose the architecture summarised in Figure 2 for building a cross-source TC. The proposed architecture involves the use of three datasets, two of them consisting of articles derived from DBpedia (DB) and Freebase (FB) KSs, and a third one consisting of a collection of tweets (TW). This architecture comprises the following stages:

1) *KS Data collection* - Given a topic c, KS-derived datasets (DB and FB) are generated by SPARQL querying for those articles whose categories and subcategories are c.
2) *Feature Extraction* - For both KS-datasets and the TW dataset, lexical features represented by a bag of words (BoW) are extracted and weighted using a TF-IDF weighting function, keeping only the top 1000 words. For the KS-datasets, a bag of entities (BoE) features is also generated, using the OpenCalais[1] and Zemanta[2] name entity recognition services.
3) *Indicator Extraction* - For the TW dataset, URL and hashtag (HSH) indicators are extracted. These indicators will be referred to as bag of links (BoL) and bag of hashtags (BoH) respectively.
4) *Incorporating background information from indicator features* - In order to overcome the number of character limitation posed by Tweets, the feature space representing a tweet is extended via the BoL and BoH indicators as follows:
   BoL based features. - Each URI from the BoL is resolved and the content of the referenced website is parsed. For each link the following lexical features are kept: i) the title of the page (BoL (T)); ii) the first paragraph of the page (BoL (1)); and iii) the last paragraph of the page (BoL (L)).
   BoH based features. - In order to assign a semantic meaning to a hashtag we implemented a series of regular expressions, which relate a term inside a hashtags to a DBpedia or Freebase resource URI (e.g. #egypt will be associated with `dbpedia.org/resource/Egypt` and `freebase:Egypt`). These resources, which we refer to as bag of resources are later on used as pointers to enable semantic enrichment.
5) *Semantic Feature Enrichment* - The semantic enrichment consists on extending a feature space with ontological classes and properties which characterise a KS resource URI. For example, the resource `dbpedia.org/resource/`

[1] http://opecalais.com
[2] http://zemanta.com

`Barack_Obama` is related to rdf types such as *yago:PresidentsOfTheUnitedStates* and *yago:NobelPeacePrizeLaureates*, and is characterised by properties such as *dbpedia-owl:commander* and *dbpedia-owl:knownFor*. The described semantic enrichment was applied to both KS BoE features and to the TW bag of resources derived from the bag of hashtags. We weighted the class features by frequency, while the property features were weighted following the specificity-generality weighting function introduced in [1].

6) *Building Cross-source topic classifier* - For our experiments, we considered as the base cross-source classifier a supervised TC classifier (SVM DB-FB) trained on the joint DBpedia (DB) and Freebase (FB) KSs, which was found to perform best for the topic classification task ([1]). This classifier takes into account both lexical, entity and semantic features introduced in the above stages of this architecture.
7) *Annotating Tweets* Finally tweets are annotated as belonging or not to the given topic c.

4. Experiments

To understand how the different information provided by external sources influence the performance of a TC, we evaluated our framework on a series of experiments, and conducted an analysis on a corpus of Tweets compiled over three months.

In our analysis we investigated the research questions of *Do information derived from external data sources indicators play an important role in TC of Tweets?*

4.1. Dataset characteristics.

For building our single source and cross-source TCs, we used the same dataset collected in our previous work ([1]), consisting of 9,465 articles from DB, 16,915 articles from FB and 10,189 tweets from Twitter (TW), covering multiple topics including three specific to ER &VD tasks (Disaster ($DisAcc$), Crime (Cri) and War (War)).

The general statistics about the TW dataset are summarised in Table 1. As we can observe, in the TW dataset the frequency of hashtags (HSH) and URLs is relatively low, indicating that only a small number of Tweets contain external data source specific information. In total 2,386 (23,41%) tweets contain at least one hashtag; and 3,348 (32.85%) Tweets contain at least one URLs. The number of unique hashtags is 1,784; while the number of unique URLs is 1,902.

The concept statistics derived for each entity in the KSs dataset are summarised in Table 2.

Topic	%Hsh	#Hsh	%URL	#URL	#dbCls (HSH)	#yagoCls (HSH)	#fbClass (HSH)
DisAcc	1.85%	233	2.59%	154	29	150	316
Cri	2.78%	220	6.41%	411	23	169	312
War	2.10%	198	2.65%	139	20	171	215

TABLE 1. Twitter dataset statistics about external data-source indicators (HSH, URL). #dbCls(HSH) refers to the number of unique KSs concepts derived for HSH from DB ontology; #yagoCls (HSH) refers to the number of unique KSs concepts derived for HSH from Yago ontology; and #fbClass (HSH) stands for the number of unique FB concepts derived for HSH from Freebase ontology.

Topic	#dbCls (BoE)	#yagoCls (BoE)	fbClass (BoE)
DisAcc	119	3,865	1,289
Cri	119	3,865	1,289
War	124	3,864	1,215

TABLE 2. Concept statistics in the multi-source DB-FB KS dataset. #dbCls (BoE) refers to the number of unique DB concepts derived for the named entities from DB ontology; #yagoCls (BoE) refers to the number of unique KSs concepts derived for entitites from Yago ontology; and #fbClass (BoE) stands for the number of unique FB concepts derived for entities from Freebase ontology.

When augmenting the single source TC classifier with concept and property features, we used a reduced vocabulary consisting of 180 unique concepts and properties from KSs, which we empirically set.

4.2. **Results.** We employed SVM classifiers for both single source (SVM TW), and cross-source (SVM DB-FB) classifiers to classify tweets into relevant topics. When training the classifiers, we split the TW dataset up into a training/testing set using an 80:20 split. This resulted in that the SVM TW classifier was trained on 80% of the TW dataset, while the SVM DB-FB classifier was trained on the full KSs data together with 80% of TW data. The test set

in both cases consists of 20% of TW data, and the results were averaged over five independent runs.

Given the sparse distribution of HSHs and URLs in Tweets, we performed two series of experiments. In the first set of experiments we utilised the full set of TW data (10,189 Tweets), which we denote as *Full*. In our second set of experiments, we only considered Tweets having at least one HSH or URLs (resulting in 4,778 Tweets), which we refer to *Filt*.

Case	Features	*DisAcc*			*Cri*			*War*		
		P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
Full	BOW	0.776	0.635	0.699	0.724	0.489	0.584	0.842	0.745	0.791
	BoL(1)	0.791	0.644	***0.710***	0.720	0.522	0.605	0.880	0.737	0.802
	BoL(L)	0.796	0.640	***0.710***	0.716	0.525	0.605	0.885	0.739	0.806
	BoL(T)	0.778	0.626	0.694	0.723	0.507	0.596	0.872	0.732	0.796
	BoH(Cls)	0.773	0.644	0.703	0.687	0.540	0.605	0.861	0.745	0.799
	BoH(P)	0.783	0.649	0.709	0.695	0.544	***0.610***	0.882	0.752	***0.812***
Filt	BOW-Filt	0.877	0.498	0.635	0.749	0.400	0.522	0.955	0.624	0.755
	BoL(1-Filt)	0.801	0.509	0.623	0.725	0.474	0.574	0.839	0.698	0.762
	BoL(L-Filt)	0.801	0.509	0.623	0.727	0.474	0.574	0.839	0.698	0.762
	BoL(T-Filt)	0.813	0.497	0.617	0.766	0.488	0.596	0.874	0.714	0.786
	BoH(Cls-Filt)	0.810	0.523	***0.636***	0.733	0.488	0.586	0.868	0.724	0.790
	BoH(P-Filt)	0.796	0.515	0.625	0.747	0.526	***0.617***	0.892	0.746	***0.813***

TABLE 3. The performance of the SVM TC using extrenal *data source indicators*.

Table 3 summarises the results obtained for the single-source TC case. When considering the full TW corpus, we can observe that the classifier built using BoL and BoH features improve upon the baseline classifier considering words only (BoW), except for the *DisAcc* topic using *BoL (T)* features. The best overall results were obtained by the *BoH (Prop)*, achieving an improvement of 2.6% over the baseline for the *Cri* topic, and an improvement of 1.5% for the *War* topic. These results are also in agreement with our previous findings ([1]), showing that the property features provide useful information for TC, and also incorporating them into TC is more beneficial than utilising concept features.

Considering the results on the filtered TW corpus, we again found the *BoH (Prop)* features to perform the best, except for the *DisAcc* topic. The improvement over the baseline classifier, however, was much bigger in this case: 4.3% for the *Cri* topic, and 5.8% for the *War* topic. An explanation for the small improvement for the *DisAcc* topic can be understood by the fact that the Tweets belonging to the *DisAcc* topic contain the less number of HSHs and URLs, and therefore less number of Tweets are semantically enriched.

Looking at the individual features derived from the URLs, in the Filt case, when most of the tweets have a URL inside them, the Title of the articles was

Case	Features	*DisAcc*			*Cri*			*War*		
		P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
Full	BOW	0.955	0.869	0.910	0.944	0.857	0.898	0.955	0.861	0.905
	BoL(1)	0.955	0.867	0.908	0.943	0.857	0.898	0.958	0.871	0.913
	BoL(L)	0.955	0.866	0.908	0.945	0.859	0.900	0.958	0.868	0.911
	BoL(T)	0.953	0.862	0.905	0.944	0.854	0.897	0.959	0.868	0.911
	BoH(Cls)	0.959	0.979	***0.969***	0.946	0.974	***0.960***	0.964	0.984	***0.974***
	BoH(P)	0.955	0.900	0.927	0.947	0.895	0.920	0.958	0.902	0.929
Filt	BOW-Filt	0.842	0.409	0.550	0.711	0.386	0.500	0.956	0.823	0.885
	BoL(1-Filt)	0.766	0.673	0.716	0.917	0.813	0.862	0.956	0.827	0.887
	BoL(L-Filt)	0.957	0.841	0.895	0.914	0.805	0.856	0.958	0.824	0.886
	BoL(T-Filt)	0.958	0.834	0.892	0.917	0.814	0.863	0.961	0.822	0.886
	BoH(Cls-Filt)	0.953	0.986	***0.969***	0.918	0.966	***0.941***	0.964	0.981	***0.973***
	BoH(P-Filt)	0.956	0.876	0.914	0.919	0.842	0.879	0.960	0.868	0.912

TABLE 4. The performance of the DB-FB cross-source SVM TC using various external *datasource indicators* .

found to be more informative of a topic. However, in the Full case, the first and the last paragraphs of the webpages were found more beneficial than the title of the webpages.

The results corresponding to the cross-source scenario are presented in Table 4. We can observe different trends compared to the single source scenario. For the case of the full TW dataset, the best cross-source feature for all the three scenarios was the *BoH (Cls)* feature. The highest improvement of 6.2% being achieved for the *Cri* topic. We can furthermore notice, that the results for the *BoH (Prop)* features are also outperforming the results obtained by the URL features. These results indicate, that incorporating semantic information derived from KSs are very important in reducing the lexical gap between KSs and TW. In particular, the addition of new words derived from the external URL websites were found worst or achieved little improvement over the baseline BoW case (for *DisAcc*, *Cri*). With respect to the URL features, we can notice that the performance of the classifier does not change drastically when utilising the first, last or the title of external URL websites. The difference in the performances is less than 1%.

Examining the results obtained for the Filtered case, we can observe similar trends, where again the *BoH (Cls)* feature exhibit the highest performance for each topic, which is then followed by the *BoH (Prop)* features. Considering the URL features, however, we can notice that the title of the websites seems to be more beneficial for TC, than the first or the last paragraphs. An explanation for this could be, that in this Filtered scenario more tweets are affected by feature augmentation than in the Full scenario. In light with the results for the single-source scenario, we can also observe a bigger improvement (up to 44.2% for *Cri*) in the Filt case than in the Full case.

5. Challenges and Limitations

In this work we enriched the representation of short text messages with information from external websites with the goal of reducing the lexical differences between KSs and Twitter.

One of such external link indicators were the hashtags from a Tweet, for which we assigned a DBpedia and Freebase URI using a simple word matching approach. We encountered various challenges, given that hashtags can often contain: (1) abbreviations (e.g. #nkorea `http://dbpedia.org/resource/North_Korea`); (2) contain compound words (e.g. #flightdelay `http://dbpedia.org/resource/Flight_delay`); and (3) some of the hastags may contain new abbreviations not present in KSs (e.g. #emfrmf) . For those cases no semantic meaning was assigned to them. In addition, one hashtag as any other word (#beirut) may have multiple meanings (e.g. the capital city of Lebanon; or a Lebanese governorate), and thus in order to assign the correct DB and FB URI one may apply a word sense disambiguation algorithm ([15]) first, which takes into account not only the lexical form of a hashtag but also the context of the hashtag.

The automatic extraction of sentences and paragraphs from external websites also poses challenges. One of the main problems considering this task is the accurate identification of boundaries on a page, since different websites employ different formats for describing the content of their pages. Particularly in pages where users can add comments (e.g. newswire articles and forum-like pages) the identification of the last paragraphs becomes challenging. In our work we parsed a full webpage as a whole, independently of whether the last part of the page referred to users' comments or not.

6. Conclusion and Future work

This study presented an approach for incorporating various background information into cross-source TCs built on multiple linked KSs. The goal of our study was to investigate whether incorporating such information can furthermore reduce the lexical differences between KSs and Twitter -imposed by the short length nature of Tweet messages-, thus allowing the creation of more accurate TC of Tweets.

We looked at two Twitter specific indicators including hashtags and URLs, for which we derived additional lexical and semantic features for training cross-source TCs. Our results on both sole Twitter and cross-source settings reveal that the indicator which provides a better feature enrichment, and therefore better classification performance was the hashtags.

Our future effort will consist on investigating alternative ways for bridging the gap between KSs and Twitter. One possible future direction could be to

investigate the impact of tweet normalisation approaches for cross-source TC, aiming at resolving the abbreviation, misspelling to standard English words ([4])[3]

7. References

[1] Amparo Elizabeth Cano, Andrea Varga, Matthew Rowe, Fabio Ciravegna, and Yulan He. Harnessing linked knowledge sources for topic classification in social media. In *24th ACM Conference on Hypertext and Social Media, HT '13*. ACM, 2013.

[2] A. Garcia-Silva, Oscar Corcho, and J. Gracia. Associating semantics to multilingual tags in folksonomies, 2010.

[3] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V. Nickerson. Discovering context: classifying tweets through a semantic transform based on wikipedia. In *Proceedings of the 6th international conference on Foundations of augmented cognition: directing the future of adaptive systems*, FAC'11, pages 484–492, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[5] Stephanie Husby and Denilson Barbosa. Topic classification of blog posts using distant supervision. In *Proceedings of the Workshop on Semantic Analysis in Social Media*, pages 28–36, Avignon, France, April 2012. Association for Computational Linguistics.

[6] Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang. Question identification on twitter. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 2477–2480, New York, NY, USA, 2011. ACM.

[7] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining, WSDM '12*.

[8] Matthew Michelson and Sofus A. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, New York, NY, USA, 2010.

[9] D. Milne and I. H. Witten., editors. *Learning to link with Wikipedia.* 2008.

[10] Óscar Muñoz García, Andrés García-Silva, Óscar Corcho, Manuel de la Higuera Hernández, and Carlos Navarro. Identifying Topics in Social Media Posts using DBpedia. In Meunier Jean-Dominique, Halid Hrasnica, and Florent Genoux, editors, *Proceedings of the NEM Summit*, pages 81–86. NEM Initiative, Eurescom ? the European Institute for Research and Strategic Studies in Telecommunications ? GmbH, September 2011.

[3]In this work, we performed some initial experiments applying a dictionary based approach for Tweet normalisation. We built a lexicon from *http://www.noslang.com/* website, consisting of 5,407 abbreviation word pairs, and replaced all abbreviations found in tweets with standard English terms. Our initial results, however, showed no improvement upon the baseline model without normalisation. Our future work will thus aim to investigate other tweet normalisation approaches too.

[11] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[12] Beaux Sharifi, Hutton, Mark-Anthony, and Jugal Kalita. Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 685–688, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[13] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, pages 2330–2336, 2011.

[14] Ke Tao, Fabian Abel, Claudia Hauff, and Geert-Jan Houben. What makes a tweet relevant for a topic? In *Making Sense of Microposts (#MSM2012)*, pages 49–56, 2012.

[15] Doina Tatar. Word sense disambiguation by machine learning approach: A short survey. *Fundam. Inf.*, July 2004.

[16] Irina Temnikova, Dogan Biyikli, and Francis Boon. First steps towards implementing a sahana eden social media dashboard. In *Proceedings of the conference Social Media and Semantic Technologies in Emergency Response (SMERST 2013)*, Coventry, UK, 2013.

[17] Andrea Varga, Amparo Elizabeth Cano, and Fabio Ciravegna. Exploring the similarity between social knowledge sources and twitter for cross-domain topic classification. In *Proceedings of the Knowledge Extraction and Consolidation from Social Media, 11th International Semantic Web Conference (ISWC2012)*, 2012.

[18] Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. Classification of short texts by deploying topical annotations. In *ECIR*, pages 376–387, 2012.

[19] Tan Xu and Douglas W. Oard. Wikipedia-based topic clustering for microblogs. *Proc. Am. Soc. Info. Sci. Tech.*, 48(1):1–10, 2011.

(1) The Organisations, Information and Knowledge (OAK) Group, Department of Computer Science, University of Sheffield, UK
E-mail address: a.varga@dcs.shef.ac.uk

(2) School of Engineering and Applied Science, Aston University, UK
E-mail address: ampaeli@gmail.com

E-mail address: f.ciravegna@dcs.shef.ac.uk

E-mail address: y.he9@aston.ac.uk

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

WEIGHTED MAJORITY RULE FOR HYBRID CELLULAR AUTOMATA TOPOLOGY AND NEIGHBORHOOD

ANCA ANDREICA[(1)] AND CAMELIA CHIRA[(1)]

Abstract. Evolution rules for Cellular Automata (CAs) able to perform computational tasks which require global coordination highlight an interesting emergent behavior. CAs can generate this complex behavior starting from a simple initial configuration based on the local interaction of simple components that evolve according to some state change rule. However, the detection of rules that exhibit coordinated global information processing is a very challenging task highly important in the study of complex systems. In this paper, we propose a new weighted rule for a cellular automaton with hybrid topology and neighborhood in which the state of a cell changes according to the cell itself and both local and long-distance cells. In the proposed approach, each cell in the neighborhood has a different weight (determined using an evolutionary algorithm) in the decision of changing the state for the current cell. Computational experiments focus on the well-known density classification task for the one-dimensional binary-state CA. Results support a better performance of the proposed weighted rule compared to the standard majority rule applied to the same CA topology.

1. Introduction

Cellular Automata (CAs) represent useful and important tools in the study of complex systems and interactions. A cellular automaton is a system evolving in discrete time steps with a discrete spatial geometry (usually a regular lattice). The CA is specified in terms of rules that define how it changes and evolves in time. A global coordinated behavior results from the local interaction of simple components [25, 15]. The emergent behavior and computational

Received by the editors: March 30, 2013.

2010 *Mathematics Subject Classification.* 68-02.

1998 *CR Categories and Descriptors.* I.2.8 Computing Methodologies [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

Key words and phrases. evolutionary algorithms, density classification task, cellular automata.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

complexity of a system can be analyzed and better understood based on CA dynamics [2].

One of the most widely studied CA problems is the density classification task (DCT) [10, 21, 16, 18, 2]. This is a prototypical distributed computational task with the objective of finding the density most present in the initial cellular state. The discovery of rules exhibiting a high degree of global self-organization is not easily achieved. DCT is not a trivial task since coordinated global information processing must rise from the interactions of components with local information and communication.

Most existing studies [6, 3, 15, 9, 16, 20, 14, 11, 18] focus on developing algorithms able to find high performant rules for one-dimensional CAs with fixed neighborhood size. Each iteration, the change of a cell state depends on the cell itself and r local cells neighbors on each side. In this case, the neighborhood size is fixed to $2 * r + 1$ and the method searches for the best rule able to correctly classify potentially any initial configuration. Network-based CAs have also been investigated in the context of DCT [23, 24, 4, 5, 22]. The topology of the CA is given by a general graph and the neighborhood of each cell varies with the number of connecting nodes in the graph while the rule remains fixed (i.e. the majority rule). For network-based CAs, algorithms are designed to search for the graph topology that triggers the best neighborhood to be used in connection with a fixed rule for the DCT. Another topology proposed in [1] is a hybrid one in which the state change of a cell is allowed to be influenced by long-distance cells. Besides the cell itself and the cells in the local neighborhood, fixed distant cells contribute to the way in which the cell state is changed over time. This hybrid neighborhood has a fixed structure combining local and, to a certain extent, global information.

In this paper, we propose a new weighted rule that can be applied for the hybrid topology mentioned above. In the proposed approach, the local neighbors and the long-distance neighbors have different weights when computing the next state of a current cell. An evolutionary algorithm is developed to search for the best performing weighted rule for DCT based on the hybrid neighborhood structure. Computational experiments indicate a better performance obtained by the proposed weighted rule compared to the standard majority rule for hybrid CA topology and neighborhood.

The rest of the paper is structured as follows: section 2 briefly presents the density classification problem, section 3 describes the relevant lattice, network-based and hybrid CA topologies, section 4 presents the proposed weighted majority rule, section 5 includes the computational experiments and results, and section 6 contains the conclusions of the paper and directions for future work.

2. The Density Classification Problem

DCT is a challenging problem extensively studied due to its simple description and potential to generate a variety of complex behaviors.

The aim of DCT is to find a binary one-dimensional CA able to classify the density of 1s (denoted by ρ_0) in the initial configuration. If $\rho_0 > 0.5$ (1 is dominant in the initial configuration) then the CA must reach a fixed point configuration of 1s otherwise it must reach a fixed-point configuration of 0s within a certain number of timesteps. Most studies consider the CA length of $N = 149$ and the CA radius of $r = 3$ (which gives a neighborhood size of 7).

The CA lattice starts with a given binary string called the initial configuration (IC). After a maximum number of iterations (usually set as twice the size of CA), the CA will reach a certain configuration. If this is formed of homogeneous states of all 1s or 0s, it means that the IC has been classified as density class 1, respectively 0. Otherwise, the CA makes by definition a misclassification [20]. It has been shown that there is no rule that can correctly classify all possible ICs [13].

The performance of a rule measures the classification accuracy of a CA based on the fraction of correct classifications over ICs selected from an unbiased distribution (ρ_0 is centered around 0.5).

3. CA Topologies and Results for DCT

This section describes some relevant CA topologies, i.e. lattice, network and hybrid, and presents related existing rule detection methods and their results for DCT.

3.1. **Regular Lattice Topology.** A one-dimensional lattice of N two-state cells is used to represent the CA. The state of each cell changes according to a function depending on the current states in the neighborhood. The neighborhood of a cell is given by the cell itself and its r neighbors on both sides of the cell, where r represents the radius of the CA. The initial configuration of cell states (0s and 1s) for the lattice evolves in discrete time steps updating cells simultaneously according to the CA rule.

The regular lattice topology is engaged in most studies tackling DCT for 1D binary-state CA. Packard [19] developed the first evolutionary approach to detect rules for one dimensional binary state CA with a radius of 3. The potential of genetic algorithms for computational emergence in the density classification task has been extensively investigated by Mitchell et al [6, 3, 15, 9, 16, 20, 14]. The evolutionary approach to DCT proposed by Mitchell et al sets the fitness of a rule table as the fraction of correct classifications made over 100 randomly chosen initial configurations with uniformly

distributed density. The three main strategies discovered are default, block-expanding (a refinement of the default strategy) and particle revealing increasing computational complexity [3]. The best particle rule found by Mitchell et al [15] has a performance of 0.769.

Juille and Polack [11] used coevolutionary learning for the density classification task reporting good results for this problem (performance of 0.86). Oliveira et al [17] present a multiobjective evolutionary approach to DCT based on the non-dominated sorting genetic algorithm (NSGA). The algorithm is implemented in a distributed cooperative computational environment and is able to discover new rules with a performance of 0.8616. Wolz and Oliveira [26] proposed a two-tier evolutionary framework able to detect rules with a performance of 0.889. The approach integrates a basic evolutionary algorithm in a parallel multi-population search algorithm. To the best of our knowledge, the DCT rules presented in [26] are the ones with best performance discovered to date [18].

3.2. **Network Topology.** Some studies [23, 24, 4, 5, 22] consider a network-based topology for the CA where the cells can be connected in any way while the rule is the same for all cells.

Watts [24] studied the small-world graph version of the DCT: the rule is fixed and the performance of different small-world networks for DCT is evaluated. A small-world graph is constructed starting from a regular ring of nodes in which each node has k neighbors followed by a random rewiring procedure [23, 24]. The rule used for small-world network DCT is simple: at each time step, each node takes the state of the majority of its neighbor nodes in the graph (if the number of state 1s equals the number of state 0s in the neighbors list then the node is randomly assigned a state with equal probability between 0 and 1). Small-world networks have a performance of around 0.8 for the DCT with this fixed majority rule for 149 cells CA.

Tomassini et al [4, 5, 22] investigated network-based CAs for the density and synchronization problems. Spatially structured evolutionary algorithms are used to find the best performant network topology for DCT when the rule is fixed to the majority rule. An individual represents a network structure and the fitness is computed based on the fraction of ICs (out of 100 ICs generated anew for each individual) correctly classified by the majority rule based on the neighborhood given by the network. The best evolved network starting from initial regular rings has a performance of 0.823 (for 149 cells) while the result for random graphs as initial population is similar (performance of 0.821 of the best network). Similar results have been obtained by [8] with a simple evolutionary algorithm able to produce network topologies with high performance for DCT based on the majority rule.

3.3. **Hybrid Topology.** A hybrid topology has been proposed in [1] where a mixed induced neighborhood keeps invariable the number of neighbors. This topology works on a lattice based on the CA radius r and a parameter n referring to the number of long-distance cells allowed to contribute to the current hybrid neighborhood. In order to create the new topology of radius r, we start with a regular ring lattice of radius $r-n$. The other $2*n$ nodes that are part of the hybrid neighborhood of a node i are long distance neighbors randomly chosen from the rest of the nodes, but following some rules that ensure the equilibrium of the neighborhood. This means that i always remains the central node of the neighborhood and the distance between node i and the long distance neighbors places half of them (n nodes) on the left hand side and the other half (n nodes) on the right hand side of i. Using this approach, a new topology which resembles a network topology but it is still very close to a regular ring lattice is obtained. This allows us to consider the same majority understanding as in the classical case of regular ring lattice topology. A detailed description of the proposed hybrid topology can be found in [1].

The potential of this topology with the corresponding notion of hybrid neighborhood is shown by the significantly better performance obtained in fewer CA iterations, compared to the classical approach that uses the regular ring lattice as topology. We should also note that the best performance obtained with the hybrid topology (0.79) is considerably higher than the performance obtained for regular lattice topology (0.64) and is very close to the performance obtained by CA based on small-world topologies (0.82).

4. New Weighted Rule for Cellular Automata with Hybrid Neighborhood

As already mentioned, the state of each cell in CA changes according to a function depending on the current states in the neighborhood. The neighborhood of a cell is given by the cell itself and its r neighbors on both sides of the cell. The initial configuration of cell states (0s and 1s) for the lattice evolves in discrete time steps updating cells simultaneously according to the CA rule.

In current existing approaches, each neighbor (including the cell itself) has the same vote weight when deciding which is the next state of the current cell. Indeed, whether we consider the regular lattice ring or the network topology for CAs, we can not differentiate between neighbors - the vote of each neighbor has the same weight. The hybrid topology described in the previous section allows us to introduce a new concept of majority rule, where different neighbors have different vote weights when deciding the next state of the current cell.

The hybrid topology involves the presence of two kinds of neighbors: local and far neighbors. The proposed rule gives different vote weights to local

neighbors and to far neighbors. Let us denote by w_l the weight assigned to local neighbors and by w_f the weight assigned to far neighbors. We identify two different scenarios that can be applied in this approach:

$$(i) 0 \leq w_l \leq 1, w_f = 1 - w_l$$
$$(ii) 0 \leq w_l \leq 1, 0 \leq w_f \leq 1$$

In the first scenario, the increase of one weight leads to the decrease of the other one, while in the second scenario, there is no relation between the weights of the two kinds of neighbors. In both scenarios, we compute the maximum weighted sum that can be obtained when each neighbor has the value 1. Let us denote by w_{max} the obtained value:

$$w_{max} = n_l * w_l + n_f * w_f,$$

where n_l represents the number of local neighbors and n_f represents the number of far neighbors.

Let s_i^t denote the state of cell i at timestep t. Let i_j, where $j = 1, 2r + 1$, represent the neighbors of cell i. The number of neighbors of a cell is $2 * r + 1$, out of which $n_l = 2*(r-n)$ are local neighbors and $n_f = 2*n$ are far neighbors, according to the hybrid topology.

For each cell i, the weighted sum of the neighbors states at timestep t, denoted by σ_i^t, is defined as follows:

$$\sigma_i^t = \sum_{j=1}^{2r+1} s_{i_j}^t * w_{i_j}$$

As indicated above, $s_{i_j}^t$ represents the state of cell i_j in the hybrid neighborhood at timestep t and w_{i_j} is the weight of neighbor i_j:

$$w_{i_j} = \begin{cases} w_l, & \text{if } i_j \text{ is a local neighbor} \\ w_f, & \text{if } i_j \text{ is a far neighbor} \end{cases}$$

The next state of a cell i is 1 if the weighted sum σ_i exceeds half of the maximum weighted sum:

$$s_i^{t+1} = \begin{cases} 1, & \text{if } \sigma_i^t \geq w_{max}/2 \\ 0, & \text{otherwise} \end{cases}$$

The proposed weighted rule is a very realistic model with many applications in real-world problems where different entities have different voting weights when deciding a next state in a system that could be modeled as a CA.

5. Experimental Results

Computational experiments consider a well studied version of the DCT problem: the one-dimensional binary-state CA of size 149 based on the radius of 3. For the hybrid topology we set $r = 3$ and $n = 1$ (i.e. 2 local neighbors on each side and 1 distant neighbor). The neighborhood size is 7 cells which leads to a rule size of $2^7 = 128$.

A simple evolutionary algorithm has been developed to detect rules for the DCT. A potential solution of the problem is encoded as a one-dimensional array of weights (one weight for the first proposed scenario and two weights for the second one). The initial population is randomly generated. The potential solutions are evaluated by means of a real-valued fitness function $f : X \to [0, 1]$, where X denotes the search space of the problem. The fitness function represents the fraction of correct classification over 100 randomly generated initial configurations. A relative fitness is used, as the set of initial configurations is generated anew for each generation of the algorithm. This way, solutions with high fitness in one generation and which survive in the next generation will be evaluated again using another set of 100 initial configurations. While the fitness is evaluated by using 100 uniformly distributed initial configurations, the performance of a rule is computed as the fraction of correct classifications for 10^4 randomly generated initial configurations. The initial configurations are generated in such a way that each cell has the same probability $\frac{1}{2}$ of being 0 or 1. The CA is iterated until it reaches a fixed-point configuration of 1s or 0s but for no more than $M \approx 2N$ time steps.

The individual resulted after each recombination will be mutated at exactly two randomly chosen positions. A weak mutation is considered, the probability of obtaining a different value for the chosen position being equal to the probability of obtaining the very same value. The algorithm is applied for 100 generations with a population size of 100, roulette selection, one point crossover with probability of 0.8, weak mutation with probability 0.2 and elite size of 10%.

TABLE 1. Performance obtained in 10 runs for $HybCA$, W_1HybCA and W_2HybCA

	$HybCA$	W_1HybCA	W_2HybCA
Best	0.7902	0.8202	0.8083
Average	0.74847	0.79475	0.76573
Std Dev	0.034297815	0.017707202	0.0374052
T-test p-value		0.001870	0.364680

The performance obtained with the new weighted majority rule (both scenarios) is compared to the performance obtained when using the classical rule table. It should be noted that the same evolutionary algorithm with the same parameters is used to evolve both standard and weighted rules in order to allow a direct comparison of results. Both standard and weighted rules use the same hybrid CA topology and neighborhood. The case of standard rules can be viewed as a simplification of weighted rules evolution where no weights need to be determined.

Table 1 presents the results obtained after 10 runs of the algorithm to detect a classical rule ($HybCA$) and for evolving a weighted rule (W_1HybCA for first scenario and W_2HybCA for the second scenario considered). The weighted rule triggers an improvement of performance from 0.7902 ($HybCA$) to 0.8202 for W_1HybCA and 0.8083 for W_2HybCA. The results are compared using the paired t-test with a 95% confidence interval. For the first scenario, the p-value is significantly smaller than 0.05, which indicates that the mean performance obtained when using the proposed weighted rule is notably better than that obtained when using a classical rule. The mean performance obtained based on the second scenario is higher compared to the classical rule but not significantly (according also to the p-value which is higher than 0.05 as shown in Table 1). The best performance obtained in 10 runs of the evolutionary algorithm is indeed more significantly improved when using the weighted rule in the first scenario (the evolved weight for local neighbors was 0.16 when the best performance was obtained). Therefore, we further consider this first scenario to perform a further analysis and report results.

A comparison that shows the efficiency of the proposed weighted rule is the number of iterations needed to correctly classify the initial configurations (for example, for a performance of 0.82, there are 8200 correctly classified initial

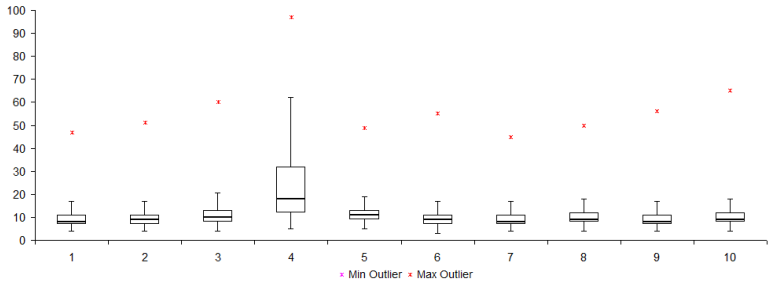


FIGURE 1. Number of iterations needed to correctly classify the initial configurations for HybCA, for each of 10 runs

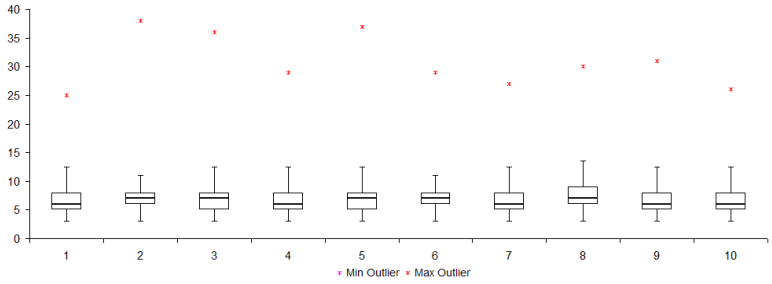


FIGURE 2. Number of iterations needed to correctly classify the initial configurations for W_1HybCA, for each of 10 runs

configurations and therefore 8200 observations of how many iterations were needed for those correctly classified initial configurations). Obtained results for 10 different runs are depicted in Figures 1 and 2. It can be easily observed that in the case of W_1HybCA the number of needed iterations is slightly smaller compared to $HybCA$. This means that the convergence is accelerated when using the proposed rule.

The obtained best rule is further tested against dynamic changes in order to evaluate the robustness of the hybrid topology when using weighted rules. Dynamic changes are understood as replacements of long distance neighbors with other randomly generated long distance neighbors. We apply 1000 random replacements for each neighborhood obtained by the 10 runs. The performance of the resulting CA is evaluated after each change and obtained results for all 10 runs are depicted in Figure 3.

As shown, there are no significant variations of the performance when the nodes neighborhoods are subject to dynamic changes. This is a good indicator of the topology robustness.

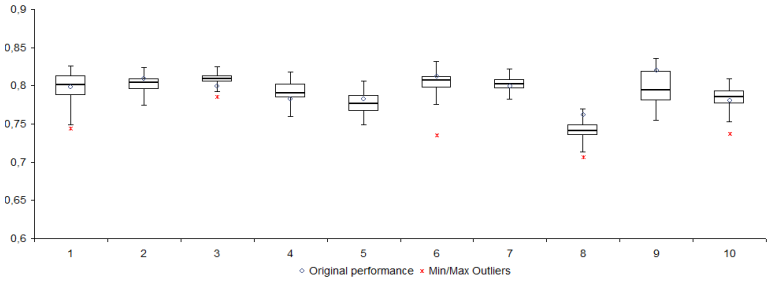


FIGURE 3. Performance obtained in 1000 dynamic steps for 10 runs of the algorithm

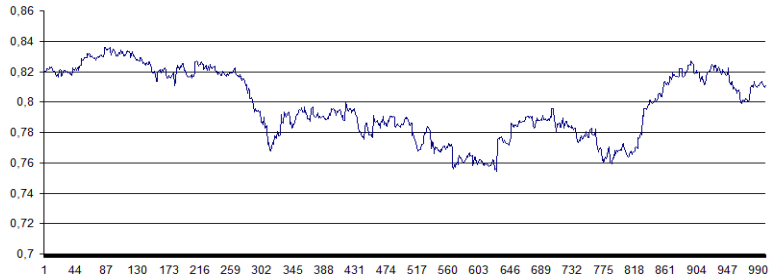


FIGURE 4. Performance obtained in 1000 dynamic steps for one run of the algorithm

Figure 4 presents the results obtained by a typical run of the algorithm (from the considered 10 runs). The original performance of the obtained rule before the topology being subject to dynamic changes was 0.8202. It can be noticed that the performance does not go below 0.7545, we even obtain better performances (the best one is 0.8362) and an average performance of 0.7977.

6. Conclusions and Future Work

A new weighted rule for a hybrid CA topology and neighborhood has been proposed. The cells in the hybrid neighborhood are allowed to have different weights (according to their local or far association to the current cell) in the decision of changing the current cell state. This approach enables a weighted rule application to the hybrid CA topology which has been investigated for the DCT in one-dimensional binary state CAs. Computational experiments emphasize that the weighted rules detected by an evolutionary algorithm have a good performance shown to improve that of standard rules

evolved for the same CA topology. The CA performance remains stable when dynamic changes are introduced in the neighborhood structure.

Future work focuses on investigating weighted schemes in connection with other CA rules and topologies, particularly for network-based CAs where weights associated with edges connecting two cells can be easily used in a weighted majority rule for graph topologies.

Acknowledgment

This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

[1] Andreica, A., Chira, C., Using a Hybrid Cellular Automata Topology and Neighborhood in Rule Discovery, submitted to IEEE Congress on Evolutionary Computation (CEC 2013).

[2] Chira, C., Gog, A., Lung, R. I., Iclanzan, D., Complex Systems and Cellular Automata Models in the Study of Complexity, Studia Informatica series, Vol. LV, No. 4, pp. 33-49 (2010)

[3] Crutchfield, J.P., Mitchell, M., The evolution of emergent computation, Proceedings of the National Academy of Sciences, USA 92 (23), (1995), pp.10742-10746.

[4] Darabos, C., Giacobini, M., Tomassini, M., Performance and Robustness of Cellular Automata Computation on Irregular Networks. Advances in Complex Systems 10: 85-110 (2007)

[5] Darabos, C., Tomassini, M., Di Cunto, F., Provero, P., Moore, J.H., Giacobini, M., Toward robust network based complex systems: from evolutionary cellular automata to biological models, Intelligenza Artificiale 5(1): 37-47 (2011)

[6] Das, R., Mitchell, M., Crutchfield, J.P., A genetic algorithm discovers particle-based computation in cellular automata, Parallel Problem Solving from Nature Conference (PPSN-III), Springer-Verlag (1994) pp.344-353.

[7] Ferreira, C., Gene Expression Programming: A New Adaptive Algorithm for Solving Problems, Complex Systems, Vol. 13, No. 2 (2001), pp.87-129.

[8] Gog, A., Chira, C., Dynamics of Networks Evolved for Cellular Automata Computation, Proceedings of the 7th International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2012), Salamanca, Spain, Hybrid Artificial Intelligent Systems, Springer-Verlag, vol. 7208-7209 (2012) pp. 359-368.

[9] Hordijk, W., Crutchfield, J.P., Mitchell, M., Mechanisms of Emergent Computation in Cellular Automata, Parallel Problem Solving from Nature-V, Springer-Verlag (1998) pp.613-622.

[10] Juille, H., Pollack, J.B.: Coevolving the 'ideal' trainer: Application to the discovery of cellular automata rules. Genetic Programming 1998: Proceedings of the Third Annual Conference (1998)

[11] Juille, H., Pollack, J.B., Coevolutionary learning and the design of complex systems, Advances in Complex Systems, Vol. 2, No. 4 (2000), pp. 371-394.

[12] Koza, J.R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge (1992).
[13] Land, M., Belew, R. K., No perfect two-state cellular automata for density classification exists, Physical Review Letters, 74:25 (1995), pp. 5148-5150.
[14] Marques-Pita, M., Mitchell, M., Rocha, L., The role of conceptual structure in designing cellular automata to perform collective computation, Proceedings of the Conference on Unconventional Computation, UC 2008, Springer (Lecture Notes in Computer Science), 2008.
[15] Mitchell, M., Crutchfield, J.P., Das, R., Evolving cellular automata with genetic algorithms: A review of recent work, In Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96). Russian Academy of Sciences (1996)
[16] Mitchell, M., Thomure, M. D., Williams, N. L.: The role of space in the Success of Coevolutionary Learning. Proceedings of ALIFE X - The Tenth International Conference on the Simulation and Synthesis of Living Systems (2006)
[17] de Oliveira, P.P.B., Bortot, J.C., Oliveira, G., The best currently known class of dynamically equivalent cellular automata rules for density classification, Neurocomputing, 70:1-3 (2006), pp. 35-43.
[18] Oliveira, G.M.B., Martins, L.G.A., de Carvalho, L.B., Fynn, E., Some investigations about synchronization and density classification tasks in one-dimensional and two-dimensional cellular automata rule spaces, Electron. Notes Theor. Comput. Sci., 252 (2009), pp. 121-142.
[19] Packard, N.H., Adaptation toward the edge of chaos, Dynamic Patterns in Complex Systems, World Scientific (1988), pp. 293-301.
[20] Pagie, L., Mitchell, M.: A comparison of evolutionary and coevolutionary search. Int. J. Comput. Intell. Appl., 2, 1, 53-69 (2002)
[21] Tomassini, M., Venzi, M.: Evolution of Asynchronous Cellular Automata for the Density Task. Parallel Problem Solving from Nature - PPSN VII, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2439, 934-943 (2002)
[22] Tomassini, M., Giacobini, M., Darabos, C., Evolution and dynamics of small-world cellular automata, Complex Systems 15 (2005), pp. 261-284.
[23] Watts, D.J., Strogatz, S.H., Collective dynamics of 'smallworld' networks, Nature 393 (1998), pp. 440-442.
[24] Watts, D.J., Small Worlds: The Dynamics of Networks Between Order and Randomness, Princeton University Press, Princeton, NJ (1999)
[25] Wolfram, S., Theory and Applications of Cellular Automata, Advanced series on complex systems, World Scientific Publishing, 9128 (1986)
[26] Wolz, D., de Oliveira, P.P.B., Very effective evolutionary techniques for searching cellular automata rule spaces, Journal of Cellular Automata 3 (2008), pp. 289-312.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania
E-mail address: [anca,cchira]@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

PEDESTRIAN RECOGNITION BY USING KERNEL DESCRIPTORS

ANCA ANDREICA, LAURA DIOŞAN, RADU GĂCEANU, AND ADELA SÎRBU[(1)]

Abstract. Recognition of people in images is important for many applications in computer vision. This paper presents an experimental study on pedestrian classification. We investigate the recently developed kernel-based features in order to represent an image and two learning algorithms: the popular Support Vector Machine (SVM) and Genetic Programming (GP). Numerical experiments are performed on a benchmark dataset consisting of pedestrian and non-pedestrian (labeled) images captured in outdoor urban environments and indicate that the evolutionary classifier is able to perform better over SVM.

1. Introduction

Pedestrian safety is an important problem of global dimensions. A World Health Organization 2010 report describes traffic accidents as one of the major cause of death and injuries around the world, accounting for an estimated 1.2 million fatalities and 50 million injuries. Enhancing comfort and safety for the driver and the occupants of an automobile has been a major motivator in innovations associated with Intelligent Vehicles and Intelligent Transportation Systems. The European Union has been conducting several projects in collaboration with auto industry and research institutes for intelligent vehicle systems in general and pedestrian safety in particular.

One approach for pedestrian safety improvement is to develop performant recognition systems. In this paper we propose an evolutionary-based model for the learning phase of such a system. The input data for this classifier is

Received by the editors: March 25, 2013.

2010 *Mathematics Subject Classification.* 68T05,91E45.

1998 *CR Categories and Descriptors.* code I.2.6 [**Learning**]: – *Concept learning*.

Key words and phrases. Object recognition, Kernel descriptors, Support Vector Machines, Kernel selection.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

represented by features extracted from images by using different kernel descriptors.

Pedestrian detection has been an intensively studied problem, especially in the last years. There is a variety of classifiers in literature trained on different combinations of features like HOG [7], SURF [2], CSS [23], Haar walvet representation [22] and tested on datasets like INRIA, Daimler or Caltech. In [23] is performed a comparison of the existing datasets and an evaluation of several classifiers (AdaBoost, linear SVM, HIK SVM and latent SVM) based on HOG and Haar walvet representation. One of the most efficient feature extraction methods is represented by kernel descriptors [4]. In addition, these features have not been used in the case of pedestrian recognition yet, to the best of our knowledge. We, therefore, investigate their usage for this problem.

For solving the classification task, there are several approaches like Boosting [17] and statistical classification: Artificial Neural Networks (ANN) [24] and Support Vector Machines (SVM) [1]. It is known that the SVM performs better as it can avoid over-learning problem that appears in ANN training and a deficiency of a slow training speed that appears in Adaboost.

In our approach we involve a GP-based learning, as evolutionary methods are able to identify a large spectrum of decision functions (linear or nonlinear). Furthermore, there are no evolutionary models for solving the pedestrian recognition problem, as far as we know.

The paper is organised as follows: Section 2 reviews the most efficient image representations and Section 3 gives a brief description of two learning algorithms (SVM and GP). The proposed framework is presented in Section 4 and followed by numerical experiments (detailed in Section 5). Finally, conclusions and ideas for future work can be found in Section 6.

2. Image representation

A highly challenging problem in computer vision is how to extract relevant image features. The features that characterise an image can be classified from many points of view. An important criterion is the area from that the feature is extracted: if the entire image is used, then some global features are computed, while if one or more image regions (patches) are utilised, then local features are determined. Another important criterion is the complexity of extraction. Image features can be extracted from scratch that means the features are extracted directly from the image (and in this case we discuss about low-level features) or can be computed based on some previously extracted features (in this case high-level features are obtained). The most popular (due to their success) low-level image descriptors are orientation and gradient histograms,

while one of the best high-level descriptor is the kernel view of orientation histograms [4]. More details about these features are given in the following.

2.1. **Histogram of Oriented Gradient (HOG) descriptors.** The first step in human detection in images is represented by feature extraction. For this stage in the recognition process, Dalal et al [7] proposed the *Histogram of Oriented Gradient (HOG)* descriptors, which show superior performance compaired to previously existing approaches (eg. SIFT [14], SURF [2], etc). HOG descriptors are briefly described in what follows.

Firstly, there is a normalization stage intended to reduce the influence of illumination effects. Power low compression is used on this purpose. There is a second stage for computing first order image gradients for detecting contour, silhouette and some texture information. The image is then divided into small regions called *cells*, for each of them accumulating a local 1-D histogram of gradient or edge orientations over all the pixels. The combined histograms give the image representation. A contrast-normalization follows in order to induce a better invariance to illumination, shadowing and edge contrast. Local groups of cells form *blocks* that might overlap and the normalised block descriptors are referred to as HOG descriptors. The descriptors from all blocks are then collected into a combined feature vector.

Static version of HOG descriptors are obtained by following the stages described above. The authors also proposed motion HOG descriptors which use oriented histograms of differential optical flow. The gradient computation is therefore replaced by flow computation and differential flow estimation [8].

2.2. **Kernel Descriptors (KD).** Kernel descriptors [4] can be seen as a generalization of orientation histograms (including HOG), which are a particular type of match kernels over *patches* (viewed as a collection of *blocks*). Moreover, kernel descriptors overcome some disadvantages of histograms based techniques, where similarity between different regions of images is computed based on their histogram. By using this approach, some quantization errors might be introduced, as individual pixel attribute values are discretized into bins and then a histogram is computed over the discrete attribute values within a patch.

Some of the kernel descriptors are the gradient match kernel (able to capture image variations) – based on a kernel of magnitudes, an orientation kernel and a position kernel –, the colour kernel (able to describe image appearance) – based on a colour kernel and a position kernel – and the local binary pattern kernel (able to capture local shape more effectively) – based on a kernel of standard deviations of neighbour pixels, a kernel of binarized pixel value differences in a local window and a position kernel.

The advantage of kernel descriptors is that they do not discretize pixel attribute values, being able to convert pixel attributes into rich patch-level features. The similarity between different images regions is therefore computed based on a match kernel function. For computation efficiency reasons, approximate, low dimensional match kernels are computed.

Kernel descriptors can be applied not only over sets of pixels, but over sets of kernel descriptors as well. In this hierarchical approach [3], kernel descriptors are recursively applied until the image features are obtained.

3. Learning algorithm

Another aspect that must be considered when the problem of object recognition has to be solved is the classification algorithm. Since the classification must be performed in an automatically manner, a machine learning algorithm can be utilised. The general problem of machine learning is to search a, usually very large, space of potential hypotheses to determine the one that will best fit the data and any prior knowledge. In supervised image classification, we are given a training set of images and their corresponding labels. The goal is to learn (based on the training set) a classifier to label unseen images. Two of the most performant algorithms are SVM and GP-based classifiers and, in what follows, we will briefly describe them.

3.1. **SVM.** SVMs are a group of supervised learning methods that can be applied to classification or regression. They use a technique known as the "kernel trick" to apply linear classification techniques to non-linear classification problems. Using a Kernel function [20], the data points from the input space are mapped into a higher dimensional space. Constructing (via the Kernel function) a separating hyperplane with maximum margin in the higher dimensional space yields a non-linear decision boundary in the input space separating the tuples of one class from another.

Suppose the training data has the following form: $D = (x_i, y_i)_{i=\overline{1,m}}$, where $x_i \in \Re^d$ represents an input vector and each y_i, $y_i \in \{-1, +1\}$, the output label associated to the item x_i. SVM algorithm maps the input vectors to a higher dimensional space where a maximal separating hyper-plane is constructed [20]. In order to construct a maximal margin classifier one has to solve the following convex quadratic programming problem, which is the primal formulation of it:

$$\begin{array}{ll} \text{minimise}_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i & \\ \text{subject to:} & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \forall i \in \{1, 2, \ldots, m\}. \end{array} \tag{1}$$

The coefficient C (usually called *penalty error* or *regularization parameter*) is a tuning parameter that controls the trade off between maximizing the margin and classifying without error. Larger values of C might lead to linear functions with smaller margin, allowing to classify more examples correctly with strong confidence. A proper choice of this parameter is crucial for SVM to achieve good classification performance.

The original optimal hyper-plane algorithm proposed by Vapnik in 1963 was a linear classifier [20]. However, in 1992, Boser, Guyon and Vapnik [5] have suggested a way to create non-linear classifiers by applying the *kernel trick*. Kernel methods work by mapping the data items into a high-dimensional vector space $\mathcal{F}$, called feature space, where the separating hyper-plane has to be found [5]. This mapping is implicitly defined by specifying an inner product for the feature space via a positive semi-definite kernel function: $K(x, z) = \phi(x)^T \phi(z)$, where $\phi(x)$ and $\phi(z)$ are the transformed data items x and z [19]. Note that all we required is the result of such an inner product. Therefore we neither need to have an explicit representation of the mapping ϕ, nor do we need to know the nature of the feature space. The only requirement is to be able to evaluate the kernel function on all the pairs of data items, which is much easier than computing the coordinates of those items in the feature space.

There is a wide choice for a positive definite and symmetric kernel K. The selection of a kernel has to be guided by the problem that must be solved. Choosing a suitable kernel function for SVMs is a very important step for the learning process.

While SVM classifiers intrinsically account for a trade off between model complexity and classification accuracy [21], the generalization performance is still highly dependent on appropriate selection of the penalty error C and kernel parameters.

3.2. **GP.** Genetic programming (GP) is a form of evolutionary computation in which the individuals in the population are computer programs, instead of bit strings [12]. GP starts from a high-level statement of what needs to be done and creates a computer program to solve the problem [12]. While genetic algorithms (GA) want to evolve only solutions for particular problems, GP evolves complex computer programs. GP individuals are represented and manipulated as nonlinear entities, usually trees in the standard approach. Any expression can be represented as a tree of functions and terminals. Depending on the problem to solve, a GP chromosome can be the syntax of an arithmetic expression, formulas in first order predicate logic or code written in a programming language [11].

The skeleton of a GP algorithm is similar to GA. It uses a population of individuals on which the standard evolutionary operators are applied. Initially a population of hierarchically arranged programs is generated by randomly combining functions and terminals. A fitness function may be applied on these programs and the best individuals are selected for reproduction. Crossover is achieved by swapping subtrees among the parents starting at two randomly selected nodes. Mutation may occur by replacing the subtree starting at a randomly selected node in the chromosome by a randomly generated subtree [12].

Flexibility is one of the main advantages of GP, and this feature allows GP to be applied for classification in many different ways. The nature of GP individuals, which include terminals (variables and constants) and nonterminals (operators and functions), gives them the ability not only to represent knowledge but also to perform computations, so that GP can be used in almost any classification-related task, not only in the core task of classifier induction, but also for preprocessing and post-processing purposes [10]. GP can be easily applied to decision tree evolution considering that chromosomes are usually represented as trees and thus they could each be regarded as decision tree classifiers. Along the same line, GP individuals can encode classification rules. A classification rule consists of two parts: the antecedent and the consequent. The antecedent states a condition over the data while the consequent contains the class label. The conditional part is usually composed of relational operators applied on various attribute data, i.e., an expression which can be represented as a tree of functions and terminals.

4. Proposed framework

Our aim is to investigate how two different learning algorithms can solve an important classification problem: the pedestrian recognition. Since the performance of a classification algorithm is strongly influenced by two ingredients (first, a suitable representation of the objects to be categorized and second, a powerful decision maker algorithm on top of this representation), we have decided to extract relevant features from images through kernel descriptors that must be analysed and processed by two classification methods: SVM and GP.

4.1. **Arguments.** *Why Kernels?* Because, by definition, they catch the similarity between arbitrary inputs, being able, in the same time, to integrate invariance (that is present in the case of image processing), to capture dependencies and to perform an efficient computation and storage.

Why SVM? It is well known that linear SVMs are currently the most frequently used classifiers in Computer Vision [13] since its training time is

approximately linear in data dimensionality and, also, approximately linear in number of training examples. Furthermore, the evaluation time (per test example) is linear in data dimensionality and is independent of number of training examples. According to [13] SVMs with non-linear kernel are commonly used for small to medium sized Computer Vision problems because their training time is typically cubic in number of training examples, while their evaluation time is typically linear in number of training examples. Furthermore, the classification accuracy is typically higher compared to linear SVMs. Shortly, linear SVMs are very fast in training and evaluation, while non-linear kernel SVMs give better results, but do not scale well (with respect to number of training examples).

Why GP? There are several arguments that sustain the utility of GP-based methods in solving classification problems, in general, and object recognition, in particular. Firstly, a GP-based method is able to perform an implicit and automatic transformation of data (the original features can be pre-processed by different methods: selection of a subset of original attributes, weighting the original attributes, construction of new features as functional expressions involving the original attributes). The feature selection is part of the evolutionary process of GP that involves individuals encoded as variable-length chromosomes, while the feature construction benefits of the GP individual's ability to combine the attributes trough different operations. Secondly, GP is able to extract various models (like classification rules or discriminant functions) from data. Furthermore, due to its capability to evolve complex discriminant functions, GP is able to solve both linear classification problems and non-linear classification problems without apriori specifying the problem type (linear or non-linear). When a linear classifier can not solve the problem, two solutions can be considered:

- a combination of several linear classifiers (as in the case of ANN or Boosting which actually encode decision functions which depend non-linearly on input data) or
- a data pre-processing step, when the original input data is transformed form the original space of representation (non-linear) into a new space that is, in general, a higher dimensional one and where the data becomes linearly separable. Usually this step is performed through the kernel functions (as in the case of SVMs). For instance a set of points can be non-linearly separable in Cartesian coordinates, but linearly separable in Polar coordinates.

Unlike other Machine Learning (ML) algorithms, GP automatically combines these two solutions during the evolution process, its individuals being able

to automatically encode both type of classifiers (linear and non-linear). Related to how GP can solve Computer Vision tasks, the discriminant functions evolved by the GP algorithm are very akin to the kind of mathematical operations and transformations usually applied to image processing. GP is flexible. It is well known that GP individuals are able to represent a great variety of learning formalisms (eg. decision trees, classification rules, discriminant functions), but also learning mechanisms (like those involved in ANNs or SVMs). Flexibility also concerns the adaptability of GP techniques to various tasks through its elements (fitness function, genetic operation, evolving mechanisms). GP ensures interpretability of the evolved classifiers since the size of GP individuals influences the comprehensibility of the model; the bigger the classifier, the harder to interpret for humans. GP is able to ensure a competitive performance.

Therefore, we study a GP-based classifier that is able to solve the given problem, obtaining improvements that concern several aspects:

- performance of the classification – a better classifier in terms of accuracy
- human-independent models – GP individuals are able to automatically decide in two very important aspects: knowledge and model representation. Instead of performing a distinct pre-processing step, the GP method is able to automatically and simultaneously select the most relevant features and construct a relevant model.
- less complex models – GP is able to automatically apply the principles of Occam's razor: if two models have the same performance (in our case if two decision algorithms ensure the same classification accuracy) the less complex model should be preferred.

Taking into account all these aspects, we have used in our numerical experiments the following methodology: several features are extracted directly from images through kernel descriptors and, afterwards, two algorithms (an SVM and a GP-based classifier) are considered in order to learn the decision model. In both cases, the learning takes place in a cross-validation framework. In k-fold cross-validation, the training data is randomly split into k mutually exclusive subsets (or folds) of approximately equal size. The decision is obtained by using $k-1$ subsets on training data and then tested on the subset left out. This procedure is repeated k times and in this manner each subset is used for testing once. Averaging the test error over the k trials gives a better estimate of the expected generalization error.

4.2. **Feature extraction by using kernel descriptors.** We considered the framework proposed by L. Bo [4] for image classification and we test different

kernel functions. We have already established that the selection of the kernel function is very important for kernel descriptor [9].

At the first step, based on the available code of Kernel descriptors developed by Xiaofeng Ren (*http : //www.cs.washington.edu/ai/Mobile$_R$obotics/projects /kdes/*), we have tested different kernels when extracting local features from an image. Because we work only with gray images, we investigate only the kernel descriptors able to capture image variations (gradient match kernels [4]). As presented in Section 2, the Bo's gradient match kernel is composed of three kernels: a kernel of magnitudes, an orientation kernel and a position kernel.

The magnitude kernel is a linear one and its role is to measure the similarity of gradient magnitudes of two pixels. The magnitude kernel type cannot be changed since it must be an equivalent of the histogram of gradients in the feature map (a pixel has an associated feature vector obtained by multiplying the magnitude and the orientation of the pixel over all considered orientation bins).

The other two kernels involved in Ren's computation of the gradient match kernel, the orientation kernel (for computing the similarity of gradient orientations) and the position kernel (for measuring the spatial closeness of two pixels), are actually Gaussian kernels. Therefore, we have changed the implementation and we have involved more possible orientation and position kernels (Exponential, Laplacian, Euclidean) in the feature extraction process.

4.3. **SVM.** Regarding the SVM, we have considered the LibSVM tool [6] since it shines above the other tools in terms of ease of use, choice of options and features.

The dual version of the optimisation problem which arises during the SVM training was solved by Sequential Minimal Optimization (SMO) algorithm [18], since it is able to quickly solve the quadratic programming optimisation problem of SVM. We have chosen this formulation of SVM since the duality theory provides a convenient way to deal with the constraints and, in this form, the optimisation problem can be solved in terms of dot products, that allows using the kernel trick. Furthermore, SMO requires an amount of memory that increases only linearly with the training set size, being able to handle very large training sets - as in the image classification case. These aspects are different from Bo's framework [4] that is based on primal formulation of SVM and on conjugate gradient optimisation methods (in fact, Newton optimisation).

We have tried to use several kernels with different parameters during the SVM learning process in order to identify the best one: the Linear kernel, the Polynomial kernel, the Gaussian kernel and the Normalised Polynomial kernel. For the Polynomial kernel several exponents have been tested (2, 3),

for parameter $\frac{1}{2\sigma^2}$ of Gaussian kernel the following values have been checked: 0.1, 0.01, 0.001, 0.0001 and for Normalised Polynomial kernel the exponent was 2.

4.4. **GP.** For the evolutionary classifier a linear and efficient GP version is actually utilized: Multi Expression Programming (MEP)[16]. MEP uses a linear representation of chromosomes and a mechanism to select the best gene for providing the output of the chromosome. This is different from other GP techniques which use a fixed gene for output. Furthermore, no extra processing for repairing newly obtained individuals is needed.

The dynamic-output chromosome has some advantages over the fixed-output chromosome especially when the complexity of the target expression is not known. Variable-length expressions can be implicitly provided, while other techniques (such as Grammatical Evolution or Linear GP) employ special genetic operators (which insert or remove chromosome parts) in order to achieve such a complex functionality. The expression encoded in a MEP chromosome may have exponential length when the chromosome has polynomial length due to code reuse [16].

5. Numerical experiments

Several numerical experiments about how the discussed learning algorithms (an SVM and a GP-based classifier) are able to solve a particular image classification task (pedestrian recognition) are presented in this section. To evaluate the performance of the considered classifiers, the Daimler-Chrysler (DC) crop wise data sets (18×36 pixels image size) have been used as provided in [15]. For all datasets a binary classification problem was actually solved: separate the images that contain pedestrians from the images that do not. 4480 images are considered: the decision model is trained on 2240 of them, while 2240 of images are used for testing.

In order to measure the classification performance, the accuracy rate was actually computed. The accuracy rate represents the number of correctly classified items over the total number of items from a given data set. However, the accuracy rate reflects the classification performance of the learning algorithm in a confidence interval. The confidence intervals associated to the performances of the systems must be computed in order to decide if a system outperforms another system. If these intervals are disjoint, then one system outperforms the other one. A confidence interval of 95% is used in order to perform a statistical examination of our results. Therefore, the probability that the accuracy estimations are not in the confidence interval is 5% (see Equation (2)):

$$\Delta I = 1.96 \times \sqrt{\frac{Acc(100 - Acc)}{N}}\% \tag{2}$$

where N represents the number of test examples.

In Table 1 are presented the accuracy rates (and their confidence intervals) by considering different image descriptor kernels (when the image descriptors are actually constructed) and two learning algorithms (SVM and MEP). The performance measures are computed by taking into account the test images and the best identified classifiers (SVM with the best hyper-parameters and MEP with an optimal configuration).

	Exponential	Gaussian	Laplacian
SVM	0.535 ± 0.010	0.657 ± 0.010	0.599 ± 0.010
MEP	0.667 ± 0.009	0.682 ± 0.009	0.737 ± 0.009

TABLE 1. Accuracy rates (%) obtained by SVM and MEP algorithms on images represented by different kernel descriptors.

Several remarks can be done based on the results from Table 1.

Regarding the kernel descriptors, our results indicate that the Gaussian kernel seems to be able to extract the most relevant features from images when the SVM classifier is used, while the Laplacian kernel provides more significant information for MEP. Even if the Gaussian kernel is largely involved in feature extraction process [4], our results suggest that a deeper study should be performed regarding the proper selection of the kernel involved in the feature extraction process. This study might also reveal some criteria for selecting the most appropriate kernel descriptor to use for the input data of a particular problem.

Regarding the learning algorithm, the evolutionary one seems to be able to better generalise over unseen data, compared to SVM. This observation is sustained by the better accuracy rates obtained for all three considered feature extraction methods.

6. Conclusions

A study on how two learning algorithms are able to perform pedestrian recognition in images is presented in this paper. Daimler-Chrysler benchmark image dataset is involved in our numerical experiments.

The first step is to convert each image in a numerical representation relevant for the classifier. Several kernel descriptors are considered on this purpose: Exponential, Gaussian and Laplacian kernels. A statistical algorithm

— SVM — and an evolutionary approach — MEP — are used for the learning phase for which the input data is represented by the previously extracted features.

Better accuracy rates are obtained when using the evolutionary model for all considered kernel descriptors. This might be considered an indicator of the superiority of the evolutionary approach over SVM for the considered problem.

Regarding the kernel descriptors used, SVM learning indicates that the Gaussian is the best one, while MEP achieves the best results by using the Laplacian kernel. Therefore, we can not conclude which is the most efficient kernel descriptor and we intend to perform a further study of how the kernel selection influences the quality of recognition.

Our future work will also include a validation of the obtained results by considering other datasets like Caltech or INRIA.

References

[1] Alonso, I. P., Llorca, D. F., Sotelo, M. A., Bergasa, L. M., de Toro, P. R., Nuevo, J., Ocana, M., and Garrido, M. A. G. Combination of feature extraction methods for SVM pedestrian detection. *IEEE Trans. Intelligent Transportation Systems 8*, 2 (Apr. 2007), 292–307.

[2] Bay, H., Tuytelaars, T., and Gool, L. J. V. SURF: Speeded up robust features. In *ECCV* (2006), pp. I: 404–417.

[3] Bo, L., Lai, K., Ren, X., and Fox, D. Object recognition with hierarchical kernel descriptors. In *CVPR* (2011), IEEE, pp. 1729–1736.

[4] Bo, L., Ren, X., and Fox, D. Kernel descriptors for visual recognition. In *NIPS* (2010), J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., Curran Associates, Inc, pp. 244–252.

[5] Boser, B. E., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In *COLT* (1992), D. Haussler, Ed., ACM Press, pp. 144–152.

[6] Chang, C.-C., and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] Dalal, N., and Triggs, B. Histograms of Oriented Gradients for human detection. In *CVPR* (2005), C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, pp. 886–893.

[8] Dalal, N., Triggs, B., and Schmid, C. Human detection using oriented histograms of flow and appearance. In *ECCV* (2006), pp. II: 428–441.

[9] Diosan, L., and Rogozan, A. How the kernels can influence image classification performance. *Studia Universitatis Babes-Bolyai : Series Informatica LVII*, 4 (2012), 97–109.

[10] Espejo, P., Ventura, S., and Herrera, F. A survey on the application of genetic programming to classification. *Transactions on Systems, Man ans Cybernetic, Part C 40* (2010), 121–144.

[11] Grosan, C., and Abraham, A. *Intelligent systems: A modern approach*. New York, Springer, 2011.

[12] Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[13] Lampert, C. Kernel methods in computer vision. presentation during Computer Vision and Sports Summer School, Prague, 2012, 2012.
[14] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.
[15] Munder, S., and Gavrila, D. M. An experimental study on pedestrian classification. *IEEE Trans. Pattern Analysis and Machine Intelligence 28*, 11 (Nov. 2006), 1863–1868.
[16] Oltean, M. Improving the search by encoding multiple solutions in a chromosome. In *Evolutionary Machine Design: Methodology and Applications*, N. Nedjah and L. de Macedo Mourelle, Eds. Nova Publishers, 2005, ch. 4, pp. 85–110.
[17] Paisitkriangkrai, S., Shen, C. H., and Zhang, J. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. Circuits and Systems for Video Technology 18*, 8 (Aug. 2008), 1140–1151.
[18] Platt, J. Fast training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods — Support Vector Learning* (Cambridge, MA, 1999), B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., MIT Press, pp. 185–208.
[19] Schölkopf, B. The kernel trick for distances. In *NIPS* (Cambridge, MA, 2000), T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., MIT Press, pp. 301–307.
[20] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1995.
[21] Vapnik, V. *Statistical Learning Theory*. Wiley, 1998.
[22] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR 1* (2001), 511–518.
[23] Walk, S., Majer, N., Schindler, K., and Schiele, B. New features and insights for pedestrian detection. In *CVPR* (2010), IEEE, pp. 1030–1037.
[24] Zhao, L., and Thorpe, C. E. Stereo- and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems 1*, 3 (2000), 148–154.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: {anca,lauras,radu,adela}@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

A COMPARISON OF REINFORCEMENT LEARNING BASED MODELS FOR THE DNA FRAGMENT ASSEMBLY PROBLEM

GABRIELA CZIBULA, ISTVAN-GERGELY CZIBULA, AND IULIANA M. BOCICOR[(1)]

Abstract. The *DNA fragment assembly* is a very complex optimization problem important within many fields, such as bioinformatics, computational biology or medicine. The problem is *NP-hard*, that is why many computational techniques, including computational intelligence algorithms, were designed to find good solutions for this problem. This paper is intended to present and investigate two reinforcement learning based models for solving the *DNA fragment assembly* problem. We provide an experimental comparison of these two models, that will study the obtained performances of the reinforcement learning based approaches, by using different action selection policies, with variable parameters.

1. Introduction

Determining the order of nucleotide bases, or the process of DNA sequencing, has nowadays become of great importance in basic biology research, as well as in various fields such as medicine, biotechnology or forensic biology. The main problem with the current sequencing technology is that it cannot read an entire genome at once, not even more than 1000 nucleobases.

The *DNA fragment assembly (FA)* refers to reconstructing the original DNA sequence from a large number of fragments, each several hundred nucleobases long, based on common subsequences of fragments. It is an NP-hard combinatorial optimization problem, growing in importance and complexity as more research centers become involved in sequencing new genomes [5].

Received by the editors: April 12, 2013.

2010 *Mathematics Subject Classification.* 68P15, 68T05.

1998 *CR Categories and Descriptors.* I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*; I.2.8[**Computing Methodologies**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

Key words and phrases. Bioinformatics, DNA fragment assembly, reinforcement learning, Q-learning.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

Reinforcement Learning (RL) [14] is an approach to machine intelligence in which an agent [13] can learn to behave in a certain way by receiving punishments or rewards for its chosen actions. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the highest reward by trying them.

In this paper we aim at investigating two reinforcement learning based models for solving the problem of DNA fragment assembly. One of these models, which was previously proposed in [2], is improved in this study. Moreover, this paper introduces a second reinforcement learning based model for the same problem and compares the two approaches.

The rest of the paper is organized as follows. Section 2 introduces the *DNA fragment assembly problem* as well as some main aspects related to *reinforcement learning*. The reinforcement learning models that we propose for solving the FA problem are detailed in Section 3. Experimental evaluations, analysis and comparisons of the algorithms are given in Section 4. Section 5 outlines some conclusions of the paper and indicates future research directions.

2. Background

This section briefly presents the *DNA FA problem* as well as some fundamental aspects related to *reinforcement learning*.

2.1. **The DNA Fragment Assembly Problem.** Determining the order of nucleotide bases (molecules composing the DNA), or the process of DNA sequencing, has nowadays become of great importance in basic biology research, as well as in fields such as medicine, biotechnology or forensic biology. The main problem with the current sequencing technology is that it cannot read an entire genome at once, not even more than 1000 bases. As even the simplest organisms (such as viruses or bacteria) have much longer genomes, the need to develop methods that would overcome this limitation arose. One of these, called *shotgun sequencing* was introduced in 1982, by Fred Sanger [11] and it consists of the next steps: first, several copies of the DNA molecule are created; then each of the copies is cut at random sites in order to obtain molecules short enough to be sequenced directly - fragments; the last and most difficult step involves assembling these molecules back into the original DNA, based on common subsequences of fragments. The DNA FA problem specifically refers to this last step. For more details, we refer the reader to [2].

2.2. **Reinforcement Learning. Background.** Reinforcement Learning [8] is an approach to machine intelligence that combines two disciplines to solve successfully problems that neither discipline can address individually: *Dynamic programming* and *Supervised learning*. RL is a synonym of learning by

interaction [10]. During learning, the adaptive system tries some actions (i.e., output values) on its environment, then, it is reinforced by receiving a scalar evaluation (the reward) of its actions. The reinforcement learning algorithms selectively retain the outputs that maximize the received reward over time. Reinforcement learning tasks are generally treated in discrete time steps. In RL, the computer is simply given a goal to achieve. The computer then learns how to achieve that goal by trial-and-error interactions with its environment.

An important aspect in reinforcement learning is maintaining an equilibrium between *exploitation* and *exploration* [15]. The agent should accumulate a lot of reward, by choosing the best experienced actions, but at the same time it must explore its environment, by trying new actions. In this study we use three policies for selecting actions: the *ϵ-greedy policy* [14], the *softmax policy* [14] and an *intelligent ϵ-Greedy based action selection mechanism*, that we have introduced in [3], which uses a one step look-ahead procedure in order to better guide the exploration of the agent through the search space.

3. Reinforcement Learning Based Models for the DNA Fragment Assembly Problem

In this section we introduce the two reinforcement learning models proposed for solving the DNA FA problem. First, we present some concepts and notations that will apply to both models.

A general reinforcement learning task is characterized by four components: a *state space* $\mathcal{S}$ specifying all possible configurations of the system; an *action space* $\mathcal{A}$ listing all available actions for the learning agent; a *transition function* δ specifying the possibly stochastic outcomes of taking each action in any state; a *reward function* defining the possible reward of taking each of the actions.

Let us consider that $\mathcal{S}eq$ is a DNA sequence and $F_1, F_2, \ldots, F_n$ is a set of fragments. As indicated in Subsection 2.1, the FA problem consists of determining the order in which these fragments have to be assembled back into the original DNA molecule, based on common subsequences of fragments. Consequently, the FA problem can be viewed, from a computational perspective, as the problem of generating a permutation σ of $\{1, 2, \ldots, n\}$ that optimizes the performance of the alignment $F_\sigma = (F_{\sigma_1}, F_{\sigma_2}, \ldots, F_{\sigma_n})$ $(n > 1)$. The performance measure PM we consider in this paper is one of the fitness functions defined in [9], which sums the overlap scores over all adjacent fragments and has to be maximized. According to [9], the performance measure PM for the sequence of fragments $F_\sigma = (F_{\sigma_1}, F_{\sigma_2}, \ldots, F_{\sigma_n})$ is defined as in Equation (1):

$$PM(F_\sigma) = \sum_{i=1}^{n-1} w(F_{\sigma_i}, F_{\sigma_{i+1}}) \tag{1}$$

where $w(a, b)$ denotes the similarity measure between sequences a and b.

3.1. **Path Finding Model.** We have previously introduced a reinforcement learning based model for the FA problem [2]. In the rest of the paper, it will be referred to as the *path finding model*. The RL task associated to the FA problem consists in training the agent to find a path from the initial to a final state having the maximum associated overall similarity. During the training step of the learning process the learning agent determines its *optimal policy* in the environment, i.e. the mapping from states to actions that maximizes the sum of the received rewards. The equivalent *action configuration* is viewed as a permutation that gives the optimal alignment of the DNA fragments. As the goal is to find a path having the maximum value of the performance measure, the reinforcement function rewards the agent with a small value (e.g. 0.1) for each transition to an non terminal state and with the performance measure of the found alignment, after a transition to a final state [2]. For training the FA agent [2] we used a Q-learning approach [14], in conjunction with an ϵ-Greedy action selection policy. For more details about the definitions of the state and action spaces or reward and transition functions, we refer the reader to [2].

Here we propose a modification of the reward function for this model, in order to better guide the agent towards good solutions. The reward function, as defined in [2], illustrates situations when feedback is given at the end of each trial episode. It will be modified so as to give feedback to the agent after each transition to a new state, even if the state is not final. The moment of rewarding is important in the learning process, as learning happens faster if feedback is being given after each transition from one state to another. If we denote by π a path from the initial to a final state, $\pi = (\pi_0\pi_1\pi_2 \dots \pi_n)$, where $\pi_0 = s_1$ (s_1 is the initial state), by $a_\pi = (a_{\pi_0} a_{\pi_1} a_{\pi_2} \dots a_{\pi_{n-1}})$ - the sequence of actions obtained following the transitions between the successive states from the path π, which gives the alignment of fragments $F_{a_\pi} = (F_{a_{\pi_0}}, F_{a_{\pi_1}}, \dots, F_{a_{\pi_{n-1}}})$ (see [2]), then the new definition of the reward function is given in Formula 2:

$$r(\pi_k|s_1, \pi_1, \pi_2, \dots, \pi_{k-1}) = \begin{cases} 0 & \text{if } \ k = 1 \\ w(F_{a_{\pi_{k-1}}}, F_{a_{\pi_{k-2}}}) & \text{otherwise} \end{cases} \tag{2}$$

where by $r(\pi_k|s_1, \pi_1, \pi_2, \dots, \pi_{k-1})$ we denote the reward received by the agent in state π_k, after its history in the environment is $\pi = (\pi_0 = s_1, \pi_1, \pi_2, \dots, \pi_{k-1})$.

Therefore, after each transition to a new state, the FA agent receives as reward the value of the similarity measure between the fragment corresponding to the current action and the fragment corresponding to the previously taken action. As the learning goal is to maximize the total amount of rewards received on a path from the initial to a final state, the FA agent is actually

trained to find a path π that maximizes the overall similarity of the associated alignment.

3.2. **Permutation Model.** In the following, we introduce a second reinforcement learning based model for the DNA FA problem, considering the general aspects introduced at the beginning of this section. In the rest of the paper, this model will be referred to as the *permutation model*.

The components of the reinforcement learning task are:

- The state space $\mathcal{S}$ (the agent's environment) will consist of $n!$ states, i.e $\mathcal{S} = \{s_1, s_2, \ldots, s_{n!}\}$. Each state $s_i, i = \overline{1, n!}$ in the environment will represent a permutation of all the elements from the fragment set $\{F_1, F_2, \ldots, F_n\} : F_{\sigma^i} = (F_{\sigma_1^i}, F_{\sigma_2^i}, \ldots, F_{\sigma_n^i})$. The initial state will be represented by the identical permutation $s_0 = (F_1, F_2, \ldots, F_n)$. A state s reached by the agent at a given moment will be considered a *terminal (final) state* if the associated performance measure is sufficiently close to a goal value. However, in order to be able to define a final state, apriori knowledge is needed to determine an upper bound of the set of values for the performance measure of all possible permutations.
- The action space consists of $\binom{n}{2} = \frac{n(n-1)}{2}$ actions available to the problem solving agent. Let us denote by $N_a = \frac{n(n-1)}{2}$ the number of actions. An action will be a pair of distinct indices $(i, j), i, j \in \{1, \cdots, n\}, i \neq j$ specifying that the fragments located at indices i and j in the current state will be interchanged in order to make a transition to the following state. Therefore, the action space may be represented as $\mathcal{A} = \{a_1, a_2, \ldots, a_{N_a}\}$, where $a_k \in \{(i, j) | i, j \in \{1, \cdots, n\}, i \neq j\}, \forall 1 \leq k \leq N_a$.
- The transition function $\delta : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ between the states is defined as:

$$\delta(s_j) = \bigcup_{k=1}^{N_a} \Delta(s_j, a_k) \ \forall j \in \{1, \cdots, n!\}, \tag{3}$$

where $\Delta(s_j, a_k) = s_l, \ \forall j, \ l \in \{1, \cdots, n!\}, \ l \neq j$ and s_l is the state resulted from s_j, by interchanging the elements on the positions specified by a_k. This means that, at a given moment, from a state $s \in \mathcal{S}$ the agent can move in N_a successor states, by executing one of the N_a possible actions. The transitions between the states are equiprobable, the transition probability $P(s, s')$ between a state s and each neighbor state s' of s is equal to $1/N_a$.

- The reward function will also be based on the performance measure for a sequence of fragments, as defined in Formula (1). In this case, the

reinforcement function associated to a transition from a state $s_j = F_{\sigma^j}$ to a state $s_l = F_{\sigma^l}$ $(j,\ l \in \{1, \cdots, n!\},\ l \neq j)$, by executing action $a_k, 1 \leq k \leq N_a$, will be:

(4)
$$r(s_l = F_{\sigma^l} | s_j, a_k) = \begin{cases} PM(F_{\sigma^l}) - PM(F_{\sigma^j}) & \text{if } s_l \text{ is non-terminal} \\ PM(F_{\sigma^l}) - PM(F_{\sigma^j}) + PM(s_0) & \text{otherwise} \end{cases}$$

where s_0 is the identical permutation.

The *FA* problem formulated as a RL problem will consist in training the agent to find a sequence of actions from the initial to the final state so as to maximize the total amount of received rewards, which equates to finding a permutation having the associated performance measure sufficiently close to a given maximum (goal) value.

4. Experiments

This section aims to comparatively evaluate the two reinforcement learning based approaches described in Section 3.

4.1. **Case Study.** The tests are performed on a small section of DNA belonging to the bacterium *Escherichia coli (E. coli)*. The DNA sequence contains 25 bases: $TACTAGCAATACGCTTGCGTTCGGT$. Using the Perl scripts that the authors of [16] produced to generate fragments from a given DNA reference, for the above mentioned *E. Coli* sequence, we obtained 8 fragments, each having a length of 10 bases: $F_1 = TGCGTTCGGT$, $F_2 = TACGCTTGCG$, $F_3 = ATACGCTTGC$, $F_4 = TACTAGCAAT$, $F_5 = GCAATACGCT$, $F_6 = CAATACGCTT$, $F_7 = CTAGCAATAC$, $F_8 = AATACGCTTG$.

These fragments are ordered in the following way to form the original DNA sequence: $F_4F_7F_5F_6F_8F_3F_2F_1$. The maximum value for the performance measure (Equation 1) is 56.01 and it is obtained for two cases: the above alignment, indicating the original DNA sequence, as well as its reverse. The overlap similarity scores for all possible pairs of fragments are obtained using the Smith-Waterman algorithm [12].

For applying the RL models to solve the FA problem, we used a software framework that we have previously introduced in [4] for solving combinatorial optimization problems using reinforcement learning techniques.

We compare the two models by running the corresponding implementations, using the Q-Learning algorithm [14], in conjunction with the three action selection mechanisms we mentioned in Subsection 2.2. Regarding the parameter setting, for all tests we used the following values: the discount factor for the future rewards is $\gamma = 0.9$; the learning rate is $\alpha = 0.95$; the number of

training episodes is $4 \cdot 10^5$. In the case of the permutation model, the maximum value of the performance measure was defined to be 57 and a state was considered final if its performance measure was at most 1 unit away from this maximum. For each of the three action selection policies, tests were made for different values of the policy parameter (ϵ - in the case of ϵ-greedy and the one step look-ahead procedure and τ - in the case of softmax): $\{0.2, 0.4, 0.6, 0.8\}$. Each algorithm was run five times, for each case of selection policy and each value of the policy parameter and the shown results are averaged over these runs. We mention that the experiments were carried out on a PC with an Intel Core i3 Processor at 2.4 GHz, with 3 GB of RAM.

4.2. **Comparative Results.** Below we present the results obtained by the Q-learning based algorithms implementing the models described in Section 3.

The path finding model proves to obtain the correct alignment of fragments with all three action selection mechanisms, the difference being the number of training epochs needed to converge to the optimal solution. The left-hand side of Figure 1 illustrates the performance measure of the solutions obtained during the training process. For all three action selection policies, the algorithm converges to the optimal solution, the one having a performance measure of 56.01. We note that the algorithm using the one step look-ahead procedure achieves the fastest convergence, reaching the correct solution after only 10^3 training epochs, for $\epsilon = 0.8$. The next best, in terms of training epochs until convergence, is the algorithm using the softmax policy, the last one being the ϵ-greedy algorithm. In all three cases, as the values of the policy parameter increase, thus leading to more exploration of the states space, the convergence is achieved more rapidly: on average, for $\epsilon = 0.2$ and $\tau = 0.2$, the algorithm converges (to the optimal or near-optimal solution) after $153 \cdot 10^3$ epochs, while for $\epsilon = 0.8$ and $\tau = 0.8$, the maximum performance measure is reached, on average, after only $25 \cdot 10^3$ epochs.

In terms of accuracy, the permutation model performs equally well, determining the correct alignment for all the tests. In what concerns the number of training epochs, this second approach outperforms the first, for the case study we considered. As can be seen on the right-hand side of Figure 1, the permutation model needs fewer epochs to reach the right solution: on average, for all three action selection policies, convergence is reached after $\sim 6 \cdot 10^3$ epochs, as opposed to $\sim 84 \cdot 10^3$ epochs, which are needed on average for the path finding model. In this case, all three action selection policies find the solution equally fast, but for different values of the policy parameter. Generally, as the value of the policy parameter increases, the convergence is achieved sooner.

In the following, we will also offer an analysis of the computational time needed for the proposed RL algorithms in order to reach the correct solution.

Table 1 illustrates the average time (in seconds) needed for the two algorithms implementing the models that we presented in Section 3. Here, p represents the

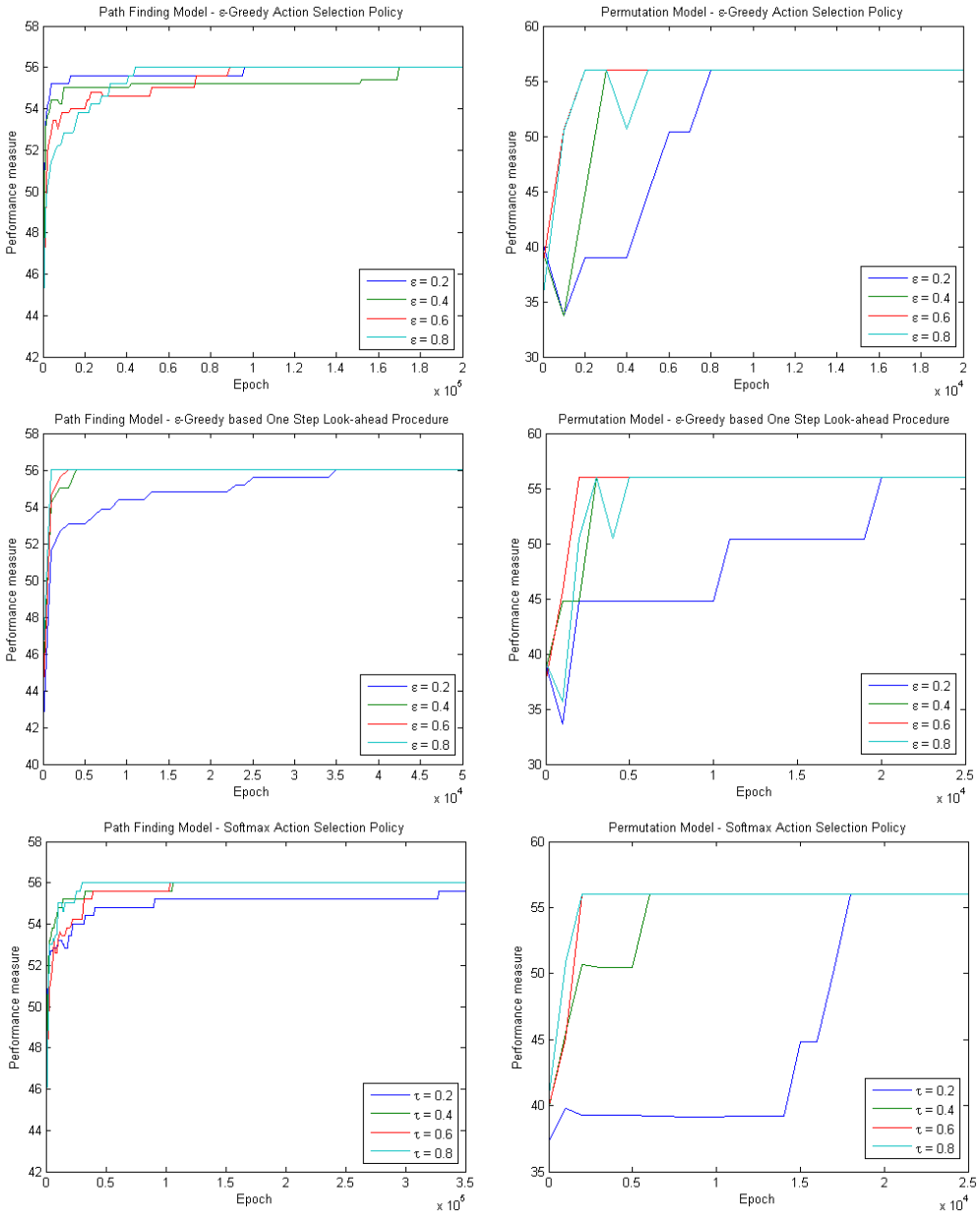


FIGURE 1. Illustration of the learning process for the two RL models.

	Path Finding Model				**Permutation Model**			
	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$
ϵ-Greedy policy	2.8	9.2	14.8	11.4	< 2	< 2	< 2	< 2
Look-ahead procedure	2.2	< 2	< 2	< 2	4	< 2	< 2	< 2
Softmax policy	29.6	17.4	40.6	10.2	705.8	44.6	35.6	26.8

TABLE 1. Average computational time (in seconds) needed for each of the presented algorithms to reach the correct solution.

policy parameter, referring to ϵ, in the case of the ϵ-greedy based mechanisms and to τ, for the softmax action selection policy.

Even though intuitively the time is directly correlated with the number of training epochs, between the two models there are some major differences, which lead to different durations for epochs. For instance, in the case of the path finding model, the agent executes exactly 8 actions, to reach a final configuration. For the permutation model, however, the number of actions is different in each epoch, as the agent keeps trying actions until it reaches a state having a performance measure as close as possible to the maximum defined value. Hence, according to the way the agent explores the search space and the way it exploits existing knowledge, an epoch in the permutation model could last less or longer than one in the path finding model. From Table 1 we notice that for the ϵ-greedy mechanism, the solution is reached in less time using the permutation model. Even if an epoch in the permutation model would last longer, the number of epochs needed until convergence is more than 26 times less than the number of epochs until convergence in the path finding approach. Another important observation concerns the softmax policy. An agent guiding its search by softmax must rank all the possible actions from a state, which requires extra computations that are not necessary for the other two policies. Therefore, as shown in Table 1, the algorithms using softmax need more time for each epoch, consequently more time, in total. Futhermore, regarding the permutation model, from any given state, there are 28 possible actions (compared to only 8, for the other model) and then the computations needed for softmax take even longer, as can be seen from Table 1.

4.3. **Discussion and Comparison to Related Work.** We have experimented with two reinforcement learning based models, proposed for the problem of DNA fragment assembly, used in conjuction with three different action selection mechanisms: ϵ-greedy [14], an intelligent ϵ-Greedy based look-ahead action selection mechanism that we have previously introduced [3] and the softmax selection policy [14]. The *path finding model* was introduced in a different study [2] and uses a Q-learning algorithm, with an ϵ-greedy action

selection policy. In this paper we have improved the path finding model and we have introduced the *permutation model*, which is also based on Q-learning.

Both algorithms demonstrated to find the correct alignment of fragments, for the DNA sequence that we considered for the experiments. The difference between them lies in the number of training epochs and the computational time each one needs to achieve convergence.

For the considered case study and parameter setting, the permutation model proved to outperform the path finding approach, in all cases, when inspecting the number of epochs. In what concerns the computational time, the situation is the same, for the ϵ-greedy policy, when the permutation based algorithm finds the correct solution in less than 2 seconds. Still, for the softmax action selection policy, the required time is considerably higher. We remark that in both cases, the algorithms using the intelligent action selection procedure [3] converge, on average, in less epochs than those using ϵ-greedy or softmax, as this procedure efficiently guides the exploration of the search space. We will further investigate how an intelligent action selection mechanism, based on softmax instead of ϵ-Greedy, could influence the outcome.

Regarding the permutation model, we note the following. As it is difficult to determine a final state, we need apriori information about the possible values of the performance measure for permutations. Another option would be to consider a state terminal if the number of actions that were performed from the initial state equals a given maximum number of steps. However, the problem with this course of action could be the fact that the agent would not aim to maximize the total sum of rewards, but the sum of rewards obtained after the given number of maximum steps. A second observation concerns the action space. In this study we use a fairly simple definition of an action, but further work will be done to investigate new types of actions, such as operators inspired from the field of genetic algorithms (e.g. crossover, mutation).

We remark that for the permutation model the number of states of the environment is smaller than for the path finding approach. However, an important drawback of the permutation model is the fact that apriori knowledge about the problem is needed in order to define a final state and this information is not available in all types of situations. Therefore, the path finding approach is more general and, as the results it obtains are good both in terms of accuracy and in terms of computational time, we conclude that it is better.

In the following, we will briefly compare our models with other approaches existing in the literature for the FA problem. Since the used data sets are different from one study to another, we cannot offer a very detailed comparison.

Kikuchi and Chakraborty [6] present an improved genetic algorithm to approach the FA problem. To improve the efficiency of the simple genetic algorithm (regarding both speed and solution quality), the authors introduce

two new ideas, one of which implies manually combining fragments at certain generations. Compared to the approach from [6], our RL based methods do not need user intervention during the training process of the agent.

A comparison of four heuristic DNA FA algorithms is provided by Li and Khuri in [7]: a genetic algorithm, a greedy algorithm, a clustering heuristic algorithm and one using structured pattern matching. All algorithms are experimentally evaluated on several data sets, with the number of fragments ranging from 39 to 773. The authors determined that the running times of all four algorithms ranged from a few seconds to several hours. Even though the data set we use contains a smaller number of fragments, the maximum amount of time needed by the RL approaches is less than 12 minutes, while the minimum is less than 2 seconds. Therefore, we believe that even for larger instances it is likely that the necessary time for the RL models to converge to the correct solution is of the order of minutes, rather than hours.

Angeleri et al. [1] introduce a supervised approach, based on a recurrent neural network to solve the FA problem. In the case of supervised models, a set of inputs with their target outputs is required. The advantage of our RL methods is that the learning process needs no external supervision, as in our approach the solution is learned from the rewards obtained by the agent during its training. Still, as mentioned before, the permutation model requires apriori knowledge about the problem.

5. Conclusions and Further Work

In the present study we investigated two reinforcement learning based models (the *permutation model* and the *path finding model*) for an important problem in bioinformatics, namely the DNA fragment assembly problem. The algorithms implementing both approaches, using three different action selection policies, have been experimentally evaluated and compared.

We plan to extend the evaluation of both Q-learning based algorithms for larger DNA sequences, and implicitly greater number of fragments, to further develop the analysis. We will also investigate possible improvements of these models by adding various local search mechanisms, by combining the softmax policy with the intelligent action selection procedure introduced in [3], by decreasing the action selection parameters (ϵ and τ) during the training process or by extending the permutation model to a distributed RL approach.

ACKNOWLEDGEMENT

This work was partially supported by the Sectoral Operational Programme for Human Resources Development 2007-2013, co-financed by the European

Social Fund, under the project number POSDRU/107/1.5/S/76841 with the title Modern Doctoral Studies: Internationalization and Interdisciplinarity.

References

[1] E. Angeleri, B. Apolloni, D. de Falco, and L. Grandi. DNA fragment assembly using neural prediction techniques. *Int. J. Neural Syst.*, 9(6):523–544, 1999.

[2] M. Bocicor, G. Czibula, and I. G. Czibula. A Reinforcement Learning Approach for Solving the Fragment Assembly Problem. In *Proceedings of the 3th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC '11)*, pages 191–198. IEEE Computer Society, 2011.

[3] G. Czibula, I. M. Bocicor, and I. G. Czibula. Temporal Ordering of Cancer Microarray Data through a Reinforcement Learning Based Approach. *PLoS ONE*, 8(4):e60883, 2013.

[4] I. G. Czibula, G. Czibula, and M. I. Bocicor. A Software Framework for Solving Combinatorial Optimization Tasks. *Studia Universitatis "Babes-Bolyai", Informatica*, Special Issue, LVI(3):3–8, 2011.

[5] A. E. Hassanien, M. G. Milanova, T. G. Smolinski, and A. Abraham. Computational intelligence in solving bioinformatics problems: Reviews, perspectives, and challenges. In *Computational Intelligence in Biomedicine and Bioinformatics*, pages 3–47. 2008.

[6] S. Kikuchi and G. Chakraborty. Heuristically tuned GA to solve genome fragment assembly problem. *IEEE CEC*, pages 1491–1498, 2006.

[7] L. Li and S. Khuri. A comparison of DNA fragment assembly algorithms. In *Proc. of the Int?l Conf. on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 329–335. CSREA Press, 2004.

[8] L. J. Lin. Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching. *Machine Learning*, 8:293–321, 1992.

[9] R. J. Parsons, S. Forrest, and C. Burks. Genetic algorithms, operators, and DNA fragment assembly. In *Machine Learning*, pages 11–33. Kluwer Academic Publishers, 1995.

[10] A. Perez-Uribe. Introduction to reinforcement learning, 1998. http://lslwww.epfl.ch/∼anperez/RL/RL.html.

[11] F. Sanger, A. Coulson, G. Hong, l. D. Hil, and G. Petersen. Nucleotide sequence of bacteriophage Lambda DNA. *J. Molecular Biology*, 162(4):729–773, 1982.

[12] T. Smith and M. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[13] I. Susnea, G. Vasiliu, A. Filipescu, and A. Radaschin. Virtual pheromones for real-time control of autonomous mobile robots. *Studies in Informatics and Control*, 18(3):233–240, 2009.

[14] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[15] S. Thrun. The Role of Exploration in Learning Control. In *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold, Florence, Kentucky, 1992.

[16] W. Zhang, J. Chen, Y. Yang, Y. Tang, J. Shang, B. Shen, and I. K. Jordan. A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies. *PLoS ONE*, 6(3):e17915, 2011.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: {gabis, istvanc, iuliana}@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

BUSINESS PROCESS MINING USING ITERATED LOCAL SEARCH

DIANA CHINCES[(1)] AND IOAN SALOMIE[(1)]

Abstract. This paper introduces ILS Miner, a novel method for discovering process models from event logs using iterated local search. A comparison between ILS Miner, GLS Miner and HeuristicMiner is presented in this paper. GLS Miner was chosen because applies Guided Local Search for discovering process models from event logs, HeuristicMiner[10] being a legacy method in business process discovery. It is shown that ILS Miner can discover process models that correctly map to the event log. ILS Miner works with business processes represented as graphs, and the final discovered process is represented as a BPMN diagram.

1. Introduction

The work efficiency in large organizations is given by optimal process flows that are followed in the company. In some cases, these process flows are just defined and followed by employees, but in most of the cases the process flows (models) need to be discovered. Nowadays, organizations use different management systems that keep track of all the work being done, generating so called event logs. The process model that is being followed in the organization can be discovered from these event logs using process mining algorithms.
Figure 1 shows an overview of process mining. The event logs are generated by the information system used in the company and are the input for the process discovery algorithms. Not all the information from the database is included in the event logs. In [3] the following assumptions are defined: each event refers to an activity (well-defined step in the process), each event refers to a case (the

Received by the editors: April 13, 2013.

2010 *Mathematics Subject Classification.* 68-00, 68M14, 68M01, 68N01, 68P10 .

1998 *CR Categories and Descriptors.* K.4.3 [**Organizational Impacts**]: *Reengineering*; K.4.3 [**Organizational Impacts**]: *Automation*; K.4.3 [**Organizational Impacts**]: *Computer-supported collaborative work*.

Key words and phrases. Guided Local Search, GLS Miner, Iterated Local Search, business process mining, event log, BPMN .

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

process instance), each event has a performer, and events have a timestamp and are totally ordered. Knowing this, an event can be defined as a tuple (c, a, p), where c is the case, a the activity and p the performer. The events being ordered in time, causal relations can be defined between the activities and the actors. Given two tuples (c, a1, p1) and (c, a2, p2) from the event logs, it results that activity a1 that is performed by person p1 is followed by activity a2, which is performed by person p2. For applying Guided Local Search for process discovery the performer is not needed, the algorithm only makes use of the cases, the activities and their order.

Three different perspectives of process mining were identified in [4]: the pro-

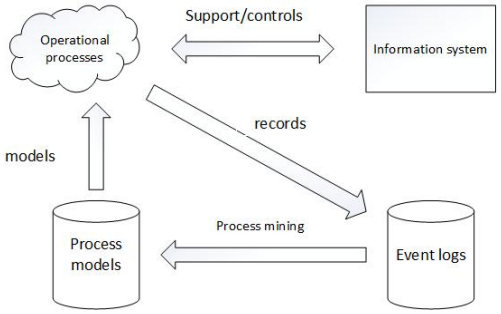


FIGURE 1. Overview of process mining[2]

cess perspective, the organizational perspective and the case perspective. The process perspective deals with discovering the process models from the event logs. Areas of applicability of the process perspective include IT management processes governing the operations of services and infrastructure as well as scientific computing workflows. This process model can be further used as input for applying optimization algorithms for improving the workflows in the organization. The organization perspective focuses on the relations between the individuals in the company and the case perspective mines the properties of the cases in the process. An important property of a case is the data attached to it. For example, if a case represents submitting an order it would be interesting to have knowledge of the products ordered, the quantity, prices paid, discounts etc.

This paper aims at presenting a novel method for discovering process models from event logs using Iterated Local Search, focusing on the process perspective.

The remainder of the paper is organized as follows. Section 2 presents the background of process mining and related work. Section 3 explains the Iterated Local Search algorithm. Section 4 describes how Iterated Local Search is used for process discovery, followed by Section 5 where the experiments

of applying ILS Miner in process discovery are presented. In Section 6 the conclusions of this paper are stated.

2. Background and related work

This section reviews the state of the art in business process discovery and describes a common structure used for storing event logs for process mining. The information systems presented in Figure 1 use different data structures to store the data. In [5] a generic framework for process discovery algorithms and a common structure to be used were introduced. The ProM framework allows the creation of plugins for the process mining algorithms. The input for a plugin is an XML file in the MXML format [5]. Tools like Open Xes [6] allow the mapping and conversion between different log storing structures and MXML files.

Process mining algorithms, such as the Alpha algorithm [8] assume that it is possible to sequentially order events, such that each event refers to a case and an activity. The Alpha algorithm does not take into consideration the timestamp of events. This information is used to automatically construct a process model, which in the Alpha algorithm is represented as a Petri Net describing the event logs.

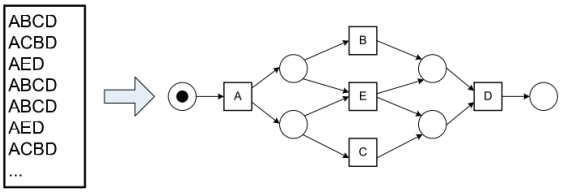


Figure 2. The event log and the process model discovered by Alpha algorithm [9]

In Figure 2 a simple event log and the Petri-Net that was generated by applying the Alpha algorithm can be observed. It can be noted from the figure that the event log is easily reproduced from the discovered model.

The GLS Miner[14] is the first algorithm in which local search is applied in the business process domain. It is shown in [14] that the GLS Miner can correctly discover heterogenous types of business processes. The search space for applying guided local search is a set of processes, represented as graphs. A process model is represented as a graph. The nodes of the graph represent the existent activities in the process model and the edges the flows between the activities. The features necessary for guided local search were defined as

all possible edges in the graph. Thus, a solution has a feature available if and only if the specific edge is available in the solution. The algorithm starts with a random graph and can either add or remove edges in order to move from a solution to another.

The HeuristicMiner [10] is a heuristic driven process mining algorithm, being considered one of the most relevant process discovery algorithms. It is a practical applicable mining algorithm that can deal with noise (e.g. exceptions from the usual workflow that appear in the event log) and can be used to express the main behavior registered in an event log. The HeuristicMiner only considers the order of the events within a case by using the timestamp of the activities. A log is defined in the same way as for the Alpha algorithm [9]. The following log will be used for explaining the HeuristicMiner, W = [ABCD, ABCD, ACBD, ACBD, AED].

HeuristicMiner builds a dependency graph [10]. A frequency based metric is used to indicate how certain we are that there is truly a dependency relation between two events A and B:

$$a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$$

A high value of the $\Rightarrow_W$ relation between a and b determines that there is dependency relationship between a and b. Applying the heuristics to the event log in Table 1 takes us to the results in Figure 3.

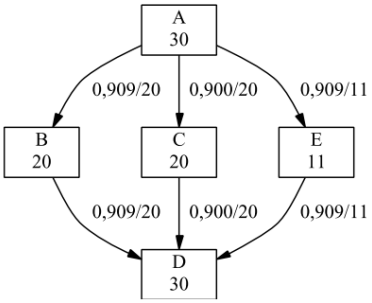


FIGURE 3. Dependency graph [10]

3. Iterated Local Search

Iterated local search can be easily explained as [12]: one iteratively builds a sequence of solutions generated by the embedded heuristic, leading to better

solutions than if one were to use repeated random trials of that heuristic. There are two main points that make an algorithm an iterated local search: (i) there must be a single chain that is being followed (thus excluding population-based algorithms); (ii) the search for better solutions occurs in a reduced space defined by the output of a blackbox heuristic.

A local search is a problem specific heuristic optimization algorithm. In most cases this algorithm can be improved iteratively, resulting in a iterated local search algorithm. Algorithm 1 shows the generic iterated local search algorithm.

begin
  $s_0 \leftarrow GenerateInitialSolution$;
  $s* \leftarrow LocalSearch(s_0)$;
  while *not termination criterion* **do**
    $s' \leftarrow perturbation(s*, history)$;
    $s'* \leftarrow LocalSearch(s')$;
    $s* \leftarrow AcceptanceCriterion(s*, s*', history)$;
  end
end

Algorithm 1: Iterated Local Search Algorithm [12]

Iterated local search runs for each problem until the termination criteria is met. The termination criteria has to be created for each particular problem. The iterated local search walk is not reversible. First, an initial solution is generated on which the search algorithm is applied. Until the termination criterion is met, the current best solution is perturbated and if it meets the acceptance criterion it may be used as the next best solution. A history of perturbations can be included in the algorithm, so that the same perturbation is not applied several times on the same solution.

4. Iterated Local Search Miner

From the previous section presenting the general Iterated Local Search algorithm, we can conclude that for applying this for a specific problem, the following items need to be defined:

- a termination criterion
- a method for perturbation
- an acceptance criterion

The search space for applying iterated local search is a set of processes, represented as graphs. The nodes of the graph represent the existing activities in the process model while the edges the flows between the activities. Algorithm 2 shows the Iterated Local Search Miner (ILS Miner).

```
input  : event log, maxIterations
output: process model mapped on the event log
begin
    solution ← random(log);
    currentCost ← computeCost(solution);
    neighbor ← findNeighbor(solution);
    k ← 0;
    while neighbor ≠ null and k < maxIterations do
        solution ← neighbor;
        perturb(solution);
        second ← findNeighbor(solution);
        currentCost ← computeCost(solution);
        secondCost ← computeCost(second);
        if secondCost ¡ currentCost then
            solution ← second
        end
        k ← k + 1;
    end
    returnsolution;
end
```

Algorithm 2: ILS Miner

We considered the termination criterion being the fact that the algorithm cannot find a better neighbor and the number of maximum iterations have been executed. Algorithm 3 shows the perturbation procedure.

```
input  : graph
output: graph'
begin
    edge ← randomEdge();
    graph.add(edge);
    return graph
end
```

Algorithm 3: Perturbation algorithm

In the perturbation algorithm the graph is randomly modified. The algorithm perturbes the graph by adding a random edge to it. The perturbated solution and its neighbor is considered to pass the acceptance criterion if it has a lower cost than the previous best solution. We have chosen to compute he cost as the sum of the frequency of edges in the graph. The frequency of an edge *(k,s)* is computed based on the event log given as input to the mining algorithm. The frequency of the occurrence *(k,s)* is computed as

$$\frac{(number\ of\ occurrences\ of\ (k,s))}{(number\ of\ occurences\ of\ (s,k)\ +\ 1)} \tag{1}$$

5. Experiments and results

The Iterated Local Search BP Miner was implemented as a ProM Framework[5] plugin. This plugin was created for proof of concept of the algorithm described in the previous section.

The results are presented as a comparison between the proposed Iterated Local Search BP Miner, the Guided Local Search BP Miner[14] and the Heuristic Miner[10]. This comparison is done considering the discovered process models and the errors encountered by each of the algorithms. Process models P1 (Figure 4), P2 (Figure 5) and P4 (Figure 6) were considered for this comparison. P1 is a simple process model, having just a few flows from the start to the end activity. P2 is more complex, having alternative flows from the start to the end activity. P3 has the most complexity as it contains loops. We considered for the experiments these three basic types of processes for prove of concept. In the future, we plan to run scalability tests on the algorithm using event logs from real data. Table 1 shows the event logs that were used for each process. The regularization value used for the experiments was 0.6, however this is easily customizable in the ProM Framework plugin.

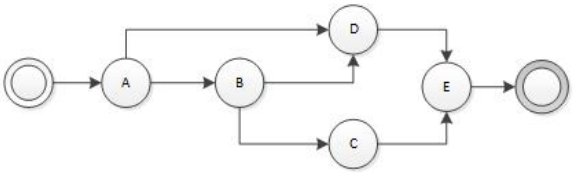


Figure 4. P1 - simple process model

Table 2 shows the comparison of the discovered process models by applying the ILS BP Miner and the GLS BP Miner. For process model P1 the ILS BP Miner, as well as the GLS Miner, have correctly discovered the model from the given event logs. We can observe that the Guided Local Search Miner has discovered a process model that correctly maps to the event log, even if this

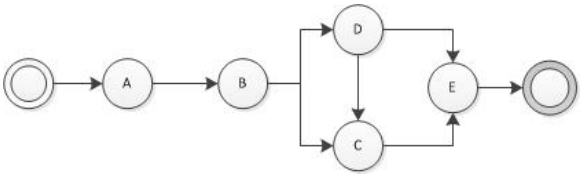


FIGURE 5. P2 - simple process model with alternative paths

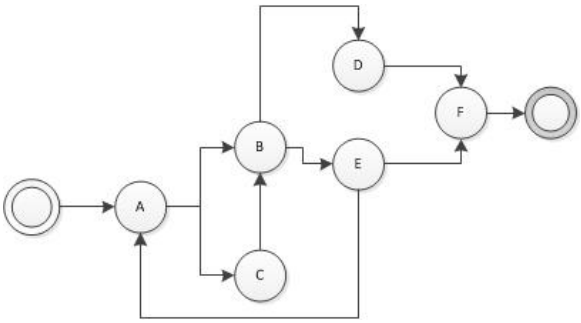


FIGURE 6. P2 - process model with loops

is not the same with the initial process model. Applying a process mining algorithm in an organization might end up with this kind of results, that the process model that is discovered from the event logs is not exactly the same with the process model that was initially defined in the organization. For process model P2 the both algorithms have correctly discovered the initial model, as shown from the error graph as well. For the last process model, the GLS Miner did not discover the entire model, but managed to obtain a nearly correct process model. ILS Miner however discovered the a model which contains less errors than the model discovered by the GLS Miner.

Table 3 shows the comparison of the discovered process models by applying the ILS BP Miner and the Heuristic Miner[10]. For process model P1 the ILS BP Miner, as well as the GLS BP Miner, have correctly discovered the model from te given event log. We can observe that for process model P2, ILS BP Miner has correctly discovered the process model, but the Heuristic Miner is missing the edge from **D** to **E**. For the last process model, ILS BP Miner has managed to obtain once again a model that is better than the Heuristic Miner. This can be observed also from Figure 6, showing the number of errors encountered by each of the algorithms.

Figure 7 shows the number of errors encountered by each of the three algorithms that were experimented.

Traces for P1			**Traces for P3**		
Case	**Activity**	**Time**	1	A	19:00:00
1	A	19:00:00		B	19:01:00
	B	19:01:00		D	19:02:00
	D	19:02:00		F	19:03:00
	E	19:03:00	2	A	19:00:00
2	A	19:00:00		B	19:01:00
	C	19:01:00		E	19:02:00
	E	19:02:00		F	19:03:00
3	A	19:00:00	3	A	19:00:00
	B	19:01:00		C	19:01:00
	D	19:02:00		B	19:02:00
	E	19:03:00		E	19:03:00
Traces for P2				F	19:04:00
1	A	19:00:00	4	A	19:00:00
	B	19:02:00		C	19:01:00
	C	19:03:00		E	19:02:00
	E	19:04:00		F	19:03:00
2	A	19:00:00	5	A	19:01:00
	B	19:01:00		B	19:02:00
	D	19:02:00		E	19:03:00
	C	19:03:00		A	19:04:00
	E	19:04:00		B	19:05:00
3	A	19:00:00		D	19:06:00
	B	19:01:00		F	19:07:00
	D	19:02:00	6	A	19:01:01
	E	19:03:00		C	19:02:00
				B	19:03:00
				E	19:04:00
				F	19:05:00

TABLE 1. Traces considered for experiments

6. Conclusion

This paper introduced ILS Miner, a new algorithm for discovering process models from event logs. The experiments were done on heterogenous types of processes and the results were compared with GLS Miner[14], another algorithm that applies local search for process discovery and HeuristicMiner[10], a legacy algorithm in process discovery. This paper shows that ILS Miner has

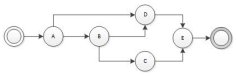
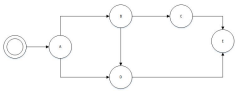
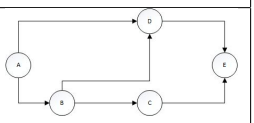
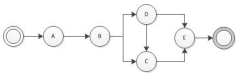
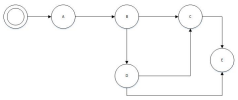
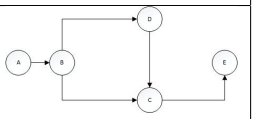
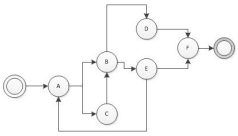
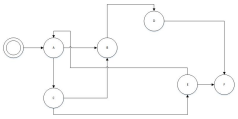
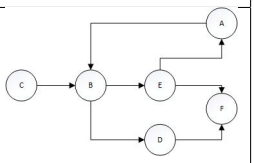
Process model	Initial process (known)	ILS BP Miner	GLS BP Miner
P1			
P2			
P3			

TABLE 2. The initial (known) process model and the discovered models using ILS Miner and GLS Miner

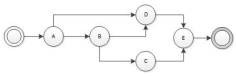
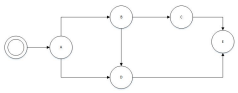
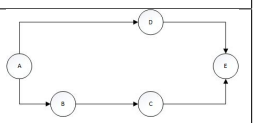
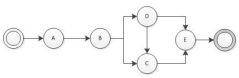
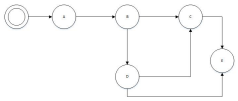
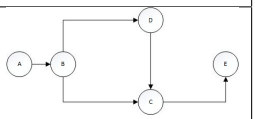
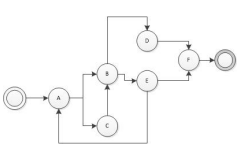
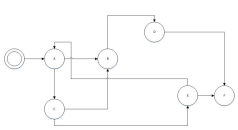
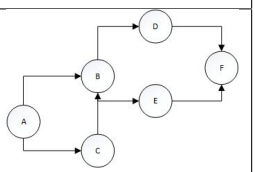
Process model	Initial process (known)	ILS BP Miner	Heuristic Miner
P1			
P2			
P3			

TABLE 3. The initial (known) process model and the discovered models using ILS Miner and Heuristic Miner

managed to correctly discover the process models from the event logs, showing that the results are better than the ones obtained by both of GLS Miner and HeuristicMiner. In the future, we plan to apply our methods on more complex process models and on real data, comparing the results by the performance of the algorithms.

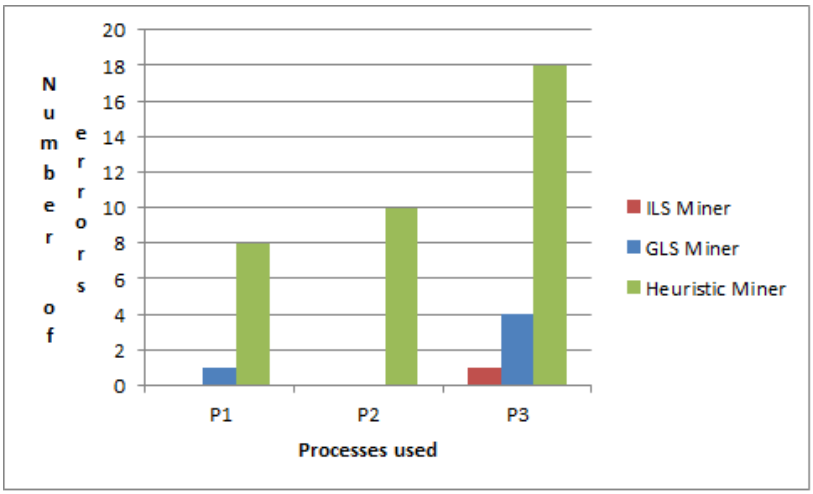


FIGURE 7. Number of errors encountered by the algorithms

REFERENCES

[1] Chris Voudouris, Edward Tsang, *Guided Local search*, Technical Report CSM–247, Departament of Computer Science, University of Essex, August 1995

[2] A.K. Alves de Medeiros, A.J.M.M. Weijters and W.M.P. van der Aalst, *Genetic Process Mining An Experimental Approach*, Data Mining and Knowledge Discovery Volume 14 Number 2 (2007), pages 245-304

[3] W.M.P. van der Aalst, A.K.A. de Medeiros, *Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance*, Second International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004), pages 6984

[4] W.M.P van der Aalst, *Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing*, Requirements Engineering Volume 10 Number 3 (2005), pages 198-211

[5] B.F. van Dongen and W.M.P. van der Aalst, *A Generic Import Framework for Process Event Logs*, J. Eder and S. DUstdar, editors, Business Process Management Workshops, Workshop on Business Process Intelligence (BPI 2006), volume 4103 of Lecture Notes in Computer Science, pages 81-92

[6] Open Xes Homepage, http://xes-standard.org/openxes/start

[7] Christos Voudouris, Edward Tsang, *Guided Local Search and its application to the traveling salesman problem*, European Journal of Operational Research 113 (1999), pages 469-499

[8] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. *Workflow Mining: Discovering Process Models from Event Logs.* IEEE Transactions on Knowledgeand Data Engineering, 16(9):11281142, 2004

[9] W.M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. Gunther, *Process Mining : A Two-Step Approach using Transition Systems and Regions*, Software and System Modeling Volume 9 Number 1 (2010) pages 87-111

[10] A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros, *Process Mining with the HeuristicsMiner Algorithm*, BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven

[11] W.M.P van der Aalst, *Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing*, Requirements Engineering Volume 10 Number 3 2005, pages 198-211

[12] Iterated Local Search, http://ideas.repec.org/p/upf/upfgen/513.html

[13] Helena R. Lourenco, Olivier C. Martin, Thomas Stutzle, Iterated Local Search, *Handbook of Metaheuristics, F. Glover and G. Kochenberger, Eds. International Series in Operations Research & Management Science, vol. 57. Kluwer Academic Publishers, Norwell, MA, 321353*

[14] Diana Chinces, Ioan Salomie, Business Process Mining using Guided Local Search, *The 12th International Symposium on Parallel and Distributed Computing - ISPDC 2013 (accepted)*

(1) Technical University Of Cluj-Napoca
E-mail address: diana.chinces@cs.utcluj.ro

E-mail address: ioan.salomie@cs.utcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

PLAYERS WITH UNEXPECTED BEHAVIOR: t-IMMUNE STRATEGIES. AN EVOLUTIONARY APPROACH.

NOÉMI GASKÓ(1), MIHAI SUCIU(1), RODICA IOANA LUNG(1), TUDOR DAN MIHOC(1), AND D. DUMITRESCU (1)

Abstract. An evolutionary detection method based on generative relations for detecting the t-immune strategies of a non-cooperative game is introduced. Numerical experiments on an economic game show the potential of our approach.

1. Introduction

Game Theory (GT) offers proper models to characterize interactions between agents with conflicting behaviours. The situations where divergent interests interact are modelled as mathematical games. Each player has a set of strategies ("moves") that define his possible actions within the game. Many types of games have been proposed since the concept was first introduced: games with complete information (players have complete information on the entire game), cooperative or non-cooperative games (depending on the players' disposition to build unions or not), one shot games (players play one round only in the same time), etc.

One of GT's main aim was to find patterns and solutions that will allow scholars to accurate anticipate game's outcomes and the behaviour of real players. An equilibrium concept designed for pure rational players, was introduced by Nash [8] and depicts that state where no individual player can gain more by modifying his option within the game (his strategy) while the others keep theirs unchanged. Even if it is one of the central solution concepts in GT, Nash equilibrium was also criticized mostly because of the hard assumptions on players rationality [6]. Experiments conducted with real people lead to the

Received by the editors: April 15, 2013.

2010 *Mathematics Subject Classification.* 91A10.

1998 *CR Categories and Descriptors.* I.2.8 [**Heuristic methods**]: *Artificial intelligence*;

Key words and phrases. non-cooperative games, t-immune strategy, evolutionary detection.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

conclusion that Nash equilibrium is seldom the output for games with real players.

The search for more realistic models for real players leaded to entire classes of equilibria. The scholars tried to incorporate the "human" factor in these models but, even if at some point they succeeded at this task, the proper tools to solve them were missing. Evolutionary computation can offer a solution for this. The recent development of fitness solutions for game equilibria detection [5], [4] allows specialized techniques on strategic games to be developed.

In this study we present a tool, based on evolutionary computation, designed to detect a good approximation of a game t-immune equilibria. This equilibria attempts to capture the situations where agents are acting in an unpredictable manner, an irrational behaviour outlined by most of the experiments with real people.

The paper is organized in four sections: an introduction that presents the domain and emphasise the importance of the approached problem; in the second section the proposed technique is presented; the conducted numerical experiments that validate the method are depicted in section three followed by the conclusions and further work section.

2. Evolutionary equilibrium detection for t-immune equilibria models

In order to detect the t-immune equilibrium for noncooperative games in strategic form a generative relation is introduced. Using this relation a fitness measure is constructed and two evolutionary methods are modified and adapted to detect t-immune equilibria.

2.1. Prerequisites.

A finite strategic game is a system $G = ((N, S_i, u_i), i = 1, ..., n)$, where:

- N represents a set of players, and n is the number of players;
- for each player $i \in N$, S_i is the set of actions available,

$$S = S_1 \times S_2 \times ... \times S_n$$

  is the set of all possible situations of the game.
  Each $s \in S$ is a strategy (or strategy profile) of the game;
- for each player $i \in N$, $u_i : S \to R$ represents the payoff function of i.

The most used solution concept in Game Theory is the Nash equilibrium [8]. Playing in Nash sense means that no one from the players can change her/his strategy in order to increase her/his payoff while the others keep theirs unchanged.

t-immune equilibrium [1] models situations where players act unpredictable, not in a rational or expected way. A strategy is t-immune when less than t

players change their strategy, but without affecting the payoffs for the other players.

Definition 1. *A strategy $s^* \in S$ is t-immune if, for all $T \subseteq N$, with $card(T) \leq t$, all $s_T \in S_T$, and all $i \notin T$:*

$$u_i(s^*_{-T}, s_T) \geq u_i(s^*).$$

t-immune equilibria models the tolerance threshold of players, how many players can behave unpredictable without affecting the other players payoffs.

2.2. **Generative relations.** In order to compute equilibria we characterize them with adequate relations on the strategy set. Such relations are called *generative relations* of the equilibrium.

We have the quality measure:

$$Q : S \times S \to \mathbb{N},$$

where S is the set of the strategy profiles.

Let s and s^* be two strategy profiles, $s, s^* \in S$.

In this case $Q(s, s^*)$ measures the quality of strategy s with respect to the strategy s^*.

The quality Q is used to define the relation $\prec_Q$:

$$s \prec_Q s^*, \text{ if and only if } Q(s, s^*) \leq Q(s^*, s).$$

The first generative relation for the Nash equilibrium has been introduced in [7].

2.2.1. *Generative relation for t-immune strategies.* Consider a quality measure $t(s^*, s)$, which denotes the number of players who gain by switching from one strategy to the other strategies:

$$t(s^*, s) = card[i \in N - T, u_i(s_T, s^*_{-T}) \leq u_i(s^*), s_T \neq s^*_T, card(T) = t, T \subseteq N],$$

where $card[M]$ represents the cardinality of the set M.

Definition 2. *Let $s^*, s \in S$. We say the strategy s^* is better than strategy s with respect to t-immunity, and we write $s^* \prec_T s$, if the following inequality holds:*

$$t(s^*, s) < t(s, s^*).$$

Definition 3. *The strategy profile $s^* \in S$ is called t-immune non-dominated, if and only if there is no strategy $s \in S, s \neq s^*$ such that*

$$s \prec_T s^*.$$

TABLE 1. Parameter settings for t-DE

Parameter	Value
Pop size	100
Max no FFE	5000-2 players; 200000-3 players
Cr	*0.8*
F	*0.3*

The relation $\prec_T$ can be considered as the generative relation for t-immune equilibrium, i.e. the set of non-dominated strategies, with respect to $\prec_T$, induces the *t-immune strategies*.

2.3. **Evolutionary detection method.** Two evolutionary algorithms are adapted for detecting t-immune equilibrium: NSGA-II (Non-dominated Sorting Genetic Algorithm II)[9] and DE (Differential Evolution) [10].

In both algorithms the Pareto dominance relation is replaced with the generative relation $\prec_T$ and two new evolutionary methods for t-immune equilibrium detection, called t-NSGA-II, respectively t-DE are obtained.

3. Numerical experiments - the Cournot game

In the normal Cournot model [3] n companies produce q_i, $i = 1, ..., n$ quantities of a homogeneous product.

$Q = \sum_{i=1,n} q_i$ is the aggregate quantity on the market, a is a constant, and the market clearing price is $P(Q) = a - Q$ if $Q < a$ and 0 otherwise.

The total cost for company i for producing quantity q_i is $C_i(q_i) = cq_i$. The marginal cost c is constant, and $c < a$.

The payoff for the company i can described as follows:

$$u_i(q) = q_i P(Q) - C_i q_i,$$

where $q = (q_1, ..., q_n)$. The final form of the payoff function can be described as:

$$u_i(q) = q_i[a - \sum_{j=1,n} q_j - c], i = 1, ..., n.$$

In our experiments we consider $a = 50$, $c = 10$, $q_i \in [0, 10]$, $i = 1, ..., n$ and the two- and three player version of the Cournot game.

Parameter settings for t-DE are presented in Table 1. Parameter settings for t-NSGA-II are presented in Table 2. Ten different runs are considered, and the mean and standard deviation is reported.

Result for the two players variant of the Cournot game are presented in Table 3. Table 4 depicts the three player version of the Cournot game, for $t = \{1, 2\}$.

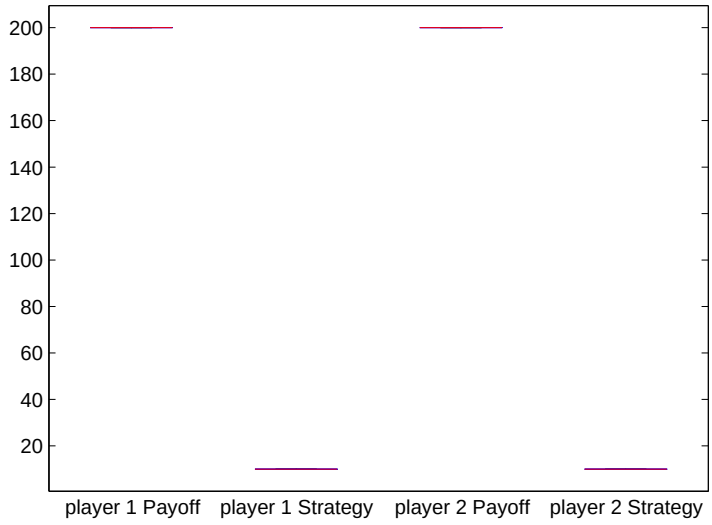


FIGURE 1. Strategies and payoffs for the two player version of the Cournot game

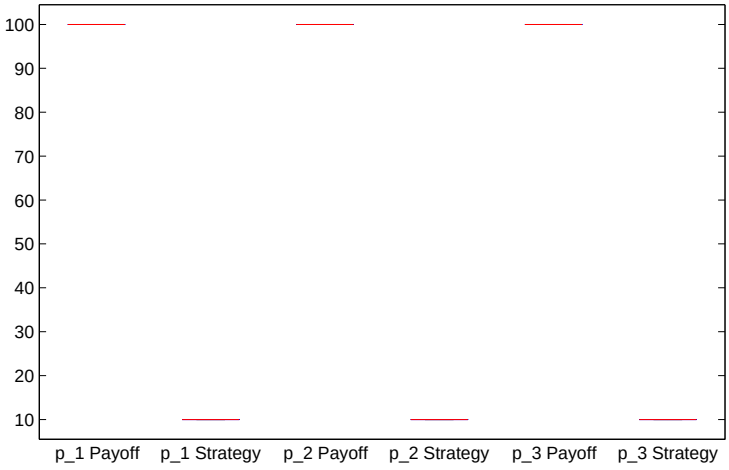


FIGURE 2. Detected 1-immune strategies and payoffs for the three player version of the Cournot game

TABLE 2. Parameter settings for t-NSGA-II

Parameter	Value
Pop size	100
Max no FFE	5000-2 players; 185108-3 players
prob. of crossover	0.2
prob. of mutation	0.2

TABLE 3. Results for 2 player Cournot game for 1-immune equilibrium (mean values for 10 different runs).

Algorithm	t_{immune}	Strategy		Payoff	
		s_1	s_2	u_1	u_2
t-DE	1	10	10	200	200
t-NSGA2	1	10	10	200	200

TABLE 4. Results for 3 player Cournot game for $t_{immune} \in \{1, 2\}$ (mean values for 10 different runs).

Algorithm	t_{immune}	Strategy			Payoff		
		s_1	s_2	s_3	u_1	u_2	u_3
t-DE	1	10	10	10	100	100	100
	2	10	10	10	100	100	100
t-NSGA2	1	10	10	10	100	100	100
	2	10	10	10	100	100	100

Figures 1, 2 and 3 depict the t-immune equilibria for two and three players.

Numerical experiments indicate that the players' threshold is minimal in both cases. The firms need to produce maximal quantity of products in order to avoid the unexpected behavior of some firms.

4. Conclusions and further work

In this study a generative relation for the t-immune equilibrium is proposed. This generative relation is used in two different evolutionary algorithms to guide the search towards desired equilibria. The Cournot oligopoly model - an economic game, is considered for numerical experiments. Results underline the stability and the potential of the proposed method.

Further experiments will focus on large games.

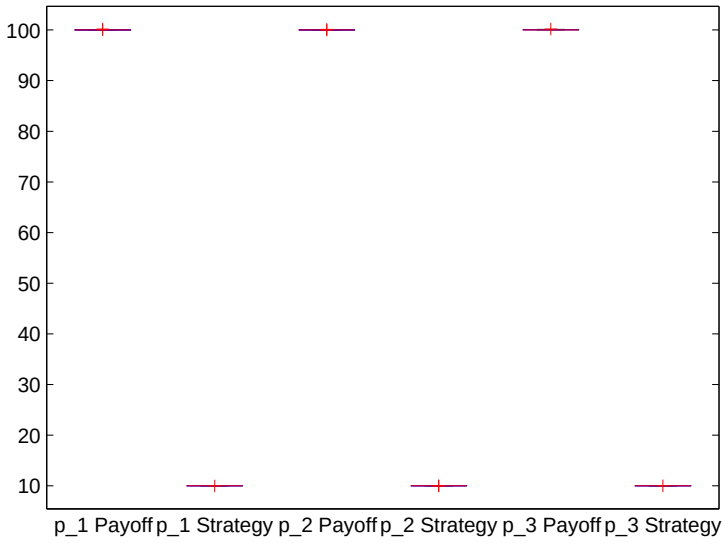


FIGURE 3. Detected 2-immune strategies and payoffs for the three player version of the Cournot game

5. ACKNOWLEDGMENT

This project was supported by the national project code TE 252 and 320 financed by the Romanian Ministry of Education and Research CNCSIS-UEFISCSU.

REFERENCES

[1] Abraham, I., Dolev, D. Gonen, R., Halpern, J.: *Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation*, Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing, 53-62, 2006.

[2] Basu, K.: *The Traveler's Dilemma: Paradoxes of Rationality in Game Theory*, American Economic Review, 84:2, 391-395, 1994.

[3] Cournot, A.: *Researches into the Mathematical Principles of the Theory of Wealth*, New York: Macmillan, 1897.

[4] Dumitrescu, D., Lung, R. I., Mihoc, T. D.: *Evolutionary Equilibria Detection in Non-cooperative Games*, EvoStar2009, Applications of Evolutionary Computing, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, vol. 5484, 253-262, 2009.

[5] Dumitrescu, D., Lung, R. I., Mihoc, T. D.: *Generative Relations for Evolutionary Equilibria Detection*, Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation, 1507-1512, 2009.

[6] Halpern, J.Y., *Beyond Nash Equilibrium: Solutions Concepts for the 21th Century*, Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing, 1-10, 2008.

[7] Lung, R. I., Dumitrescu, D.: *Computing Nash Equilibria by Means of Evolutionary Computation*, International Journal of Computers, Communications & Control, 3:364-368, 2008.
[8] Nash, J. F.: *Non-cooperative games*, Annals of Mathematics, 54:286-295, 1951.
[9] Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T.: *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6(2): 182-197, 2002.
[10] Storn, R., Price, K.V., *Differential evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, Journal of Global Optimization, 11:341-359, 1997.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: gaskonomi@cs.ubbcluj.ro

E-mail address: mihai.suciu@ubbcluj.ro

E-mail address: rodica.lung@econ.ubbcluj.ro

E-mail address: mihoc@cs.ubbcluj.ro

E-mail address: ddumitr@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

COMPUTATIONAL TOOLS FOR RISK MANAGEMENT

TUDOR DAN MIHOC[(1)], RODICA IOANA LUNG[(1)], AND D. DUMITRESCU [(1)]

Abstract. A risk management tool based on forecasting using artificial neural networks is presented in this paper. The system is composed by two modules, each containing a sparsely connected feed forward network trained using a back propagation algorithm. The method is tested on real data from the BVB stock market. The presented results emphasise the high potential of using such a system in helping investors to detect the proper moments to exit the market in order to preserve their investments.

1. Introduction

Dealing with unexpected situations on financial markets is a certitude for economists. Access to proper tools for guiding the agents in the process of decision making can make the difference between bankruptcy and salvation.

The efficient market hypothesis (EMH) stipulates that price movements on stock market are completely random and hence unpredictable, and thus making any attempt of forecasting them is impossible [5]. However the direct participants to the financial markets regard EMH with suspicion. Traditionally there are three methods for predicting the market prices: fundamental analysis, technical analysis and traditional time series forecasting.

Fundamental analysis models the market as what was initially suppose to be: an instrument for companies to attract capital by selling a share of their business. The main assumption is that shares have the value related to the investments returns and to the company value. On the long term basis, fundamental analysis is a very efficient tool, however on short term basis fails

Received by the editors: April 15, 2013.

2010 *Mathematics Subject Classification.* 68T10, 68T05.

1998 *CR Categories and Descriptors.* I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Subtopic – *I.2.6 Learning*; I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Subtopic – *I.2.8 Problem Solving, Control Methods, and Search*.

Key words and phrases. forecasting, stock market, artificial neural networks.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

to give good results. Also is a subjective view to the market therefore is difficult to construct an automatic decision support system on it.

Technical analysis is another approach referring to the methods that use informations from past transactions (prices, volumes, indices) in order to predict future. Patterns and trends in variations are detected and the forecast is made in assuming that these patterns repeat themselves in a sort of cyclic manner. As Caulson observed in [7] many methods from this category fail to have even a rational explanation for their use. However technical analysis is the most employed method in forecasting the financial markets.

The third class of methods is time series forecasting. Using statistic techniques (such as auto-regression integrated moving average or multivariate regression) a non linear function is constructed in order to model the price fluctuations. Time series is very well suited for short time forecasting only. The main disadvantage of this method is the amount of highly accurate data necessary for processing.

Artificial Neural Networks (ANN) become in recent years a useful tool in predicting markets behavior. Their property to approximate any function mapping between inputs and outputs allows (under the premises of technical analysis) an ANN to detect automatic the market's fluctuation patterns.

In this paper we present a Risk Management Tool based on feed forward ANNs (RMT-NN), trained using a back-propagation algorithm.

2. Methodology

Our aim is to give a fair and accurate prediction of the main index on Bucharest stock exchange market (BVB). The market agents will be able then to take more informed decisions when they need to identify the proper moment to exit the market. The system was implemented using the widely used *Fast Artificial Neural Network Library (FANN)* – a C library that supports sparsely connected networks [4].

The system is fetching the daily value for the stock indexes from an outside server at the end of a trading day, is "learning" the new data, and makes a new prediction for the next period.

The core of the application has two modules each containing a ANN with the same architecture, same input but different outputs.

2.1. **Input selection.**

We consider for this model a selection of normalized data from BVB indexes [3]:

i BET – Bucharest Exchange Trading index (a free float weighted capitalization index of the most liquid 10 companies listed on the BVB regulated market),

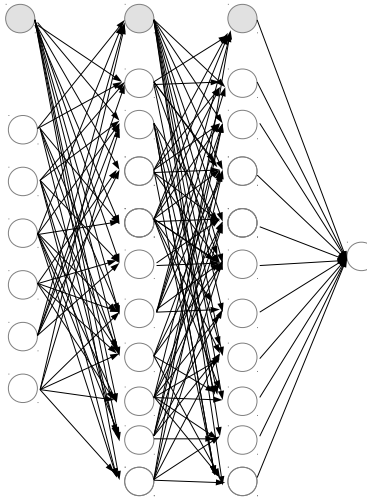


FIGURE 1. The adopted model for each of RMT-NN's modules for this study: 7 inputs nodes, two hidden layers each of 11 nodes and the exit layer with one node

ii ROTX – Romanian Traded index (a free float weighted capitalization index that reflects in real time the price movement of "blue chip" companies traded on the BVB market),
iii BETC – Bucharest Exchange Trading - Composite Index (reflects the price movement of all the companies listed on the BVB regulated market),

and a selection of averages on the past 5 or 20 days of these indexes:

iv the average for the last 5 days for the BET index,
v the average for the last 5 days for the ROTX index,
vi the average for the last 20 days for the BET index.

2.2. **Output.** Each module of the RMT-NN will give out a single output representing the predicted BET index for the next day (the first module) and a prediction of the average BET index for the next 10 days.

2.3. **Model.** In order to build an efficient ANN capable to inform the financial market agent about the trends a prediction model is proposed. We will consider valid the main hypothesis of technical analysis: patterns and trends repeat themselves within the market fluctuations.

The selected model for solving this problem was a fixed sparse 7:11:11:1 structure with the connective density 0.6.

	MSE (first module)	MSE (second module)
Average	0.000110	0.000726
St. Dev	0.000537	0.002100

TABLE 1. Average and Standard deviation of the Mean Square Errors for each module of the RMT-NN system.

A back propagation algorithm was used with delta rule and the sigmoid function selected for the training. The default learning rate was 0.7 and a momentum of 0.1 was imposed.

Each new training was done on up to 20 days old information (approximative the length of a working month). For one day data there were up to $5 * 10^6$ epochs or until the error was below 10^{-6} – error that would correspond to less than 0.1% error from the real value for that day.

2.4. **Dataset.** The data set used for training and testing the developed model is composed with pre-processed informations from the Romanian stock market from the period 1/1/2009 to 31/12/2009.

3. Results

The presented results are from a 90 days period. In Table 1 the average and the standard deviation is presented for the Mean Square Errors for each module. The worst error on the test period was up to 10% from the correct value. However that occurred in the first 10 days since the system begun the prediction, so it didn't have time learn the market's pattern. In Figures 2 and 3 we can see the poor prediction values in comparison with the real ones for beginning of the test period and how the predictions become more and more accurate as time passes.

3.1. **Discussion.** The implemented ANN shows that a system is able to predict with a high degree of accuracy the BVB markets movements. The natural question that rises, as always with such predicting systems, is: Can this be a real decision support? Can someone make profits using such systems? Several researchers conclude in their works that such method to transaction would bring profit only if there are no commissions for the buying and selling process (an unrealistic assumption). Considering that Romanian brokers have commissions sometimes even up to 2% such instrument for forecasting would be useless for market gamblers. However our aim was to build a risk management tool based on forecasting and not a buy/sell transaction alert system.

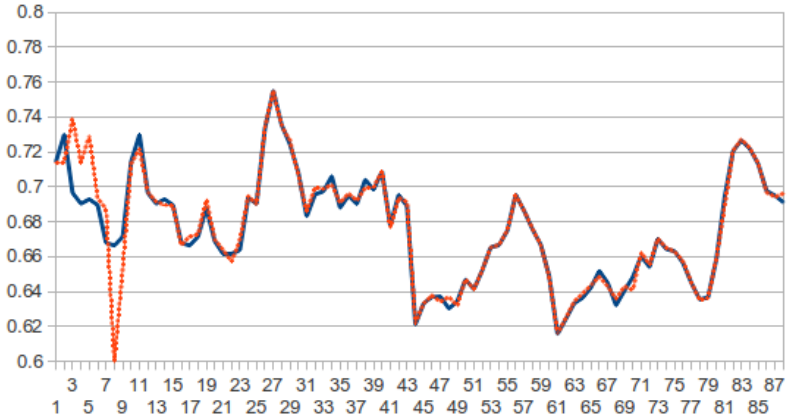


FIGURE 2. Normalized BET index (full line) and predicted values (dot line) of RMT-NN system's first module for the test and training period.

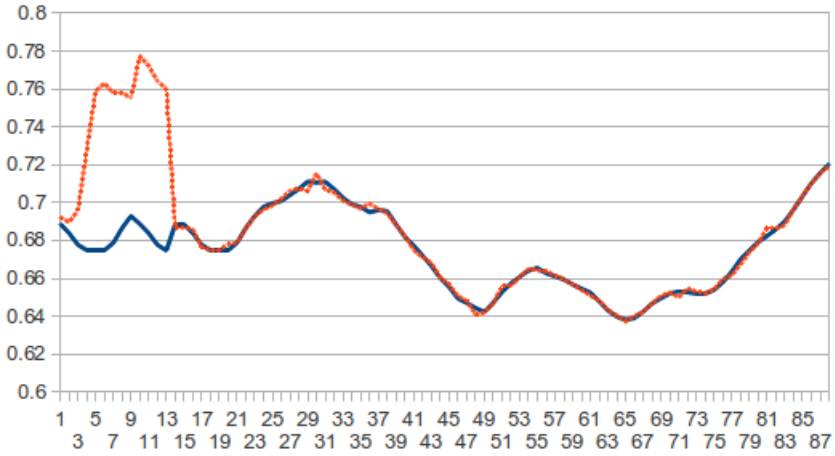


FIGURE 3. Average of normalized BET index (full line) and the predicted values for the average of normalized BET index (dot line) of RMT-NN system's second module for the test and training period.

4. CONCLUSION

In this paper we present a support system for BVB trade agents. The system provides forecasting assisting market players to decide when to bail out and exit the market with the least possible losses .

The system is composed from two modules, each one containing a sparsely connected artificial neural network. Both ANNs were trained using a standard propagation algorithm.

The system was trained and tested with the day-by-day values from pre-processed informations from the BVB.

A comparison between the predicted values and the real ones for the test period underline the high potential of the method. A fairly good prediction can be reached without an extensive market knowledge.

Further work will consist in combining this model with integrated pitch-fork analysis by adding several pre-process modules that can identify specific patterns on the market in order to increase the accuracy of the predictions.

References

[1] Brabazon, A., and O'Neill, M.*Biologically Inspired Algorithms for Financial Modelling*,Series: Natural Computing Series, XVI, 2006.

[2] P.. M. Tsang, Paul Kwok, S.O. Choy, Reggie Kwan, S.C. Ng, Jacky Mak, Jonathan Tsang, Kai Koong, Tak-Lam Wong, *Design and implementation of NN5 for Hong Kong stock price forecasting*, Engineering Applications of Artificial Intelligence, Vol. 20, No. 4 (June 2007), pp. 453-461.

[3] Pop, C., and Dumbrava, P., *Bucharet Stock Exchange Evolution November 1995 – November 2005.* Interdisciplinary Research Management, 2, 349-367, 2006.

[4] Nissen, Steffen, *Implementation of a fast artificial neural network library (fann). Report*, Department of Computer Science University of Copenhagen (DIKU), 2003, 31.

[5] Eugene F. Fama. *The Behavior of Stock-Market Prices*, The Journal of Business, Vol. 38, No. 1. (Jan., 1965), pp. 34-105

[6] Frank, R. J., Neil Davey, and S. P. Hunt. *Time series prediction and neural networks.* Journal of Intelligent & Robotic Systems 31.1 (2001): 91-103.

[7] Coulson, D. Robert. The intelligent investor's guide to profiting from stock market inefficiencies. Probus, 1987.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: mihoct@cs.ubbcluj.ro

E-mail address: rodica.lung@econ.ubbcluj.ro

E-mail address: ddumitr@cs.ubbcluj.ro