Proceedings of the International Workshop on Artificial Intelligence and Cognition

AIC 2015

Torino, Italy, September 28-29, 2015

(Eds.)

Antonio Lieto, Cristina Battaglino, Daniele P. Radicioni, Manuela Sanguinetti

Editors' address:
Università degli Studi di Torino
Dipartimento di Informatica
Corso Svizzera, 185
10149 Torino, Italy
lieto.antonio@gmail.com
battagli@di.unito.it
radicion@di.unito.it
msanguin@di.unito.it

ICAR CNR - Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni
Edificio 11, Viale delle Scienze
90128 Palermo, Italy
lieto.antonio@gmail.com

Preface

This book of Proceedings contains the accepted papers of the AIC 2015 - the International Workshop on Artificial Intelligence and Cognition, held in Turin (Italy) on September 28th and 29th, 2015. AIC workshops aim at fostering the collaboration between the researchers coming from the fields of computer science, philosophy, engineering, psychology, neurosciences etc. and working at the intersection of the Cognitive Science and Artificial Intelligence (AI) communities.

AIC workshops (Lieto and Cruciani 2013; Lieto, Radicioni and Cruciani, 2014) have produced, in the past years, a recognized level of discussion in Europe on the crossborder themes between AI and Cognitive Science and selected and expanded versions of their scientific papers have been published in dedicated special issues on international journals such as Connection Science and Cognitive Systems Research (edited by Lieto and Cruciani (2015), and Lieto and Radicioni (2016), respectively).

AIC 2015 has been made possible thanks to the "Fondazione Ricerca e Talenti" (http:// www.ricercaetalenti.it) of the University of Turin that has fully sponsored the whole event. We would like to thank them for their financial and organizational support. Also we are grateful to the members of the Scientific Program Committee for their valuable work. Finally, thanks should be given to our wonderful student volunteers for their help in many practical issues.

In this workshop proceedings appear 2 abstracts of the talks provided by the keynote speakers Aldo Gangemi and Amanda J. Sharkey and 13 peer reviewed papers accepted by the Program Committee Members through a process of peer-review.

Specifically the 13 papers were selected out of 21 submissions coming from researchers of 16 different countries from all the continents.

In the following, a short introduction to the content of the volume is presented.

In the paper "Cognitive Programming", by Loizos Michael, Antonis Kakas, Rob Miller and Gyorgy Turan, the authors point out some foundational issues regarding the design of cognitive systems and propose a novel methodological approach for human-computer interaction based on what they call "cognitive programming" paradigm.

In the paper "Towards a Visual Remote Associates Test and its Computational Solver", by Ana-Maria Olteteanu, Bibek Gautam and Zoe Falomir, the authors describe a computational solver for a visual version of the Remote Associate Test (RAT, a test used for measuring creativity in humans) and present the result of an evaluation done w.r.t. human responses.

The paper "Modeling the Creation and Development of Cause-Effect Pairs for Explanation Generation in a Cognitive Architecture", by John Licato, Nick Marton, Ron Sun and Selmer Bringsjord presents the rationale for modelling the learning of causeeffects explanations in the CLARION cognitive architecture by using, as reference point, a Piaget's experiment introduced to understand how children generate explanations.

The paper "A cognitive view of relevant implications", by Claudio Masolo and Daniele Porello, presents an interesting link between Relevance Logic and Conceptual spaces and It provides a cognitive view and formalization of relevance implication. In the paper "Information-Theoretic Segmentation of Natural Language", by Sascha Griffiths, Mariano Mora McGinity, Jamie Forth, Matthew Purver and Geraint A. Wiggins., the authors extend to natural language a statistical model originally devised in the domain of music perception and cognition; in particular, the authors adopt a statistical (information-theoretic) learning approach on sequential data.

In the paper "Pattern Recognition: A Foundational Approach" by Agnese Augello, Salvatore Gaglio, Gianluigi Oliveri and Giovanni Pilato, the authors discuss some foundational issues regarding the "patterns problem" and propose a three layer architectures as a suitable solution for pattern understanding.

In the paper "World Modeling for Tabletop Object Construction", by Arda Inceoglu, Melodi Deniz Ozturk, Mustafa Ersen and Sanem Sariel the authors dicuss the problem of scene recognition in a robotic environment and propose a framework not relying only on perceptual factors but also relying on a knowledge updating process for their scene recognition approach.

The paper “A Network-based Communication Platform for a Cognitive Computer”, by Mostafa W. Numan, Jesse Frost, Braden J. Phillips and Michael Liebelt, presents a novel hardware-based approach for the design of cognitive computer based on an energy efficient approach for computation with a parallel production system. In the paper “Developing Fuzzy Cognitive Maps with Self Organizing Maps”, by Marcel Wehrle, Edy Portmann, Alex Denzler and Andreas Meier, the propose the combination of SOM and FCM in retrieving the semantic structure of web documents.

In the paper “Property-based semantic similarity: what counts the most?”, by Silvia Likavec and Federica Cena, the authors discuss the problem of conceptual similarity in ontologies by exploiting the Tversky-distance and by pointing out the importance of weighting the features (the object properties in ontological terms), the values filling such features and the importance of the hierarchy of values.

In the paper “Do the self-knowing machines dream of knowing their factivity?”, by Pierluigi Graziani, Alessandro Aldini and Vincenzo Fano, the authors the authors present a formal account of the ”GÃ¶delian Argument”, according to which the human mind would be equivalent to a finite machine unable to understand its own functioning.

The paper “Extracting Concrete Entities through Spatial Relations”, by Olga Lidia Acosta Lopez and C. Antonio Aguilar, the authors describe system able to bootstrap the recognition of concrete entities from medical domain texts by taking advantage of the use of the expression of spatial relationships.

Finally, the paper “A Framework for Uncertainty-Aware Visual Analytics in Big Data”, by Amin Karami, proposes a framework combining Fuzzy SOM (self organising maps) within the MapReduce framework to model uncertainty and knowledge visually within big data sets.

November 25, 2015
Torino

The AIC 2015 Chairs
`http://www.di.unito.it/~lieto/AIC2015/program_committee.html`

Sponsoring Institution Message

This publication collects the papers selected for AIC2015 - International Workshop on Artificial Intelligence and Cognition (Turin 28th-29th September 2015), an event organised with the support of Fondazione Fondo Ricerca e Talenti.

Fondazione Fondo Ricerca e Talenti is one of the first university foundations in Italy, and the first to apply innovative fundraising mechanisms to research activities. Its aim is twofold:

- promoting fundraising activities for the University of Turin, to which the Foundation belongs;
- financing scholarships and supporting scientific dissemination activities, for the benefit of young researchers of our University.

We firmly and concretely believe in the importance of research. We know, as many do, that research is the foundation of our competitiveness, of our health, of our capacity to deal with social and cultural challenges, of our future.

We also believe in our researchers, as much as in research. We know that their ideas need an opportunity to grow and show their potential. Our strive to provide such opportunity - even a small, but real opportunity - is at the core of our mission.

For us, doing so means three very simple things: reward merit, be inclusive and engaging, do not hesitate to think out of the box, stick to our vision and keep an eye on our future. These principles allowed us in two years, with very few human and financial resources, to sponsor dozens of bursaries and dissemination events, to create a network of hundreds of voluntaries supporting our initiatives on the field and to have excellent echo on the media and at institutional level (including the European Commission).

We want to build ties with students, with the civil society and with the private sector in order to make the University of Turin a forge of opportunities at the service of our youth and of our territories as a whole.

This is the reason why we particularly welcome spin-off initiatives like this publication, which contributes to further develop and disseminate research ideas stemming from our financed seminars on cutting-edge matters like Artificial Intelligence and Cognition.

In line with our spirit, we hope that this publication will highly benefit the scientific community and will have a positive impact on they way we - as Human Intelligences and Cognitive Beings - understand and live this complex world.

Fondazione Fondo Ricerca e Talenti
The President
Gianmaria Ajani

AIC 2015 Programme Commmittee

Gabriella Airenti, University of Torino, Italy
Bruno G. Bara, University of Torino, Italy
John Bateman, University of Bremen, Germany
Tarek R. Besold, University of OsnabrÃŒck, Germany
Mehul Bhatt, University of Bremen, Germany
Erik Cambria, Nanyang Technological University, Singapore
Angelo Cangelosi, University of Plymouth, UK
Cristiano Castelfranchi, ISTC-CNR, Italy
Antonio Chella, University of Palermo, Italy
David Danks, Carnegie Mellon University, USA
Mark A. Finlayson, Florida International University, USA
Christian Freksa, University of Bremen, Germany
Marcello Frixione, University of Genova, Italy
John Fox, University of Oxford, UK
Savatore Gaglio, University of Palermo and ICAR-CNR, Italy
Nicola Guarino, LOA-ISTC CNR, Italy
Anna Jordanous, University of Kent, UK
Antonis K. Kakas, University of Cyprus, Cyprus
Ismo Koponen, University of Helsinki, Finland
Oliver Kutz, University of Bolzano, Italy
Othalia Larue, Wright State University, USA
Francesca A. Lisi, University of Bari, Italy
Vincenzo Lombardo, University of Torino, Italy
Diego Marconi, University of Torino, Italy
Orazio Miglino, University of Napoli Federico II, Italy
Vincent C. MÃŒller, Anatolia College, Greece
Alessandro Oltramari, Carnegie Mellon University, USA
Alessio Plebe, University of Messina, Italy
Paul Rosenbloom, University of Southern California, USA
Alexei Samsonovich, George Mason University, USA
Giovanni Semeraro, University of Bari, Italy
Ron Sun, Rensselaer Polytechnic Institute, Troy, NY, USA
Guglielmo Tamburrini, University of Napoli Federico II, Italy
Fabio Massimo Zanzotto, University of Roma 'Tor Vergata', Italy

Student Volunteers

Eleonora Ceccaldi, Francesco Ghigo, Enrico Mensa

Contents

Invited Talks (Abstracts)

Long and Short Papers

How Many (Polymorphic) Frames? Classic KR in the World Wide Web

Aldo Gangemi

LIPN University Paris13-CNRS-Sorbonne France, ISTC-CNR Rome Italy

Abstract. The good old notion of frame, as introduced in the seventies, is a natural hub for cognitive sciences, knowledge representation, and natural language understanding. Developments in the last ten years, mainly due to the socio-technical ontology of the Web, have apparently changed the nature and scale of the notion, and pose new challenges. I will present a bird's eye view of the current situation, with a quick glance at human-robot interfaces.

Robot Ethics: Illusions, Challenges and Rewards

Amanda J. Sharkey

University of Sheffield, UK

Abstract. In this talk, I introduce the topic of Robot Ethics: pointing out the growing use of robots in social roles, and the need to identify and consider the ethical issues involved before they are too well established. From ancient times to the present day, robotics has depended on creating the illusion of life. This illusion is exploited in the development of robot companions, and is helped by the human tendency to be anthropomorphic and to behave as though robots were able to understand and respond to them. The risks posed by the development of robot companions for older people, and for children were considered. The main risks of robot companions for older people were identified as being: loss of human contact, loss of dignity, deception, loss of privacy, and loss of autonomy. However, for some people, such as those with dementia, robot companions such as the Paro robot seal can result in health and well being benefits. There are a related set of ethical concerns about the introduction of robots as companions, or teachers of children. These include those related to human autonomy, attachment, deception, privacy and social learning. A brief review of underlying ethical theories is provided that included the Capability Approach. It is concluded that despite the serious ethical concerns raised by the idea of robot companions, there are some circumstances where robots could be used to improve the lives of vulnerable older people, or children with special needs, by increasing their access to some of the capabilities that make life worth living.

Cognitive Programming

Loizos Michael[1], Antonis Kakas[2], Rob Miller[3], and Gyorgy Turán[4]

[1] Open University of Cyprus, loizos@ouc.ac.cy
[2] University of Cyprus, antonis@ucy.ac.cy
[3] University College London, r.s.miller@ucl.ac.uk
[4] University of Illinois at Chicago and MTA-SZTE Research Group on Artificial Intelligence, gyt@uic.edu

Abstract. The widespread access to computing-enabled devices and the World Wide Web has, in a sense, liberated the ordinary user from reliance on technically-savvy experts. To complete this emancipation, a new way of interacting with, and controlling the behavior of, computing-enabled devices is needed. This position paper argues for the adoption of *cognitive programming* as the paradigm for this user-machine interaction, whereby the machine is no longer viewed as a tool at the disposal of the user, but as an assistant capable of being supervised and guided by the user in a natural and continual manner, and able to acquire and employ common sense to help the user in the completion of everyday tasks. We argue that despite the many challenges that the proposed paradigm presents, recent advances in several key areas of Artificial Intelligence, along with lessons learned from work in Psychology, give reasons for optimism.

1 An Emerging Need and The Overall Challenge

Today's huge market pressure for the use of smart systems by everyone and in every aspect of their daily life is forcing Artificial Intelligence (AI) to stand up and deliver. What was perhaps thought out of reach in the past needs to become a reality to satisfy the ever increasing desire of humans to use their new machines — computer devices linked with the Internet — in their everyday activities.

Unlike anything we have seen to date, this new vision of user-machine interaction will allow ordinary users without technical background to instruct or program their devices in a natural and personalized manner, and will allow the devices to assist (and enhance the abilities of) their users in dealing with everyday tasks. This *symbiotic* relation splits the burden of communication among the user and the device, offering a "programming paradigm for the masses", avoiding the extremes of using natural languages that are too complex for ordinary devices, or programming languages that are too complex for ordinary users.

Early examples of such interactions already exist, ranging from the personal assistant softwares provided by major smart-device manufacturers, to the (expected) applications for expert analysis of problems in specialized domains built on top of the Watson engine. But perhaps the clearest example of this emerging form of interaction, which we shall call ***cognitive programming***, is that of searching for information on the World Wide Web. The use of web search engines constitutes a form of programming exercised by billions, independently of technical ability, through a programming language of keywords in natural language, in a manner compatible with the cognitive abilities of humans. Through their searches, users gradually develop a sense of how to improve the way they program or instruct the search engine with queries that achieve the users' intended aim. On the other side, search engines capture the preferences or typical behaviors of users, to help propose search queries or choose how to rank results.

We will refer to systems interacting with users through cognitive programming as ***cognitive systems***, as these systems are, in spirit at least, of the same kind as the cognitive systems proposed relatively recently in several works in AI; see, for example, the new journal of Advances in Cognitive Systems, the journal of Cognitive Systems Research, and works such as [26–28, 50].

Unlike work in existing autonomous agents / systems, we think of a cognitive system as having an operational behavior similar or parallel with that of a human personal assistant. Its domain of application is limited to certain common everyday tasks, and its operation revolves around its interaction with its user in a manner that is compatible with the cognitive reasoning capabilities of the latter. To understand (and correct when needed) the reasoning process of the system, the user expects the system to use *common sense* to fill-in important relevant information that the user leaves unspecified, and to be able to keep learning about the domain and the user's personal preferences through their interaction.

The goal for building systems that are *cognitively compatible with humans* ultimately imposes a set of considerations on cognitive programming, as this determines the communication channel between the user and the system. The overall challenge of developing the proposed paradigm of cognitive programming ultimately rests on fleshing out and addressing these considerations:

- Cognitive programming should be a process akin to human-human communication. The need for detailed operational instructions should be minimized.
- There should be a level of interaction between the user and the system where the two understand and can anticipate the behavior of each other.
- Cognitive compatibility with the user should be accommodated by acknowledging the central role that natural language has in human communication, and in the way humans store, retrieve, and use commonsense knowledge.
- Cognitive programs should develop incrementally to meet the aims of the user through an open-ended process. Cognitive systems should be able to learn, and be able to improve from their past interaction with the user.
- Cognitive programs should be robust, never failing, but continuously improving / completing their ability to offer personalized solutions to the user, while adapting to a possibly new or changing user position, stance, or profile.

The emphasis of this position paper is on describing the desirable characteristics and the technical challenges resulting from the aforementioned considerations. It examines the salient and foundational issues that need to be considered, and offers possible suggestions for a first version of a cognitive programming language. This proposal is grounded in our recent experience of trying to automate the cognitive task of story comprehension,[5] and on the comparison of the resulting psychologically-informed approach with earlier work in AI for addressing other types of scientifically-oriented problems, such as problems of diagnosis and planning that span beyond the ordinary capabilities of human intelligence.

1.1 Scientific Position for Cognitive Programming

The scientific position underlying our approach and proposal for cognitive programming is that *symbolic AI* can offer the tools needed for the aforementioned considerations, as long as one *abandons the traditional view* of the role of logic for reasoning, and one is strongly guided by work in Cognitive Psychology. To a certain extent, then, this position takes us back to the early days of AI.

We embrace McDermott's view in his paper "A critique of pure reason" [31], that developing a logical theory alone — even a non-monotonic one — without consideration of the reasoning process can not lead to human commonsense intelligence. A vast amount of empirical work from Psychology (see, e.g., [12]) shows that commonsense inferencing has a looser form than that of scientific reasoning, and that the conventional structure and form of logical reasoning, as epitomized by mathematical or classical logic, is not appropriate. Given strong evidence from recent work in Psychology (see, e.g., [33]) in support of an argumentation-based theory for human reasoning, we adopt a form of argumentation as the basis for a cognitive system's reasoning process. Drawing from work in Cognitive Psychology (see, e.g., [13, 21, 23, 44]) on how human knowledge is (or might be) structured and used, we base our approach on the cognitive process of *comprehension*, within which logical inference is only one component.

Although work in logic-based AI may accept, to a certain extent, the need to deviate from strict logical reasoning (e.g., non-monotonicity, belief revision, logic programming), efforts to automate reasoning still typically proceed on the basis of developing proof procedures that are sound and complete against some underlying semantics of "ideal inferences". Unlike such work, on which cognitive programming may be based and from which it may be guided, cognitive programming shifts the emphasis from deep and elaborated reasoning to richly structured knowledge, assuming that commonsense intelligence resides in the "complexity of knowledge representation" rather than the "complexity of thought". As in many cases of Computer Science, data structures and data organizations matter and can make all the difference in having an effective and viable solution.

2 Computational Model and System Architecture

The central notion underlying the computation of a cognitive system is that of ***comprehension***, a notion adopted from story or narrative text comprehension in Cognitive Psychology (see, e.g., [22]). In our setting, comprehension proceeds

[5] The system *STAR: Story Comprehension through Argumentation*, along with benchmark stories and other material, is available at: http://cognition.ouc.ac.cy/narrative/

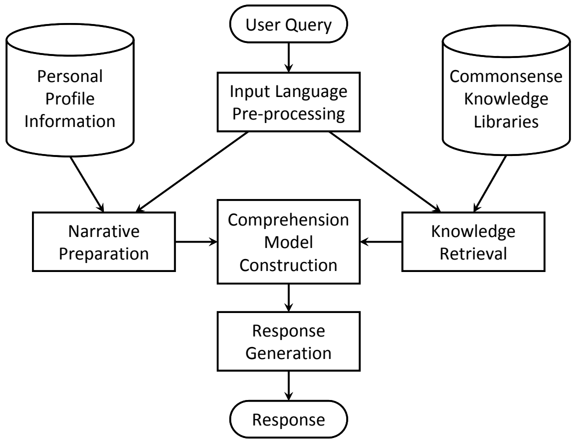


Fig. 1. General Architecture of a Cognitive System

by first combining the explicit input given by the user with information that is available in the user's profile (i.e., personal facts), forming an ***input narrative*** of the task at hand. This narrative is then synthesized with information that the system has about the domain (i.e., commonsense knowledge) and its user (i.e., personal preferences), leading to the construction of a ***comprehension model***.

A comprehension model is an elaboration of the input narrative with new information, or ***inferences***, capturing the (or a possible) implicit meaning or intention of the narrative. Critically, the comprehension model is ***coherent***, and includes only inferences that are important for successful understanding, while omitting cluttering details and speculations. If, for example, a user enquires for "private celebration of wedding anniversary", it is essential for the comprehension model to include the inference "place for two people", but not the side inference "married for at least one year" or the mere possibility "dinner at fancy restaurant".

The *central hypothesis* of our proposed cognitive programming framework is, then, that the availability of a comprehension model allows the system to better act and assist its user in the requested task. The general high-level architecture of cognitive systems that follows from this hypothesis is depicted in Figure 1.

We shall analyze the various components of this architecture in subsequent sections. For now, we shall discuss the interaction of the user with the system.

2.1 Cognitive Programming Interaction Modes

The most basic form of user-machine interaction is ***querying***, whereby the user, or the cognitive assistant of some other user, or even some sensor device, inputs a specific request or query to the cognitive system. The system then identifies or compiles relevant commonsense knowledge, perhaps even invoking a process of online learning, and responds with some action (e.g., a suggestion of whether to accept or not an offer) that would help in addressing the task that has prompted the query. When an output is thus produced by the cognitive system, another form of interaction, that of ***supervising***, allows the user to give feedback to the system on the appropriateness of its output. For example, a user may override the suggestion or decision of a cognitive assistant with or without an explanation. The overridden output is then treated as training data for the system to learn (better) the user's personal opinion or preference on the particular case at hand.

Independently of any given query, the user may interact by ***personalizing*** the cognitive system through general statements about the user's preferences, such as "I like to spend the evenings with my family" or "Family is more important than work for me". The system responds by transforming such statements in an appropriate internal language, and recording them in the user's profile, which, in turn, personalizes other aspects of the user's interaction with the system.

In the context of a particular domain of application or discourse, interaction through ***guiding*** allows the user to offer general information that would aid the cognitive system to understand the salient aspects of the domain. Such information is also provided indirectly when, for instance, the user interacts with the system in any of the preceding ways. No matter how information is provided, guiding initiates a process to recognize concepts that are relevant and important for the user. In turn, this information can be used to prepare relevant knowledge on these concepts, by directing a background process of offline or batch learning of general commonsense knowledge that is related to the particular domain.

In what is arguably the lowest (i.e., closest to the machine, and analogous to the use of traditional programming languages) level of interaction, ***instructing*** allows the user to input particular pieces of knowledge to the cognitive system on how to operate or react under very specific circumstances. Such inputs are expressed in the system's internal language, and can be imputed directly in the user's personal profile or personalized knowledge libraries. We do not envisage that this would be the prevalent way of user interaction with cognitive systems.

2.2 Illustrative Example of a Cognitive System

Suppose that Bob wishes to manage his evening work appointments with the assistance of a cognitive system. He cognitively programs the system by guiding it with domain-specific information like "dinner plans, family time, work appointments, dietary constraints", prompting the system to gather relevant commonsense knowledge. Bob further personalizes the system with facts, such as "Bob is vegetarian", and preferences, such as "I like to spend evenings at home", "Customers from abroad are very important", and "I should never miss my children's birthday parties". Some of this latter type of information might have also been learned by the system by finding regularities in Bob's past queries to the system (e.g., if Bob often specified the keyword "vegetarian" in past queries), or through supervision of past proposed suggestions by the system (e.g., if Bob often declined suggestions by the system for late dinner outside his house).

When Bob's cognitive system receives a request from Bob's immediate boss, John, for "Working dinner today with John", the system combines this input with facts in Bob's profile or other current information the system has from sensors, calendars, etc., to construct an expanded *input narrative*. This narrative is then comprehended through the use of the system's commonsense libraries, and the comprehension model is used to decide on whether the request is to be accepted.

If no additional information is given to the cognitive system, the system will reject the request, since having dinner with John would mean going to a restaurant that evening, which would conflict with Bob's preference to be at home in the evenings. Such inferences would be supported by commonsense knowledge of the form "Normally, working dinners are at restaurants", and "Normally, dinner is in the evening". In a more advanced case the system could generate alternative suggestions, such as to have dinner with John at home that evening. The request would also be rejected if the system were to receive from the calendar the information that "Today is the wedding anniversary of Bob", giving an additional reason for Bob's inability to have dinner with John, since "Normally, a wedding anniversary is celebrated privately"; this piece of common sense supporting the decision could be offered as an explanation of the system's response.

If (possibly after the initial rejection of the request) additional information is given that "John will be accompanied by important customers from abroad", this new piece of the *story* will be incorporated in the input narrative, leading to a *revision* of the comprehension model, and to the retraction of the system's earlier decision, as now the request is supported by Bob's preferences. The system would then suggest to accept the request, and perhaps reschedule the celebration of the wedding anniversary for another evening. Had further additional information been available that "Bob's son is having a birthday party tonight", a further revision would have been caused that would again reject the request, but possibly suggesting an alternative plan through the use of commonsense knowledge such as "Normally, a pre-dinner drink (and an apology) is an alternative to dinner".

3 Foundations of Cognitive Programming

What is an appropriate theoretical model of computation and semantics of programming that would underlie the development of the cognitive programming paradigm? What is the form of the internal language of the cognitive system, which would support the *computational cognitive metaphor* of story or narrative text comprehension as the central form of program execution? This internal language ultimately determines the *form of representation* of knowledge used by the cognitive system. Adopting a symbolic representation raises several questions: What is an appropriate logic and form of reasoning? Is logic alone sufficient to capture the cognitive requirements, such as that of a natural language user-interface and a computational model of comprehension? If not, what are the cognitive elements that would need to accompany a logical approach?

We turn again to Cognitive Psychology (see, e.g., [16, 21, 53]) for guidance:

- Knowledge is composed of *loose associations* between concepts, that, unlike logic rules, are stronger or weaker depending on the context.

– Reasoning gives rise to a *single comprehension model*, avoiding the cognitively expensive task of considering possible non-deterministic choices.
– Reasoning proceeds *lazily* by drawing only inferences that are *grounded* directly on the explicit concepts given in the narrative, in an *incremental* manner as parts of the narrative become available. When conflicting information is encountered, the comprehension model is suitably *revised* [43].
– *Cognitive economy* — necessitated by human cognitive limitations, which are bound to appear also in cognitive systems with massive knowledge libraries — is achieved by requiring the comprehension model to be *coherent*, including inferences that are tightly interconnected, and excluding inferences (even undisputed ones) that are peripheral to the understanding of the given narrative [1, 15, 32, 49], or to the completion of another cognitive task [45].

The above guidelines leave, nonetheless, several key issues on the treatment of knowledge unanswered. Below we elaborate on two of those: a more detailed view of knowledge representation, and the process of knowledge acquisition.

3.1 Representation of Cognitive Programs

In constructing the comprehension model, the cognitive system needs to *retrieve relevant commonsense knowledge* and possibly to *adapt* this to the narrative (and hence to the particular query and task) at hand for subsequent reasoning. This imposes two desired properties for knowledge representation that seem at odds with each other: knowledge should be represented in a fashion sufficiently flexible to be easily accessible and adaptable (e.g., in terms of the vocabulary and syntax being used), but at the same time knowledge should be represented in a fashion sufficiently concrete to be amenable to symbolic reasoning. We refer to this problem of representation as ***the challenge of knowledge plasticity***.

A way to address this challenge might be the adoption of *multiple representations* for the internal language of the cognitive system, and hence, of the commonsense knowledge that the system handles. Representations can exist, for instance, to capture a general categorization of the knowledge, typical or exemplar entities and situations, detailed knowledge for specific cases, etc. Perhaps the system's commonsense knowledge is represented at a more general and abstract level when it is initially acquired through offline or batch learning. When queries are provided by the user, a form of ***knowledge compilation*** might turn the relevant general knowledge into a task-specific form that can be directly used to link the knowledge with the input query (and resulting narrative) for reasoning.

How the knowledge is structured in such levels and how a user input is compiled down these levels to the specific one on which the execution / reasoning occurs presents one of the central challenges for cognitive programming. We posit that an *argumentation perspective* might be useful in capturing the important aspects of the most specific of these levels, where knowledge is already compiled into a form appropriate for formal reasoning. This representation framework falls under the general scheme of abstract argumentation frameworks [11] that have been used to formalize and study several problems in AI (see, e.g., [3, 4]), including story comprehension [5, 9], and natural language interpretation [6]. Abstract argumentation will need to be suitably relaxed and adapted to reflect the cognitive requirements that we have set for cognitive systems (see, e.g., [39]).

Based on our work on story comprehension [9] and our attempts to develop a cognitive programming language for that task [10], we offer below some pointers on what a cognitively-guided argumentation framework might look like.

Arguments are built via simple association rules, each comprising a small set of concepts as its premise and a single concept as the conclusion that is supported or promoted (but not necessarily logically entailed) when the premise holds. In relation to the example discussed in Section 2.2, a relevant association rule would be "$\{\mathsf{dinner\_at(Person,Place)},\ \mathsf{with\_boss(Person)}\} \rightsquigarrow \mathsf{restaurant(Place)}$", capturing the argument that having dinner with one's boss normally happens at a restaurant. We view such association rules not as components of scientific theories (e.g., of causality, of norms and obligations, of the mind), relying on elaborative and careful reasoning, but rather as *phenomenological* manifestations of the inferences that would follow from such theories, via a "flat" representation.

Even so, not all association rules can be applied in parallel. Different association rules may promote conflicting conclusions, not all of which can be included in a comprehension model. Resolving conflicts is the essence of the argumentative stance we employ. We adopt the view that association rules are annotated to denote their (possibly relative) level of strength, so that when in conflict, these strengths ensure that the stronger rules will draw inferences, effectively qualifying (by offering a strong counter-argument to) the use of the weaker rules.

With the addition of a *time* dimension, such association rules are sufficiently expressive to represent causality. Thus, if we mark the conclusion of an association rule as holding *temporally after* the premise, the conclusion could correspond to the effect that is brought about when the premise holds. Such causal links are known from Psychology to be important in ascertaining the coherence of a comprehension model. Analogously, if we mark the conclusion of an association rule as holding *temporally before* the premise, the conclusion could correspond to an explanation of why the premise came to be. Drawing such explanatory inferences (when justified to do so) is again critical in the process of comprehension.

Such aspects of causality in world knowledge have featured prominently in the foundations of Artificial Intelligence (cf. the Situation Calculus [30], the Event Calculus [25], and several action languages [14, 19, 29, 46]). The central problems of *frame*, *ramification*, and *qualification* will need to be addressed within the cognitive programming framework, but only in a simplified and qualitative form, as it suffices for our treatment of cognitive programs as phenomenological theories.

3.2 Acquisition of Cognitive Programs

Key in a cognitive system's working is the availability of relevant knowledge, or cognitive programs. Even though the user could contribute to this knowledge by directly instructing the system, we envision that the main mechanism through which cognitive programs would be acquired will be offline or batch learning.

The most promising source of training material for learning commonsense knowledge is currently natural language text, both because of the existence of parsing and processing tools that are more advanced than those that exist for other media (e.g., images), but also because of the high prevalence of textual corpora. The World Wide Web has, typically, played the role of such a textual corpus for machine learning work seeking to extract facts (see, e.g., [41]). When seeking to extract, instead, knowledge appropriate for reasoning, an additional consideration comes into play: knowledge encoded in text from the World Wide Web is biased and incomplete in several ways with respect to our commonsense real-world knowledge, and would be more aptly called *websense* [36]. We posit, however, that certain deficiencies that a cognitive system could have by employing websense would be overcome through the user's feedback and supervision.

Acquisition of knowledge could proceed in several ways. For one, the cognitive system may memorize fragments of text that describe exemplars of certain concepts or scenarios (e.g., a typical restaurant scenario). In a somewhat more structured form, the cognitive system may compute and store statistics about word co-occurrences, e.g., in the form of n-grams, or in the form of frequencies of words appearing in a piece of text conditioned on certain other words also appearing. This last form of statistical information can be interpreted as a weighted association rule, with the weight indicating the "strength" or "probability" of the association holding. In an even more structured form, statistics as above can be stored not on words, but on relations extracted by parsing the text.

Beyond statistical information, one can attempt to learn reasoning rules over words or relations, using typical machine learning techniques. Some such techniques represent learned rules in a form understandable by humans (e.g., DNF formulas). Recent work has shown, in fact, that one can learn not only deductive rules, but also abductive ones, which provide possible explanations given a certain input to be explained [18]. Learning *causal* rules can also proceed naturally by treating consecutive sentences in a textual corpus as the before and after states needed for causal learnability [35]. Treating fragments of texts as partial observations of some underlying, even if unknown, truth or reality can be shown to *guarantee* [34] that rules learned in this manner will draw inferences that are not explicitly stated in, but follow from, a given piece of text. This task, known as *textual entailment* [8], contributes to one of the necessary processes (namely, the drawing of relevant inferences) for constructing a comprehension model.

The amount of knowledge that can be extracted from text is massive, and measures need to be taken to account for this. Section 2.1 has already pointed out that the user guides, explicitly or implicitly, the cognitive system on what concepts the system needs to focus on, and in turn these concepts determine what training material the system will seek for learning knowledge. Even with such guidance, the system may need to refrain from learning knowledge in the most specific form possible, since that would commit the knowledge to a very rigid representation that could not be used later in the context of different queries. Instead, the system should probably choose to retain the learned knowledge in a general representation, some examples of which we have discussed above.

This type of batch and query-independent learning could operate continuously, with the learned knowledge guiding its further development by identifying those concepts for which more training is needed. This process ensures, then, the gradual improvement of a system's cognitive programs, and hence their performance. When a query is posed, the process of knowledge compilation may invoke a further (online) form of learning, treating the offline-learned general knowledge as training data. This query-driven learning is much more focused (and could, in fact, be done implicitly [17]), and should, therefore, be sufficiently efficient to be carried out in real time between the user posing a query and receiving a response. The results of this online learning may be stored, and be reused for future query answering. Supervision by the user may provide additional training material for online learning, which would produce, therefore, user-specific knowledge.

In all cases, learning should proceed in a manner that anticipates reasoning. Valiant's *Probably Approximately Correct* (PAC) semantics for learning and reasoning [47, 48] points to how one could establish formal guarantees on the quality of learned cognitive programs and the comprehension models and inferences they produce. Recent work has proposed PAC semantics for two situations that are of particular interest to cognitive systems: when reasoning involves the chaining of multiple pieces of knowledge [37]; and, when a user's interaction with a cognitive system is personalized by learning to predict the user's intentions [38, 40].

4 Major Challenges for Cognitive Programming

Developing cognitive systems through the cognitive programming paradigm poses major technical challenges. We group and summarize below certain such challenges that would need to be overcome to make progress in this direction.

User-Machine Interaction. Cognitive systems need to interact with human users in a natural way through some fragment of natural language. Hence, the natural language processing capabilities of the supporting modules of cognitive programming are important. In particular, central questions include:

– How do we structure and restrict the complexity of natural language for the user-interface fragment of natural language, without, on the one hand, losing the expressiveness required by the applications, and while keeping, on the other hand, a form of natural communication with human users?
– How can we use existing natural language processing (NLP) systems for the syntactic and grammatical analysis of the user input to ascertain the concepts involved and to extract the narrative information? The use of better NLP tools should help us develop incrementally improved cognitive systems.
– How does the user become aware of the language and knowledge capabilities of the underlying cognitive programming framework? How can we develop useful schemes of dialogues between the user and cognitive systems for user feedback and for natural forms of supervision of the system by the user?

Reasoning with Common Sense. The basic form of argumentative cognitive reasoning and comprehension depends critically on many factors, when this is to be scaled up to be applied in many (if not all the) domains of discourse of common sense. The major questions that need concrete technical answers are:

– Does commonsense knowledge have a generic and task-independent vocabulary and form? What is an appropriate such form and how is this adapted (in real time, through knowledge compilation) into a useful task-specific form? In particular, how do we address the need for *syntactic plasticity* of commonsense knowledge, so that it can be adapted in a manner syntactically compatible with the vocabulary that the current input narrative is using?
– How are relevant parts of commonsense knowledge identified efficiently and reliably given an input narrative? In particular, how do we address the need for *conceptual plasticity* of commonsense knowledge, so that the concepts referred to in the input narrative are matched to concepts in the knowledge base? Is a meta-level form of "context indexing" of the knowledge needed?
– How do we integrate effectively the "pure reasoning" with the process of comprehension, while being guided by the central principle of coherence?

Acquiring Common Sense. Given that we have an appropriate representation for commonsense knowledge, we are then faced with the challenge of how to automatically learn and populate a commonsense library. Questions include:

– Is an offline or batch learning process for commonsense knowledge acquisition the only form of learning required, or do we also need a form of online learning at the time of query processing and knowledge compilation?
– How do we distinguish learned user-specific knowledge from learned generic commonsense knowledge given that the user supervises both processes, and how could learned knowledge be *reused* across users and cognitive systems?
– What are the main technical problems of "mining" commonsense association rules from the World Wide Web? What NLP techniques, search and download tools, storage and indexing schemes would be required? How do we overcome the possibly biased and incomplete nature of learned knowledge?
– How do we learn the annotations and priority tags of commonsense association rules? Can this process be automated, or is it ultimately user-specific?

To address many of these challenges, further *empirical study* with the help of Cognitive Psychology will be needed to help reveal possible answers and guide the development of the computational framework. The availability of a computational framework would then facilitate the experimental examination of the computational viability and effectiveness of various guidelines in improving the cognitive programming framework and the programming experience of the users. In particular, the central and major issues of knowledge plasticity and knowledge compilation are amenable to empirical psychological investigation.

In general, the development of cognitive programming needs to be informed and guided by the psychological understanding at different levels of human cognitive processes. Understanding how the mind operates at some higher conceptual level when dealing with everyday cognitive tasks can help us in developing possible models of computation in cognitive programming. On the other hand, understanding how humans introspectively perceive or understand the operation of their cognitive processes can help us develop *human-compatible* models of computation: models of computation that humans can naturally relate to.

5 Concluding Remarks

Ideas and proposals related to one form or another of cognitive systems go back to the very beginning of the history of AI, and it would be an interesting topic in itself to explore the development and confluence of these ideas. Among work carried out in more recent years on cognitive computing and systems, Watson is, perhaps, closest to a complete system, and has attracted the most attention from the media. Unlike its emphasis towards "help[ing] human experts make better decisions by penetrating the complexity of Big Data",[6] our proposal focuses on assisting ordinary people by supporting their everyday decision making.

Although both Watson and our envisioned systems seek to solve a cognitive task, the difference in emphasis outlined above suggests that for the latter systems it is crucial that the problem-solving process itself be cognitive, inspired by human heuristics and transparent to the ordinary people's way of thinking. It could be argued that the label "cognitive" should be reserved for such types of systems, and not be conferred to every system that solves a cognitive task.

Adopting this more stringent view of cognitive systems points to a second — in addition to developing intelligent machines — end for building them. Through their operation, cognitive systems could be used to empirically validate or falsify the theoretical models they implement, supporting the scientific process of hypothesizing, predicting, and revising. This iterative process would allow AI to contribute to the refinement of psychological theories of human cognition.

Following a vision where humans and machines share a similar level of common sense, we have proposed cognitive programming as a means to build cognitive systems. Cognitive programming adopts the view of a machine as a personal assistant: a human asks for the completion of a task, perhaps without fully and unambiguously specifying what is needed, but relying on the assistant's experience, and, ultimately, common sense, to perform the task. Cognitive programming aims to bring the flexibility of traditional programming to the masses of existing technology users, enabling them to view their personal devices as novice assistants, amenable to training and personalization through natural interaction.

Our proposal offers a blueprint of what needs to be done and the challenges that one will have to face. We are optimistic that it can be realized to a large extent by building on existing techniques and knowhow from Artificial Intelligence, especially when one takes a pragmatic view by synthesizing the theory and methods of AI with empirical results and ideas from Cognitive Psychology.

Unsurprisingly, the representation and reasoning requirements for cognitive programming are reminiscent of those of production rules as one finds in Computational Cognitive Psychology (see, e.g., [2, 20, 52]). For cognitive programming,

[6] See, for instance, this website: http://www.research.ibm.com/cognitive-computing/

production rules need to include the element of causality in their representation, be enhanced with a declarative form of representing and handling conflicts, and use some notion of (relative) strength of knowledge — or, of the arguments built from the underlying commonsense knowledge — when drawing inferences.

Logic Programming, and recent developments from this [24], have moved in this direction of production or reactive systems with such enhancements, but remain largely bound to the strict formal logical semantics. Similarly, frameworks for autonomous agents, such as BDI agents [42] and robotic agent programming [7], which aim amongst other things to give cognitive abilities to agents, also rely on strict logical or operational semantics. These approaches serve, therefore, a different class of problems from those aimed to by cognitive systems based on commonsense knowledge, and for which the role of comprehension is important.

One may argue that progress on natural language understanding would suffice to realize our vision of cognitive programming. Despite the important role of such progress, a fully automated natural language system would seem to require a machine architecture similar to that of the human brain. Given the gap between the formal logic-driven machine architectures of today (with long, rigid, and error-intolerant chains of computation — a limitation already identified by von Neumann [51]), and the cognitive capabilities and constraints of the human mind, our proposal of cognitive programming hopes to provide the middle-ware needed today to move closer to the ideal of an automated natural language system.

Acknowledgements

We are grateful to our colleague Irene-Anna Diakidoy (Department of Psychology, University of Cyprus) for her collaboration in our joint work on story comprehension, and the invaluable help that she has given us in our effort to understand the computational implications of relevant work in Cognitive Psychology. We also wish to thank Hugo Mercier and Francesca Toni for many valuable discussions on the psychology of reasoning and the nature of cognitive systems.

References

1. J. E. Albrecht and E. J. O'Brien. Updating a Mental Model: Maintaining Both Local and Global Coherence. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19(5):1061–1070, 1993.
2. J. R. Anderson, B. E. John, M. A. Just, P. A. Carpenter, D. E. Kieras, and D. E. Meyer. Production System Models of Complex Cognition. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, pages 9–12, 1995.
3. P. Baroni, M. Caminada, and M. Giacomin. An Introduction to Argumentation Semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.
4. T. J. M. Bench-Capon and P. E. Dunne. Argumentation in Artificial Intelligence. *Artificial Intelligence*, 171(10–15):619–641, 2007.
5. F. J. Bex and B. Verheij. Story Schemes for Argumentation about the Facts of a Crime. In *Proceedings of the 2nd Workshop on Computational Models of Narrative*, 2010.

6. E. Cabrio and S. Villata. Natural Language Arguments: A Combined Approach. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 205–210, 2012.
7. K. L. Clark and P. J. Robinson. Robotic Agent Programming in TeleoR. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5040–5047, 2015.
8. I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2013.
9. I.-A. Diakidoy, A. Kakas, L. Michael, and R. Miller. Story Comprehension through Argumentation. In *Proceedings of the 5th International Conference on Computational Models of Argument*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 31–42, 2014.
10. I.-A. Diakidoy, A. Kakas, L. Michael, and R. Miller. STAR: A System of Argumentation for Story Comprehension and Beyond. In *Proceedings of the 12th International Symposium on Logical Formalizations of Commonsense Reasoning*, pages 64–70, 2015.
11. P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.
12. J. S. Evans. Logic and Human Reasoning: An Assessment of the Deduction Paradigm. *Psychological Bulletin*, 128(6):978–996, 2002.
13. S. L. Frank, M. Koppen, L. G. M. Noordman, and W. Vonk. Computational Models of Discourse Comprehension. *Discourse Processes*, 45(6):429–463, 2008.
14. M. Gelfond and V. Lifschitz. Representing Action and Change by Logic Programs. *Journal of Logic Programming*, 17(2/3–4):301–321, 1993.
15. R. J. Gerrig. The Scope of Memory-Based Processing. *Discourse Processes*, 39(2–3):225–242, 2005.
16. A. C. Graesser, K. K. Millis, and R. A. Zwaan. Discourse Comprehension. *Annual Review of Psychology*, 48:163–189, 1997.
17. B. Juba. Implicit Learning of Common Sense for Reasoning. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 939–946, 2013.
18. B. Juba. Learning Abductive Reasoning Using Random Examples. In *Proceedings of the 1st Workshop on Cognitive Knowledge Acquisition and Applications*, 2015.
19. A. C. Kakas, L. Michael, and R. Miller. Modular-$\mathcal{E}$ and the Role of Elaboration Tolerance in Solving the Qualification Problem. *Artificial Intelligence*, 175(1):49–78, 2011.
20. D. E. Kieras and D. E. Meyer. An Overview of the EPIC Architecture for Cognition and Performance with Application to Human-Computer Interaction. *Human-Computer Interaction*, 12(4):391–438, 1997.
21. W. Kintsch. The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, 95:163–182, 1988.
22. W. Kintsch. *Comprehension: A Paradigm of Cognition.* Cambridge University Press, 1998.
23. W. Kintsch and P. Mangalath. The Construction of Meaning. *Topics in Cognitive Science*, 3:346–370, 2011.
24. R. Kowalski and F. Sadri. Programming with Logic without Logic Programming. *manuscript*, 2015.
25. R. Kowalski and M. Sergot. A Logic Based Calculus of Events. *New Generation Computing*, 4(1):67–95, 1986.

26. P. Langley. The Cognitive Systems Paradigm. In *Proceedings of the 1st Annual Conference on Advances in Cognitive Systems*, pages 3–13, 2012.
27. P. Langley, J. E. Laird, and S. Rogers. Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research*, 10(2):141–160, 2009.
28. A. Lieto, A. Minieri, A. Piana, and D. P. Radicioni. A Knowledge-Based System for Prototypical Reasoning. *Connection Science*, 27(2):137–152, 2015.
29. N. McCain and H. Turner. Causal Theories of Action and Change. In *Proceedings of the 14th AAAI Conference on Artificial Intelligence*, pages 460–465, 1997.
30. J. McCarthy and P. J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence*, 4:463–502, 1969.
31. D. McDermott. A Critique of Pure Reason. *Computational Intelligence*, 3:151–160, 1987.
32. D. S. McNamara and J. Magliano. Toward a Comprehensive Model of Comprehension. *The Psychology of Learning and Motivation*, 51:297–384, 2009.
33. H. Mercier and D. Sperber. Why Do Humans Reason? Arguments for an Argumentative Theory. *Behavioral and Brain Sciences*, 34(2):57–74, 2011.
34. L. Michael. Reading Between the Lines. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1525–1530, 2009.
35. L. Michael. Causal Learnability. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1014–1020, 2011.
36. L. Michael. Machines with WebSense. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning*, 2013.
37. L. Michael. Simultaneous Learning and Prediction. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, pages 348–357, 2014.
38. L. Michael. Introspective Forecasting. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 3714–3720, 2015.
39. L. Michael. Jumping to Conclusions. In *Proceedings of the 2nd International Workshop on Defeasible and Ampliative Reasoning*, 2015.
40. L. Michael. The Disembodied Predictor Stance. *Pattern Recognition Letters*, 64:21–29, 2015. Special Issue on Philosophical Aspects of Pattern Recognition.
41. T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-Ending Learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2302–2310, 2015.
42. A. S. Rao and M. P. Georgeff. BDI Agents: From Theory to Practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems*, pages 312–319, 1995.
43. D. N. Rapp and P. Van den Broek. Dynamic Text Comprehension: An Integrative View of Reading. *Current Directions in Psychological Science*, 14:297–384, 2005.
44. A. J. Sanford and S. C. Garrod. The Role of Scenario Mapping in Text Comprehension. *Discourse Processes*, 26(2–3):159–190, 1998.
45. P. Thagard. *Coherence in Thought and Action*. MIT Press, 2002.
46. M. Thielscher. From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem. *Artificial Intelligence*, 111:277–299, 1999.
47. L. G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
48. L. G. Valiant. Robust Logics. *Artificial Intelligence*, 117(2):231–253, 2000.

49. P. Van den Broek. Comprehension and Memory of Narrative Texts: Inferences and Coherence. In M. A. Gernsbacher, editor, *Handbook of Psycholinguistics*, pages 539–588. Academic Press, 1994.
50. D. Vernon, G. Metta, and G. Sandini. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *Transactions on Evolutionary Computation*, 11(2):151–180, 2007.
51. J. von Neumann. The General and Logical Theory of Automata. In A. H. Taub, editor, *John von Neumann: Collected Works. Volume V: Design of Computers, Theory of Automata and Numerical Analysis*, chapter 9, pages 288–328. Pergamon Press, 1961. Delivered at: Hixon Symposium, September 1948.
52. R. M. Young. Production Systems in Cognitive Psychology. In N. J. Smelser and P. B. Baltes, editors, *International Encyclopedia of the Social & Behavioral Sciences*. Elsevier, 2001.
53. R. A. Zwaan and G. A. Radvansky. Situation Models in Language Comprehension and Memory. *Psychological Bulletin*, 123:162–185, 1998.

Towards a Visual Remote Associates Test and its Computational Solver

Ana-Maria Olteţeanu*, Bibek Gautam**, and Zoe Falomir*

*Cognitive Systems, Universität Bremen
Bremen, Germany
**Department of Computer Science,
Southeastern Louisiana University
{amoodu,zfalomir}@informatik.uni-bremen.de

Abstract. The Remote Associates Test (RAT) is a test used for measuring creativity as relying on the power of making associations, and it normally takes a linguistic form (i.e., given three words, a fourth word associated with all three is asked for). The aim of this paper is towards generalizing this test to other domains, checking for its possible application in the visual domain (i.e., given three images, an object associated to them is asked for). A pilot visual version of the Remote Associates Test (RAT-V) was created and given to human participants. A previous solver of the compound linguistic Remote Associates Test (comRAT-C) was adapted to become a prototype which can solve the visual Remote Associates Test (comRAT-V).

Keywords: Remote Associates Test, Human Creativity, Visual Associates, Computational Creativity, Cognitive Systems

1 Introduction

Humans are capable of creativity across a wide variety of tasks and domains, including the linguistic (e.g. riddles, novels), visual (e.g. visual arts, design), auditory (e.g. musical), tactile (e.g. fashion and fabrics, texture), gustatory and olfactive (e.g. culinary creativity, perfumery), etc. Creativity in many domains runs across various sensory or linguistic modalities (e.g. literature, scientific discovery, innovation).

Complex creativity tasks, like the solving of insight problems, might ellicit both linguistic and visual creativity. Creativity tests which include both visual and linguistic elements do exist - like the Torrance Tests of Creative Thinking (TTCT), which contains both verbal and figural tests [6]. However, no such tests exist which can be given separately in both linguistic and visual forms, thus affording cross-domain comparison of a particular set of creative processes. The usefulness of such a test would be to: (i) check whether the same creative processes act across domains; (ii) compare performance results in various domains; and (iii) posit domain-relevant differences.

Aiming to fill this gap, this paper takes a well established creativity test, the Remote Associates Test [7] – for which a previous computational linguistic solver was implemented (comRAT-C [10]) under a theoretical creative problem-solving framework (CreaCogs [9, 8]) – and describes our approach towards developing a visual derivate of this test.

The rest of the paper is organized as follows. The Remote Associates Test and the construction of its visual counterpart (vRAT) are discussed in Section 2. A study with human participants who were given vRAT queries is described in Section 3. A short description of the linguistic comRAT-C together with its current prototype adaptation to solve visual queries is discussed in Section 4. Results on the experimentation carried out with human participants are provided in Section 5, while results of the computational comRAT-V prototype are described in Section 6. A discussion of this pilot test and prototype system are provided in Section 7 and further work is proposed.

2 Outlining the Remote Associates Test (RAT) and its Visual Counterpart

Imagine you are given three words - like CREAM, SKATE and WATER - and asked which is a fourth element common to all of them. This describes the Remote Associates Test originally devised by Mednick and Mednick [7]. The answer to this particular query is ICE.

The Remote Associates Test has been used in the literature [1, 5], and adapted to various languages [2, 4]. To check whether this creativity test could be adapted to more than linguistic examples, the authors decided to work towards a visual version of the RAT.

Different versions of the RAT [7] exist, after some researchers have argued that the items in the test were not all equal. Worthen and Clark [11] argued that some of these items are functional, and others structural. Functional items are those between which a non-language relationship is present (e.g. items like "bird" and "egg"), while structural items have previously been associated within a syntactic structure (e.g. items like "black" and "magic"). Compound remote associates correspond to structural associates in Worthen and Clark's categorization.

Normative data from compound remote associates [3] has been used before by the authors to evaluate a computational solver of the RAT [10] implemented using language data. In this paper, the authors use their understanding of this task to build a visual Remote Associates Test. In a previous formalization [10], the Remote Associates Test was described as follows: 3 words are given, w_a, w_b, w_c, and a word needs to be found, w_x, which relates to all three initial words. In the compound RAT case, terms $(w_a, w_x), (w_b, w_x)$ and (w_b, w_x) or their reverse, $(w_x, w_a), (w_x, w_b), (w_x, w_c)$ have to be successive or composed terms in the language in which the RAT is given in. In the case of composed terms, w_z might be another word composed of one of the initial terms and the solution term, like $(w_x w_a)$ or $(w_a w_x)$. For example, for the query AID, RUBBER and WAGON, the answer term BAND constructs composed terms with some of the query terms (BAND-AID, BANDWAGON), but not with others (RUBBER BAND). Note that the answer term is also not in the same position in the three linguistic structures.

In order to devise a visual RAT, the same mechanism was applied, with entities w_a, w_b, w_c and w_x being visual representations of objects and scenes. Thus, given entities w_a, w_b, w_c, there exists an entity w_x, which generally co-occurs visually with the other shown entities w_a, w_b and w_c.

For example, Fig. 1 provides the following entities: HANDLE, GLOVE and PEN. HAND is an appropriate answer to this query, being a visual entity which co-occurs with each of the given three. The visual entity HAND can be considered a visual associate of each of the initial objects HANDLE, GLOVE and PEN.

Fig. 1. Example of a visual RAT question. This is the first training query, showing the participants the following visual entities: HANDLE, GLOVE and PEN.

Each initial object is considered to have a variety of other visual associates. Therefore, this work assumes that visual associates are terms which play the role that word terms play in the language-based RAT. Visual associates which co-occur together, in a previously encountered visual scene or experience, play the same role as composed words or linguistic structures in which w_a and w_x co-occur. Thus, visual experiences containing the visual entities (HANDLE, HAND), (HAND, GLOVE) and (PEN, HAND) are required to solve the visual query shown in Fig. 1.

Next section explains the visual RAT test carried out by human participants.

3 Study with Human Participants: Answering the Visual RAT and Providing Visual Associates

The study carried out on human participants contained two parts. Participants where asked: (1) to solve some visual RAT queries and (2) to provide visual associates to some concepts not included in the previous queries. Participants were split in 4 groups, each group being given part of the RAT queries to solve, and the objects in the other queries to provide visual associates for.

Part 1

20 visual RAT queries plus 2 initial examples were set-up for initial experimen-

tation with human participants. Each of the queries showed 3 visual stimuli: objects (5 in training, 54 in test) or scenes[1] (1 in training, 6 in test).

The given training examples are showed in Fig. 1 and Fig. 2. The answer item is not contained in either of the initial images. The expected process is that participants could ellicit their visual memory about such co-occurrences of visual associates. Note that, Fig. 2 avoids presenting the sea while presenting the image of a beach, as the expected answer to this visual RAT query is WATER.

Fig. 2. The second training vRAT query showed the items above to the participants: BATHTUB, GLASS and BEACH.

Participants were instructed that:

- they would be presented with three objects or scenes, and asked to find a fourth element that is related to each of them;
- they could then choose between various ways in which they first perceived the answer when they arrived at it: (i) Visual imagery (they imagined the answer), (ii) Word (they thought of the answer verbally) and (iii) Other (in this case, they were asked to specify);
- they should provide a difficulty rating for each test item on a Likert scale, with a range from 1 (Very Easy) to 7 (Very Hard).

Afterwards, the test with the visual RAT queries followed.

Part 2

Participants were asked to contribute visual associates to a set of objects, which were query items for queries they have not received, as explained before. This task was explained as follows:

> *Visual associates are things you see when you imagine a particular object. These might be other objects, which are situated next to the object that you are imagining in some circumstance, or specific parts of the object you are imagining.*

[1] A *scene* is considered a visual display in which multiple objects might be considered salient. Parts of other objects may also be present when showing an *object* entity, but these parts were clearly not salient stimuli.

For example, visual associates for "glove" might be: hand, thorns, snow, scalpel, hot pan, bike, dirt. Visual associates for "pen" might be: paper, notebook, letter, test, form, cheque, desk, ink, drawing, writing, pen holder, ear, pen case, pencil, etc.
Imagine each item, and then write the visual associates that come to mind.

Grouping Procedure
The test was administered to four groups, via four different surveys developed using Google forms. The participants were asked to select their group themselves using a randomizer[2] which presented two Euro coins, on head or tails position. Depending on the coins arrangement provided by the randomizer, participants proceeded to one of the four groups tasks. All groups were shown the same initial two training examples. The 20 questions were split in four 5-question groups. Each of the four groups was asked to solve 3 sets of questions (thus 15 vRAT queries), and asked to offer visual associates for the objects in the fourth group of queries (thus 15 objects). The types of tasks (questions + visual associates) given to each group are specified in Table 1.

Table 1. The four groups in the study and their assigned tasks. Note that "Q" denotes a question, and n the number of participants in each group.

Study items	**Group 1** $n = 8$	**Group 2** $n = 15$	**Group 3** $n = 8$	**Group 4** $n = 12$	**Answers per item**
vRAT Training Examples	Yes	Yes	Yes	Yes	Shown to all
vRAT Q1-5	Yes	Yes	Yes	**No**	Gr. 1, 2, 3 ($n = 31$)
vRAT Q6-10	Yes	Yes	**No**	Yes	Gr. 1, 2, 4 ($n = 35$)
vRAT Q11-15	Yes	**No**	Yes	Yes	Gr. 1, 3, 4 ($n = 28$)
vRAT Q16-20	**No**	Yes	Yes	Yes	Gr. 2, 3, 4 ($n = 35$)
Visual associates for objects in questions	Q16-20	Q11-15	Q6-10	Q1-5	all objects across groups

Note that participants did not provide visual associates to a vRAT test item that they have previously answered, in order to avoid bias towards mentioning associations which were already made salient by the test items. The design we used in this study allowed for visual associations to be given to all objects across participants.

4 A Visual Computational Solver (comRAT-V)

This section describes how the computational visual RAT problem-solver works by describing its knowledge base content (Section 4.1) and its query solving process (Section 4.2).

[2] `https://www.random.org/coins/?num=2&cur=60-eur.germany-1euro`

4.1 Knowledge Base in comRAT-V

A previous system, comRAT-C, solved the compound RAT using language data [10]. Specifically, the most frequently occurring words appearing together as a tuple (2-grams or bigrams) were obtained from a genre-balanced Corpus of Contemporary American English (COCA)[3].

As the authors could not find in the literature any visual linked and annotated database which included the concepts used in the 20 queries included in the human test, the strategy followed was to ask the participants in the study for visual associates, as the previous section explains. Therefore, visual associates were obtained for all objects appearing in the 20 vRAT queries, that is, participants provided visual associates for a total of 60 objects. The objects were presented in such a way that a common associate will not be salient. These visual associates where used for the Knowledge Base (KB) of comRAT-V in the same way in which 2-gram relations were used by comRAT-C.[4]

Data thus obtained was cognitively valid data of visual associates obtained via introspection. This data was given to comRAT-V, which used it to construct its (visual) Concepts and Links knowledge base. The queries to be shown to humans were then given to comRAT-V. For each query, the three Concepts or Objects given in the query were ellicited from the KB, then Links were used to yield their visual associates. comRAT-V then offered the item(s) it converged upon as a possible answer.

A faster automatic way of extracting object associates from visual scenes data can be envisaged (see Section 7). However, the current prototype served our purpose to check whether comRAT will work with visual domain queries, and what was its performance.

4.2 Query Solving Process

The comRAT-C organized the data in its KB in Expressions, Concepts and Links between co-occurring Concepts. The comRAT-C solved RAT queries by activating the Concepts involved in each query in its KB, using the Links to navigate to syntactical items which those Concepts co-occurred with, and offering as a possible answer those items upon which this search and activation process converged, as shown in Fig.3.

The comRAT algorithm has been generalized to solve the linguistic and the visual RAT, which are equivalent in the nature of the processes they ellicit, although the type of data they input is different. Thus the likelihood of finding an answer based on frequency of the known items is computed in comRAT-V as in comRAT-C [10] when the system needs to choose one of multiple possible answers. When no 3-item convergence is made, comRAT-V checks for 2-item

[3] Corpus of Contemporary American English (COCA): `http://corpus.byu.edu/coca/`

[4] Thus an object and its visual (and implicitly spatial) associate is considered to be similar to a language term and its syntactic neighbour.

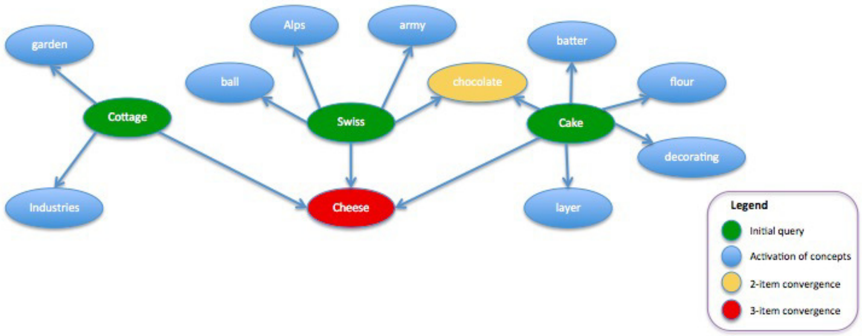


Fig. 3. Visual depiction of the search and convergence process in comRAT-C.

convergences. If multiple such terms are found, comRAT-V proceeds to compute the most likely of the terms and offers it as an answer.

5 Results from the Visual RAT Test with Human Participants

This section describes the participants to the visual RAT and vRAT results.

Participants
43 participants completed the study, 30 male and 13 female. The ages of the participants ranged between (btw.) 20 and 60 years old (y.o.), as follows: 6 btw. 20-30 y.o, 19 btw. 30-40 y.o., 14 btw. 40-50 y.o., 4 btw. 50-60 y.o. The self-assessed English level of the participants ranged between Intermediate and Native, as follows: 9 Intermediate, 21 Advanced, 10 Proficient, 3 Native.

Results
As shown in Figure 4, the percentage of participants solving the set of queries varied, between 6.45% (Q5) and 97.1% (Q20), with an average query solving percentage of 63%. Based on this, some queries may be classiffied as the three most difficult (Q5, Q13, Q16) and others as the three easiest (Q8, Q18, Q20).

As shown in Figure 5, participants declared they first perceived the answer mostly visually (56.6%) or as a word (38.9%). Some participants also declared that they did not know (3.26%) or that they perceived the answer via another sense, like feeling the heat when the answer was fire (0.16 %).

6 Results of the Computational Visual RAT (comRAT-V)

Visual associates provided by the participants to our study were added to comRAT-V's knowledge base. With this data, and no use of query frequency comRAT-V was already solving 14 of the 22 query items (63.64%). Then comRAT-V calculated the frequency of occurence of the visual associates, in order to apply

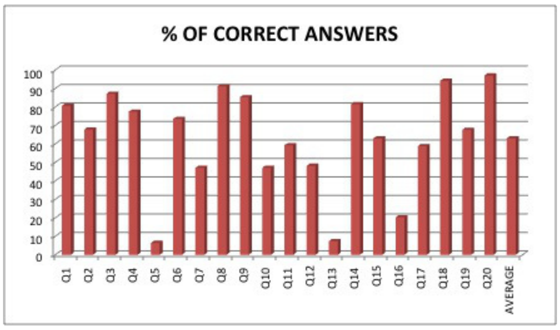


Fig. 4. Percentage of correct answers per query, as solved by the human participants.

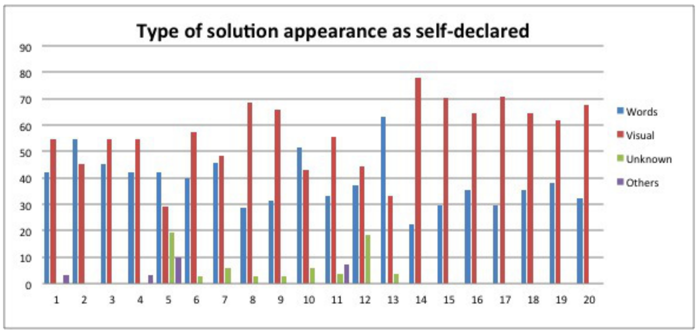


Fig. 5. Type of solution appearance as self-declared by participants.

the same frequency-based likelihood algorithm as comRAT-C [10] when selecting the answers. Given this data, comRAT-V managed to answer correctly 16 out of the 22 items, as shown in Table 2. Out of these, 13 correct answers came from 3 known items convergences, with 3 answers coming from 2 items known convergences.

Table 2. Analysis of the accuracy of responses provided by the system.

	1 item known	2 items known	3 items known	Total
Correct	0	3	13	16
Plausible	0	3	0	3
Not solved	2	1	0	3
Total	2	7	13	22
Accuracy	-	42.86% (85.71%)	100%	72.73%

Some queries encountered two or more possible answers. For Q8, two answers are possible from a 3-known items convergence - MEAT and CHEESE. However, the correct answer, MEAT, is chosen due to the frequency based likelihood. Similarly, Q21, in which a COMB, RAZOR and SHAMPOO are presented, encounters a larger set of possible correct answers. Amongst the possible three-item convergence answers (e.g.WATER, BATHROOM, MIRROR, etc.), the correct answer HAIR was chosen by comRAT-V. Queries can be answered correctly based on a two item convergence - for example Q14 was answered in this way, as only two of the visual associates linked the query items to the answer.

7 Discussion and Further work

Our current visual RAT prototype showed promise, as human participants were able to solve it (63%), a variety of difficulties were present in the different queries and 56.6% of participants said they arrived at the answer through visual imagery. Moreover, various participants declared that they enjoyed the vRAT test.

Whether queries were or were not solved through a visual imagery process is yet to be proven, as subjective reports are not reliable in this case. A fMRI-based experiment showing different language-based compound queries and vRAT queries might be able to show whether this is indeed the case. Humans might still translate visual stimuli in language stimuli, especially as the answer was asked for in language, and semantic relations are hard to avoid altogether.

However, as comRAT-V performed well based on visual associates provided by the human participants, we can assume that the queries can be solved using visual associations by humans as well. More visual affordance data is required to strengthen the current results, as these are based on visual associates and frequency of visual associates provided by the participants. As further work, the authors will focus on gathering more data for the comRAT-V knowledge base. Two ways to gather such data are envisioned:

- Get more human participants to provide visual affordances to all the objects used in the vRAT test, without giving them the test and/or
- Find a way to extract such visual associates automatically from images depicting indoor and outdoor scenes.

The authors plan to analyze whether there is a relationship between the results obtained with comRAT-V and human results in the vRAT. The authors also plan to increase the number of queries for the vRAT, since a larger set of queries might provide more insight and stronger results. A future focus will also be to investigate the different classes of difficulty in such queries, the preferred answers in multiple queries and the relation between fluency in providing visual associates by human participants and ability to solve the vRAT.

Acknowledgments

Dr. Ana-Maria Olteţeanu and Bibek Gautam are grateful for DAAD's support through its RISE project. Dr.-Ing. Zoe Falomir acknowledges funding by the project COGNITIVE-AMI (GA 328763) by the European Commission through FP7 Marie Curie IEF actions. All three authors acknowledge the support offered by the Universität Bremen and the Spatial Cognition Centre.

References

1. Ansburg, P.I.: Individual differences in problem solving via insight. Current Psychology 19(2), 143–146 (2000)
2. Baba, Y.: An analysis of creativity by means of the Remote Associates Test for Adult Revised in Japanese (JARAT FORM A). Japanese Journal of Psychology (1982)
3. Bowden, E.M., Jung-Beeman, M.: Normative data for 144 compound remote associate problems. Behavior Research Methods, Instruments, & Computers 35(4), 634–639 (2003)
4. Chermahini, S.A., Hickendorff, M., Hommel, B.: Development and validity of a Dutch version of the Remote Associates Task: An item-response theory approach. Thinking Skills and Creativity 7(3), 177–186 (2012)
5. Dorfman, J., Shames, V.A., Kihlstrom, J.F.: Intuition, incubation, and insight: Implicit cognition in problem solving. Implicit cognition pp. 257–296 (1996)
6. Kim, K.H.: Can we trust creativity tests? A review of the Torrance Tests of Creative Thinking (TTCT). Creativity research journal 18(1), 3–14 (2006)
7. Mednick, S.A., Mednick, M.: Remote associates test: Examiner's manual. Houghton Mifflin (1971)
8. Olteţeanu, A.M.: Two general classes in creative problem-solving? An account based on the cognitive processes involved in the problem structure - representation structure relationship. In: Besold, T., Kühnberger, K.U., Schorlemmer, M., Smaill, A. (eds.) Proceedings of the International Conference on Computational Creativity. Publications of the Institute of Cognitive Science, vol. 01-2014. Osnabrück (2014)
9. Olteţeanu, A.M.: From simple machines to Eureka in four not-so-easy steps. Towards creative visuospatial intelligence. In: Müller, V. (ed.) Philosophy and Theory of Artificial Intelligence. Synthese Library, Berlin:Springer (to appear)
10. Olteţeanu, A.M., Falomir, Z.: comRAT-C - A computational compound Remote Associates Test solver based on language data and its comparison to human performance. Pattern Recognition Letters (2015), http://dx.doi.org/10.1016/j.patrec.2015.05.015
11. Worthen, B.R., Clark, P.M.: Toward an improved measure of remote associational ability. Journal of Educational Measurement 8(2), 113–123 (1971)

Modeling the Creation and Development of Cause-Effect Pairs for Explanation Generation in a Cognitive Architecture

John Licato[1], Nick Marton[2,3], Boning Dong[2,3], Ron Sun[3], and Selmer Bringsjord[2,3]

Analogical Constructivism and Reasoning Lab (ACoRL)[1]
Indiana University/Purdue University - Fort Wayne
Fort Wayne, IN, USA

Rensselaer AI and Reasoning (RAIR) Lab[2]
Rensselaer Polytechnic Institute (RPI)[3]
Troy NY USA

Abstract. The ability to generate explanations of perceived events and of one's own actions is of central importance to how we make sense of the world. When modeling explanation generation, one common tactic used by cognitive systems is to construct a linkage of previously created cause-effect pairs. But where do such cause-effect pairs come from in the first place, and how can they be created automatically by cognitive systems? In this paper, we discuss the development of causal representations in children, by analyzing the literature surrounding a Piagetian experiment, and show how the conditions making cause-effect pair creation possible can start to be modeled using a combination of feature-extraction techniques and the structured knowledge representation in the hybrid cognitive architecture CLARION. We create a task in PAGI World for learning causality, and make this task available for download.

Keywords: Explanation, Cognitive Architecture, CLARION, Analogy, Causality

1 Introduction

Faced with some unfamiliar event, an agent[1] will attempt to make sense of it by constructing an explanation, even if the explanation that ultimately gets accepted is not entirely coherent. Generating explanations is also important to artificial cognitive systems, particularly those that need to communicate with other humans, for example, to present rationales for its own actions.

[1] In this paper, 'agent' will refer to any actor (artificial or natural) capable of cognitive thought, 'cognitive system' will refer to any system that attempts to model cognitive phenomena, and 'cognitive architecture' will refer to full cognitive systems (such as CLARION) satisfying the definition of cognitive systems in [20].

Previous work (e.g., [6, 9, 14]) modeled the generation of explanations by using structured representations of cause-effect pairs. In an extremely simple case, explaining some explanandum e involves finding a cause-effect pair (c, e), where c is either believed to be true by the reasoner or plausible to the reasoner in some sense. More complicated explanations can be generated by collecting a sequence of cause-effect pairs and lining them up to produce a causal chain [14], by drawing from multiple source analogs simultaneously [8, 9], or a number of other possible approaches. But these approaches all seem to presuppose the existence of cause-effect pairs, and little is done in the way of actually modeling how the initial cause-effect pairings are initially created.

In this paper, we attempt to understand how the sort of cause-effect pairs that are used in explanation generation can be created in an autonomous agent, in a psychologically plausible way. Section 2 reviews some literature on the emergence of causality in children, focusing on a classical Piagetian experiment we will call the *floating task*. We then describe a task, implemented in the simulation environment PAGI World, for testing abilities that underly the autonomous creation of cause-effect pairs, along with an algorithm to perform this task, implemented in the cognitive architecture CLARION (Section 3). Section 4 discusses future work and concludes.

2 The Development of Causality

If we are to understand how cause-effect pairs can be created automatically by a cognitive system, it would be very helpful to understand how the ability to reason causally develops in humans. We will start with a particularly relevant Piagetian experiment.

2.1 The Piagetian Floating Task

In one of Jean Piaget's early works, *The Child's Conception of Physical Causality*, Piaget introduced a task to elicit clues from children as to how they generate explanations. In what we will refer to here as the *floating task*, Piaget presents a series of objects to a child (e.g., a wooden boat, a pin, a pebble, and so on) and asks the child to predict whether or not the object (the candidate floating object) would float. The child makes his prediction, explaining his or her reasoning when possible, and then the object is placed in the water. The child watches whether or not his prediction was correct, and then is asked to explain why the object did or did not float.

Piaget found that the responses given by children seemed to be roughly categorizable into four stages. These stages are to be seen as continuously changing behavioral phenomena, meant to describe general trends noticed in subjects' explanations. In the first stage, explanations are characterized by "animistic and moral reasons," e.g. a boat will float "because they must always lie on the water," or a piece of glass will sink "because it's not allowed to put glass on the water" [17, p.136]. Piaget described stage-1 explanations as moral because they seemed to him to imply a sense of social obligation on the part of the inanimate objects, as opposed to adherence to some natural law.

In the second stage, starting at about 5 years of age, we see the appearance of dynamism, or the invocation of an abstract force in explanations. Children explain that boats float because they're heavy, big, or because the "water is strong." However, they apply their explanations in inconsistent or contradictory ways. Compare this to the third stage (starting at about 5 or 6 years), where children instead tend to use the explanation that boats are *light*, rather than heavy. The difference here, according to Piaget, is subtle but important: floating is no longer explained by an appeal to a simple property of the candidate floating object. Rather, the lake "produces an upward-flowing current which sustains the lighter [floating] body." In other words, floating is understood to be a property that emerges out of an interaction necessitated by both properties of the lake and properties of the candidate floating object.

Finally, in the fourth stage (starting at about age 9, but parts of which are seen as early as ages 6–8), we start to see reasoning taking into account multiple properties of an object simultaneously. By referring to the hollow-ness of the boat, for example, children relate the boat's volume to its weight. Furthermore, whereas in stage 3 properties of the candidate floating object like light-ness or heavy-ness are no longer regarded by the child to be absolute, internal properties. Instead, they are seen as properties that only hold relative to something else (in this case a corresponding volume of water).

2.2 Why Piaget?

Piaget's work is extremely voluminous, spanning almost 60 years, and careful scholars have noted evolutions in Piaget's thought that at times puts the younger Piaget against the older [3]. In part because Piaget's writings are so spread out over so many books, many of his concepts, which he refined in his later years, are subject to misinterpretations of the highest order. For some corrections of misunderstandings of Piagetian concepts, see [4, 15, 12].

For example, the description of stages that we reiterated in Section 2.1 is exemplary of the type of stage-based development that critics are quick to claim is virtually useless, since the scientific consensus is that "cognitive changes are gradual and cumulative" [1]. Contrary to such claims, however, Piaget was very aware of the limitations of using stages in describing children's behavior:

> "[S]tages must of course be taken only for what they are worth. It is convenient for the purposes of exposition to divide the children up in age-classes or stages, but the facts present themselves as a continuum which cannot be cut up into sections. This continuum, moreover, is not linear in character, and its general direction can only be observed by schematizing the material and ignoring the minor oscillations which render it infinitely complicated in detail" [18, p.17].

That being said, it is not the goal of this paper to mount a full-scale defense of the Piagetian body of literature. Although it cannot be denied that some of

Piaget's theories are incompatible with, and need to be refined by, more recent work in developmental psychology, let it suffice to point out that the critics of Piaget are overzealous in indiscriminately discarding the entirety of his work, especially the almost 60 years of qualitative observations of children's behavior. Even if one were to ignore all of Piaget's proposed explanations for developmental mechanisms, his observations remain a fertile ground for cognitive modelers, as they provide at the very least a set of expected behaviors of children of different ages when faced with very specific tasks. We described some of these behaviors in Section 2.1, and the current paper intends to model them.

3 Modeling the Development of Cause-Effect Representation in CLARION

The CLARION cognitive architecture [19] is divided into four subsystems: the action-centered, non-action-centered, meta-cognitive, and motivational subystems. Each of these is split into explicit and implicit components, thus enabling the deliberative processes associated with localist representations to work in parallel with the automatic processes associated with distributed representations. This dual-process approach to modeling cognition has been shown to be capable of modeling a variety of behavioral phenomena in psychologically plausible ways. For example, [22] implemented similarity-based and rule-based reasoning in the non-action-centered subsystem (NACS for short). Building on these processes, [13] showed that structured knowledge, and thus primitive deductive and analogical reasoning, can also be modeled in the NACS. And building on the structures of [13], the authors demonstrated a high-level approach to generating explanations of varying quality in [14].

The present paper can be considered another in that sequence. As mentioned earlier, the previous model of explanation [14] used cause-effect pairs, implemented as *template structures* (a particular type of organization of localist chunks in the explicit level of the NACS). But where do these cause-effect pairs come from? One way, suggested by the performance of the younger children in Piaget's floating task, is known as *feature selection.* Given a set of features of the object under consideration, the child will somehow select some subset of these features (in the case of stage-1 children, a subset consisting of a single feature) and hypothesize that the presence of this particular feature is the cause of the phenomena under observation (floating or sinking). In CLARION, feature selection comes naturally out of the operations of a backpropagation network built into CLARION [21].

In CLARION's NACS, localist chunks corresponding to outputs can be placed on the top level, and microfeatures corresponding to inputs and hidden nodes can be placed on the bottom level. In Section 3.2, we set up the NACS in this way, and apply a feature selection algorithm to the floating task. First, we turn to a description of our computational simulation of the floating task.

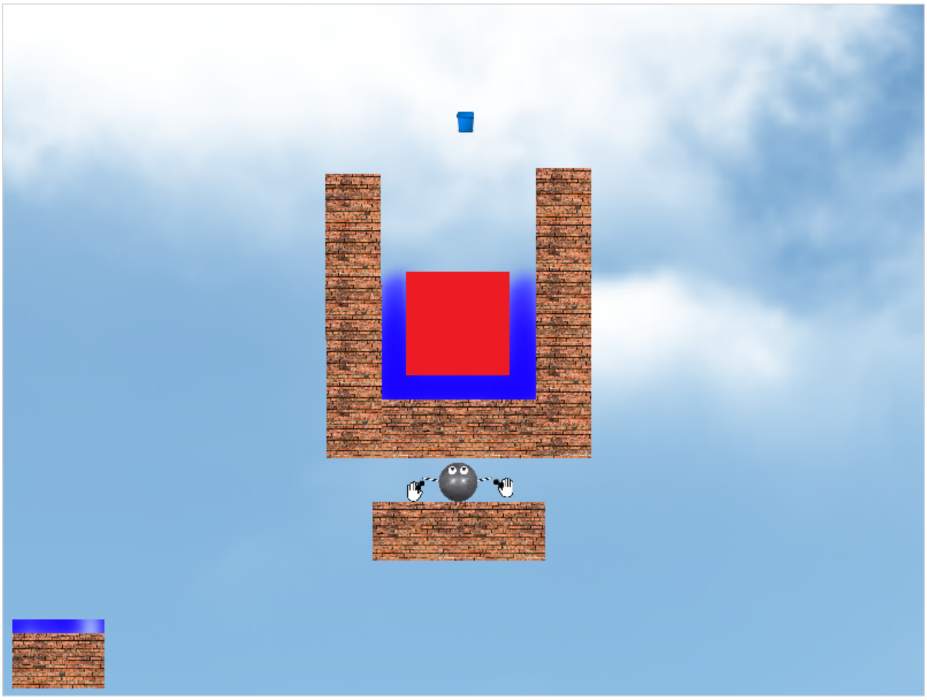


Fig. 1. The Floating Task

3.1 A Task in PAGI World

PAGI World [2, 16, 12] is a simulation environment for the evaluation and development of AGI and cognitive systems. PAGI World is built in Unity, allowing for execution on all major operating systems. It is built on Unity's 2D physics engine, so that mass, volume, velocity, texture, temperature, etc., can be experienced by the AI actor in a realistic way. The AI actor (a ball-shaped creature with two hands, who we sometimes refer to as 'PAGI guy') is controlled by a script that can be written by researchers in any programming language that supports TCP/IP. The information sent between the controller script and PAGI World is mostly low-level: PAGI World sends information from its visual, tactile, and other sensors (including some medium-level data such as object names), while the controller script can send commands to apply a force vector to PAGI guy's body and hands to control it.

PAGI World is easy to learn and use, thanks to design choices that we hope will encourage researchers to make use of PAGI World. Because it can be run on almost any operating system and controlled using almost any programming language, PAGI World provides a platform for cognitive architectures of all types (particularly those which claim to be general-purpose) to compare their performance on the exact same tasks.

Piagetian experiments are somewhat difficult to model computationally for two important reasons: First, they often rely on objects that need to move in a physically realistic way, and it is nontrivial for researchers to program sufficiently realistic simulations for every model they create; second, assessing agents in Piagetian experiments makes heavy use of explanatory dialogue, that is, the experimenter must be able to ask questions about the task and the subject must be able to answer them. Although this second difficulty is one that is still beyond the reach of AI researchers, the first difficulty is handled quite nicely by PAGI World, since PAGI World has the ability to simulate water and create objects that float or do not float in it.

Thus, for all of the reasons discussed above, PAGI World is an ideal choice for hosting the Piagetian floating task. In our implementation, PAGI guy is positioned below a tank of water. An object with a randomly generated volume and weight is created, and appears in the middle of the tank, where it then either floats to the top, sinks to the bottom, or stays relatively motionless (Figure 1). After a few seconds, this object disappears and the process repeats. This allows PAGI guy to collect data about what it observes, so that we can later ask questions.

Algorithm 1 The Feature Selection Algorithm Used in Each Experiment of Section 3.2

```
Require: Set of features F = {f_i}
Require: Set of training examples EX = {ex_1, ..., ex_n}
Require: Number of epochs e
Require: Number of iterations it
  for i iterations do
      for all f_i ∈ F do
          Build neural network n_i, using only f_i as sole input
          for  doe epochs
              for  doex_i ∈ EX
                  Execute network n_i with ex_i
                  Update weights of n_i w/backpropagation
              end for
          end for
          for  doex_i ∈ EX
              Execute n_i with ex_i; compare prediction to actual result
          end for
          Determine final error rate by averaging over all ex_i ∈ EX
      end for
      Select the feature that had the highest accuracy rate
  end for
  Return the number of times each feature was selected as having the highest accuracy
  rate
```

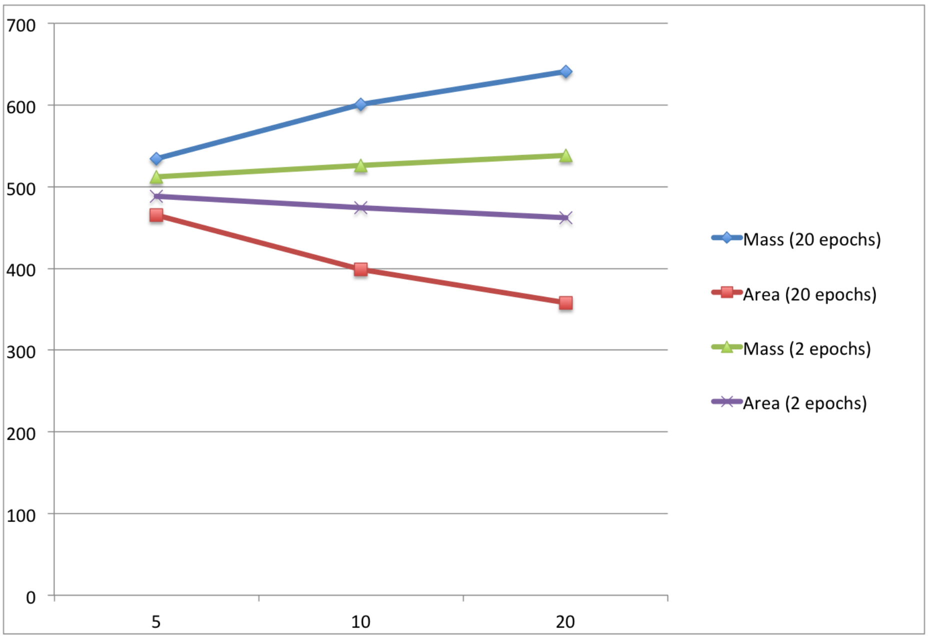


Fig. 2. A plot of the number of training examples n (x-axis) vs. the amount of times each feature was chosen for having the lowest error rate (y-axis). Note that only mass and area are shown here, since the other color-related features showed less than once per thousand iterations.

3.2 Bottom-Up Feature Selection in CLARION

In this section we demonstrate that a simple feature selection algorithm can be implemented in CLARION, by using a network that takes in low-level micro-features and outputs a prediction as to whether an object will float, sink, or remain stationary. Feature selection is an inherent property of backpropagation, in the sense that as backpropagation updates weights, certain nodes (which can correspond to features) will have higher weights connected to them than others.

CLARION is designed to work with low-level distributed networks that can be trained with backpropagation. We started by creating a network consisting of five inputs, all microfeatures in the bottom level of CLARION's NACS: mass, volume, and three microfeatures for color (red, green, blue). Each input can be activated by a value between 1 and 255. Three outputs are created, each of them implemented as a chunk in the top level of the NACS: float, sink, and stationary. We also create five additional microfeatures $h_1, ..., h_5$, to serve as the hidden layer of the network.

Feature selection proceeds as follows. We collected sensory data from instances of the floating task in PAGI World, where an object of randomized color, mass, and weight appears in the middle of the tank and floats, sinks, or remains stationary. Each instance of an object appearing in the floating task is recorded and called an *example*. The input features are then individually isolated; that is, we only activate one feature at a time, allow the activation to propagate up to the hidden microfeatures $(h_1, ..., h_5)$, and further up to the output chunks, and the output chunk with the highest activation is taken to be the 'prediction' of this particular instance. We repeat this for n examples; weights are updated using backpropagation after every example. One successful run-through of all n examples is called an *epoch*. We then execute another epoch, runing through the same n examples again.

After e epochs, we evaluate the average error on the same n examples that the network was trained on. Note that this differs greatly from standard machine-learning practice: generally a test data set is used that is non-overlapping with the training data set. However, we are not necessarily interested in getting the correct prediction; we are interested in modeling the reasoning of the child in a way that is psychologically plausible. It is psychologically plausible that a child would use a limited set of examples from his memory to validate hypotheses or features, and it is less plausible that a child would run through a set of thousands of training examples first.

In any case, the evaluation of error on the n examples gives us an error rate for the feature that was isolated. We can then repeat this entire process with the other features, obtaining an error rate for each feature. The feature that had the lowest error rate is taken to be the winner of this iteration. (Originally, we also recorded the feature that had the second-lowest error rate, but because the results were so overwhelmingly in favor of mass and volume (a color-related feature was selected less than once per 1000 iterations), we only present the data here for the lowest error rate.) The feature-selection algorithm is laid out in a more convenient form in Algorithm 1.

The iterations were repeated 1000 times per experiment. We carried out this experiment six times, for three different values of n ($n \in \{5, 10, 20\}$) and two different values of e ($e \in \{2, 20\}$). Figure 2 shows the value of n on the x-axis, and the number of times (out of 1000 iterations) some particular feature was chosen as having the lowest error rate on the y-axis.

The values of n we chose for each experiment were intentionally very small. It seems implausible that children carrying out the floating experiment would actually be trained using hundreds of instances before they output their predictions. Therefore, we kept n very low in order to see what results emerged. As it turns out, the results match our intuitions: using our feature-selection algorithm settles extremely quickly on either the mass or volume features, and the only growth we see as n and e are increased is a slowly growing gap between the amount of times mass is chosen and the amount of times volume is chosen (a gap which was larger for 20 epochs than it was for 2 epochs).

The fact that even tiny values of n and e identify mass and volume as the most relevant features is consistent with the idea that, in line with Piaget's suspicions, the growth allowing the more complex explanations of stage-2 and later reasoning is a growth in the complexity of the representations themselves—that is, new nodes (corresponding to new concepts) might be created to represent abstract ideas such as density, water-current, and higher-level features constructed out of the lower-level ones used in our experiments.

4 Future Work and Conclusion

This paper presents a task designed to closely model the Piagetian floating task, and then shows how the behaviors of stage-1 children can be explained as feature selection over simple representations in the CLARION cognitive architecture. Future work will attempt to explain the sequence of behaviors shown by Piaget in the floating task. For example, the ability to consider multiple properties at once (which appears in stage-3 children) may be explained using a template structure designed to group properties together. Likewise, the shift from single-place predicates to relations seen in stage 4 might be explained by a stabilization of the property groupings and the emergence of two-place predicates (a similar strategy is used in the DORA model [5]).

Another series of tasks, highly relevant to the study of the development of causality in children, may be interesting to examine using the model developed in this paper. These are the series of "collision" tasks [10, 11], in which infants can identify when some basic notions of physical causality are violated. When shown two objects that are about to collide, but one of them unexpectedly changes direction or stops before the collision is supposed to have taken place, infants will stare at the anomalously behaving object longer than they would at objects colliding normally. We have already started creating this task in PAGI World and hope to show that the present model can match the performance of human children closely.

The backpropagation used in this paper for feature selection is one of many ways CLARION can select features. In the future, as we tackle more complex tasks, we can make use of, e.g., principal component analysis (PCA) or sparse autoencoders [7].

Causality, of course, is an immensely complex and well-studied topic, and early steps such as those taken in this paper can only hope to scratch the surface. Future work will expand the philosophical, psychological, and historical perspectives on the notion of causality and how it relates to explanation generation.[2]

5 Acknowledgements

This work was funded by Office of Naval Research (ONR) grant number N000141310342.

[2] The floating task presented in this paper is available for download, along with PAGI World, at the website:

`http://rair.cogsci.rpi.edu/projects/pagi-world/pagi-world-tasks/`

We encourage researchers to test their particular cognitive architectures or systems on this and other tasks, and report their results.

References

1. Anderson, J.R., Simon, H.A., Reder, L.M.: Radical Constructivism and Cognitive Psychology. In: Ravitch, D. (ed.) Brookings Papers on Education Policy. Brookings Institute Press, Washington, DC (1998)
2. Atkin, K., Licato, J., Bringsjord, S.: Modeling Interoperability Between a Reflex and Reasoning System in a Physical Simulation Environment. In: Proceedings of the 2015 Spring Simulation Multi-Conference (2015)
3. Beilin, H.: Piaget's Enduring Contribution to Developmental Psychology. Developmental Psychology 28(2), 191–204 (1992)
4. Chapman, M.: Constructive Evolution: Origins and Development of Piaget's Thought. Cambridge Univ Press (1988)
5. Doumas, L.A., Hummel, J.E., Sandhofer, C.: A Theory of the Discovery and Predication of Relational Concepts. Psychological Review 115(1), 1–43 (2008)
6. Friedman, S.E., Forbus, K.: An Integrated Systems Approach to Explanation-Based Conceptual Change. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. Atlanta, GA (2010)
7. Gregor, K., LeCun, Y.: Learning Fast Approximations of Sparse Coding. In: Proceedings of the 27th International Conference on Machine Learning. pp. 399–406 (2010)
8. Hummel, J.E., Landy, D.H.: From Analogy to Explanation: Relaxing the 1:1 Mapping Constraint...Very Carefully. In: Kokinov, B., Holyoak, K.J., Gentner, D. (eds.) New Frontiers in Analogy Research: Proceedings of the Second International Conference on Analogy. Sofia, Bulgaria (2009)
9. Hummel, J.E., Licato, J., Bringsjord, S.: Analogy, Explanation, and Proof. Frontiers in Human Neuroscience 8(867) (2014)
10. Leslie, A.M.: Spatiotemporal Continuity and the Perception of Causality in Infants. Perception 13, 287–305 (1984)
11. Leslie, A.M., Keeble, S.: Do Six-Month-Old Infants Perceive Causality? Cognition 25, 265–288 (1987)
12. Licato, J.: Analogical Constructivism: The Emergence of Reasoning Through Analogy and Action Schemas. Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY (May 2015)
13. Licato, J., Sun, R., Bringsjord, S.: Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In: Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN) (2014)
14. Licato, J., Sun, R., Bringsjord, S.: Using Meta-Cognition for Regulating Explanatory Quality Through a Cognitive Architecture. In: Proceedings of the 2nd International Workshop on Artificial Intelligence and Cognition. Turin, Italy (2014)
15. Lourenço, O., Machado, A.: In Defense of Piaget's Theory: A Reply to 10 Common Criticisms. Psychological Review 103(1), 143–164 (1996)
16. Marton, N., Licato, J., Bringsjord, S.: Creating and Reasoning Over Scene Descriptions in a Physically Realistic Simulation. In: Proceedings of the 2015 Spring Simulation Multi-Conference (2015)
17. Piaget, J.: The Child's Conception of Physical Causality. Routledge (1930/1999)
18. Piaget, J.: The Moral Judgment of the Child (1960)
19. Sun, R.: Duality of the Mind: A Bottom Up Approach Toward Cognition. Lawrence Erlbaum Associates, Mahwah, NJ (2002)
20. Sun, R.: Desiderata for Cognitive Architectures. Philosophical Psychology 17(3), 341–373 (Sep 2004), `http://www.informaworld.com/openurl?`

genre=article\&doi=10.1080/0951508042000286721\&magic=crossref|
|D404A21C5BB053405B1A640AFFD44AE3
21. Sun, R., Peterson, T.: Autonomous Learning of Sequential Tasks: Experiments and Analyses. IEEE Transactions on Neural Networks 9(6), 1217–1234 (November 1998)
22. Sun, R., Zhang, X.: Accounting for Similarity-Based Reasoning within a Cognitive Architecture. In: Proceedings of the 26th Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates (2004)

A Cognitive View of Relevant Implication

Claudio Masolo and Daniele Porello

Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy

Abstract. Relevant logics provide an alternative to classical implication that is capable of accounting for the relationship between the antecedent and the consequence of a valid implication. Relevant implication is usually explained in terms of information required to assess a proposition. By doing so, relevant implication introduces a number of cognitively relevant aspects in the definition of logical operators. In this paper, we aim to take a closer look at the cognitive feature of relevant implication. For this purpose, we develop a cognitively-oriented interpretation of the semantics of relevant logics. In particular, we provide an interpretation of Routley-Meyer semantics in terms of conceptual spaces and we show that it meets the constraints of the algebraic semantics of relevant logic.

1 Introduction

Paradoxes of classical material implication often show a mismatch between our intuitions concerning valid patterns of reasoning and the formalization of implication provided by classical logic. Debates on the nature of implication can be traced back to the very origin of modern logic, involving for instance Brentano, Husserl, and Frege. Turning to contemporary developments of mathematical logic, the problem of the logical properties of implication has been approached by providing systems that aims to mend classical logic from inference patterns that are not motivated on the basis of a specific view of reasoning.

Since in any logical system, the implication has the important role of encoding the properties of logical inference, by rejecting the properties of classical implication, one is often lead to rejecting classical logic. For instance, *intuitionistic logic* criticizes the non-constructive nature of classical implication. For that reason, intuitionists designed an alternative logic that rejects inference by contradiction and the law of the excluded middle. Moreover, *relevant logic* criticizes the lack of connection between the premises and the conclusion of a logical inference made explicit by some valid formula of classical logic, e.g., $A \to (B \to A)$—once A holds, one can infer that any B entails A—or $(A \to B) \vee (B \to A)$—every pair of propositions can be connected by means of an implication. By keeping track of the antecedent-consequent connection, relevant logic prevents these paradoxes.

Furthermore, classical implication does not model any sort of relationship between the knowing subject and the matter of the proposition. The truth-conditional definition of the classical implication $A \to B$ is given in terms of those states of affairs such that either the state of affairs corresponding to A does not hold or the state of affairs corresponding to B holds. Prosaically, $A \to B$ is true whenever A is false or B is true. The relationship between the antecedent and the consequent of a classical implication can be understood only in terms of mere co-occurrence between the states of affairs of the corresponding propositions. The knowing subject is construed as a spectator of an independent reality that displays itself. A number of approaches to non-classical logics can be categorized as proposals to make logical implication sensitive to *cognitively relevant* aspects. For instance, intuitionistic logic models the abstract concept of a knowing subject and intuitionistic semantics is better understood in terms of *proof-conditions* instead of truth-conditions, where a proof is intended to model the activity of a knowing subject [20] with respect to propositions. A significant number of non-classical logics are motivated by the idea of taking into account the activity of the knowing subject, e.g., just to mention a few, *justification logics* [8], *proof-theoretical semantics* [21], and number of *relevant* and *substructural logics* [13, 5]. Each of this approaches stresses that the information required to asses the status of a proposition is an essential part of the meaning of the proposition.

We place our analysis within the tradition of relevant logics [13, 2], a family of logics that have been traditionally interpreted as logics of information [12, 1, 13]. In particular, the analysis of relevant implication aims to investigate the connection between the information contained in the antecedent and the information contained in the consequence. Although relevant logics are effective in preventing paradoxes of material implication, a drawback is that their algebraic semantics has been criticized on the ground that it lacks any strong intuitive motivation [5]. To cope with that, a number of approaches to relevant logics provided an intuitive reading of the semantics. From the point of view of cognition, the most interesting approach is due to Mares [13] who interprets deduction in relevant logics in terms of *situated inference*. Intuitively, a situation contains information that is relevant to make a proposition hold, thus situations are truth-makers of propositions. In this paper, we provide a version of the semantics of relevant logic based on a notion of situation defined in terms of the theory of *conceptual spaces* [9], a theory on how we conceptualize the reality and how we reason on this conceptualization. Our aim is to motivate the idea of situated inference provided by Mares by means of the rich theory of cognition formalized by means of conceptual spaces. The exhibition of a concrete instance of the semantics of relevant logics based on a well developed model of cognition has a double impact: (*i*) it provides a clean cognitive interpretation of relevant logics; and (*ii*) it shows that relevant logics capture cognitively important aspects of inferences.

The paper is organized as follows. Sections 2 and 3 introduce the background on relevant logic and conceptual spaces. Section 4 informally describes the interpretation of the semantics of relevant logic in terms of conceptual spaces, while Section 5 provides the formal construction. Section 6 concludes the paper.

2 Relevant logic

We introduce a minimal background on the relevant logic **R** [2, 13, 7]. We confine ourself to the implicative fragment of **R** that, by slightly abusing the notation,

1. $A \to A$
2. $(A \to B) \to ((B \to C) \to (A \to C))$
3. $A \to ((A \to B) \to B)$
4. $(A \to (A \to B)) \to (A \to B)$

Table 1. Axioms for **R**

we still label by **R**. Let *Atom* be a set of propositional atoms and $p \in Atom$, the language of **R** is inductively defined by:

$$L_{\mathbf{R}} := p \mid A \to A$$

The axioms for **R** are presented in Table 1 while its Hilbert system is introduced as usual through the notion of derivation $\vdash_{\mathbf{R}}$. The base case states that $\vdash_{\mathbf{R}} \phi$, where ϕ is an axiom in Table 1. The rule of *modus ponens* is then added: if $\vdash_{\mathbf{R}} A$, $\vdash_{\mathbf{R}} A \to B$, then $\vdash_{\mathbf{R}} B$. By reasoning in **R**, a number of paradoxes of classical implication are blocked. For instance, the monotonicity of the entailment $A \to (B \to A)$, which is an axiom in classical logic. Its meaning is: if A holds, then every B entails A, regardless the relevance of B for assessing A. Accordingly, in relevant logics that axiom is not valid. Moreover, in case we also assume a disjunction in our language, $(A \to B) \vee (B \to A)$ is not a theorem of **R**.

2.1 Routley-Meyer Semantics

We present the model of substructural logic in terms of ternary relations, that is due to Routley and Meyer [15, 18]. Ternary relations can be viewed as a generalization of (relational) Kripke semantics for intuitionistic and modal logics. Let S be a set of points and $R \subseteq S^3$. Moreover, let $1 \in S$ be a designated element. We define the following notations:

- $R^2(xy)zw$ iff there is an $u \in S$ such that $Rxyu$ and $Ruzw$;
- $x \leq y$ iff $R1xy$.

Definition 1 (Substructural frame)). *A substructural frame $\mathcal{S} = (S, 1, R)$ is a set S, with $1 \in S$, equipped with a ternary relation R such that:*

A1. $x \leq x$ $(R1xx)$
A2. $Rxxx$
A3. *if* $R^2(xy)zw$, *then* $R^2(xz)yw$
(*if there is u s.t. $Rxyu$ and $Ruzw$, then there is v s.t. $Rxzv$ and $Rvzw$*)
A4. *if $Rxyz$, then $Ryxz$*
A5. *if $Rxyz$ and $x \leq w$, then $Rwyz$*

A valuation in a substructural frame is defined by $v : Atom \to \mathcal{P}(S)$. The valuation is required to satisfy the following *heredity condition*: for every $p \in Atom$, if $x \in v(p)$ and $x \leq y$, then $y \in v(p)$. The valuation extends to any formula of R, by the semantics of implication:

– $s \models A \to B$ iff for all r,t such that $Rsrt$, if $r \models A$, then $t \models B$.

Heredity has to extend to complex formulas, and it is easy to check that it is the case. The concept of truth in a model is defined by evaluating propositions at the particular designed state 1.

Definition 2 (Substructural model). *A substructural model $(\mathcal{S}, v)$ is a substructural frame $\mathcal{S}$ equipped with a valuation v that satisfies heredity on atoms. A formula A is true in a substructural model $(\mathcal{S}, v)$ iff $1 \models A$. Moreover, A is valid iff it is true in every substructural model $(\mathcal{S}, v)$.*

This semantics is sufficient to show that the logic $\mathbf{R}$ is sound and complete with respect to substructural models. The motivation for introducing a ternary relation R is that it is needed for the semantics of implication: R relates the states that are making $A \to B$, A, and B hold. Although the semantics based on ternary relations has been criticized for its abstract nature, there is a number of possible intuitive reading of R, cf. [5]. One of the reading of R groups the first two components of the relation, $R[xy]z$, and can be read as "the combination of information in x and y is in z". This interpretation has been analyzed in more details by Mares [13] in terms of *situated inference*. In very abstract terms, the valuation associates situations to formulas and $s \models A$ holds whenever the information contained in situation s is relevant for A. The clause for implication states that $A \to B$ holds at s if the information contained in s combined with the information contained in r produces information t that is relevant for B. We shall focus on this reading in order to provide a concrete cognitively-oriented interpretation of ternary relations semantics.

3 Conceptual spaces

Gärdenfors [9] proposes a cognitive model of representations based on the notion of *conceptual space*. The theory of conceptual spaces is grounded on the notion of *similarity*: "[j]udgments of similarity (...) are central for a large number of cognitive processes (...) such judgments reveal the dimensions of our perceptions and their structures" ([9], p.5). *Quality dimensions*—e.g., temperature, weight, pitch, brightness—correspond to "the different ways stimuli are judged to be similar or different" ([9], p.6). They are modeled as (possibly discrete) sets of points that represent *exact* similarities between individuals. Those points represent the *qualities* of individuals: two individuals are located in the same point when they are (cognitively or empirically) indistinguishable with respect to the considered dimension, e.g., they have the same temperature, the same quality. Furthermore, dimensions have a *geometrical structure* that organizes their points according to the level of similarity between stimuli.

A set S of dimensions is *integral* if an individual located in one dimension is necessarily located also in all the other dimensions in S. For example, {*hue*, *brightness*} is integral because if an individual has a hue it necessarily has a brightness (and viceversa). A set of dimensions is *separable* if it is not integral, e.g., {*hue*, *pitch*}. In Gärdenfors's terminology, *domains* are maximal sets of integral dimensions. For example, the *hue*, *chromaticness*, and *brightness* dimensions that form the color domain {*hue*, *chromaticness*, *brightness*} are integral and separable from any other dimension. Domains are central in the work of Gärdenfors because, by means of the separability condition, they can be used to assign *properties* to individuals *independently* of other properties. For instance, in empirical terms, the weight and the color of an individual can be measured independently. The *classificatory* nature of the sensory systems is defended also by Matthen [14]. In these views, properties do not have a strong ontological connotation, they do not capture how the world is but how it appears to us through our sensory systems (or artificial sensors).[1] The properties and the conceptual spaces are understood *relativistically*: their structure depends on the underlying culture, on measurement methods and sensors (in *science*), or on interpretation of the behavior of subjects (in the case of *phenomenology*). However the determinate-determinable relation, see [19], makes sense also in this case. Fully determinate properties, i.e., maximally resolving properties according to the sensors one dispose of, are represented by points in the domain. Vice versa, determinable properties, properties that abstract from the resolution of the sensor, are represented by *regions*, i.e., sets of points in the domain. For instance, 'being scarlet' and 'being crimson' can be seen as points, while 'being red' as a region containing the previous two points. *Natural properties* are convex regions.

Conceptual spaces are defined as collections of one or more domains and *concepts* are represented as regions in conceptual spaces. They are *static* theoretical entities "in the sense that they only describe the *structure* of representations" ([9], p.31). *Natural concepts* are sets of regions in different domains "together with an assignment of salience weights to the domains and information about how the regions in different domains are correlated" ([9], p.105).

Finally, an *individual* is represented as a point in a conceptual space, a vector of coordinates in the dimensions of the space. The points of the space can then be seen as the representations of *possibilia*, the set of all the possible individuals.

4 From conceptual spaces to substructural models

Our goal is to provide a cognitive interpretation of the relevant logic **R**. More specifically, following the idea of Mares, we provide an interpretation of the substructural models of **R** (cf. Definition 2) in terms of the theory of conceptual spaces properly modified and simplified for our goal. In this section we informally present our idea while Section 5 contains the technical details.

We assume a finite and fixed number N of (disjoint) domains. The ith domain is noted $\mathbf{D}_i$. The dimensions of the domains are not relevant for our task, then, to simplify our framework, we do not consider them. Consequently, we lose the original distinction between qualities and properties and all our domains are assumed as separable from the others. In addition, (fully) determinates, originally represented by points of a domain, are here singletons. In this way, both

[1] Causation links between how the world is and how it appears to us can be considered.

the determinates and the determinables (the regions) are elements of a domain $\mathbf{D}_i$. This move simplifies the formalization and is consistent with a mereological view of domains where determinates correspond to atomic regions (see [6]). Furthermore, $\subseteq$ is the only relation between regions we consider, no topological or geometrical relations are introduced.[2] Finally, we represent the classification of objects[3] under the properties in the domains but not their *categorization* under the concepts. Actually, concepts are not needed for our goal. This may appear as an oversimplification of the original theory of conceptual spaces. However, note that (*i*) our notion of domain is perfectly aligned with the original that can be seen as a limit case of the one of concept, i.e., regions in the domains are simple concepts; (*ii*) links between domains useful to define natural concepts are modeled via *correlations* (see below); (*iii*) the basic framework introduced here can be easily modified to take into account dimensions while categorization is an extension that could underline a new kind of implication (in addition to the ones we discuss in Section 6) to be addressed in future work.

The original idea of representing individuals as vectors of points (singletons in our case) each one belonging to a different domain is too strong for our aims. This view assumes a complete knowledge about the individuals, while we are interested in the acquisition of knowledge, information, or data, about individuals. We then weaken the original theory by allowing two kinds of partial knowledge about individuals: (*i*) the exact location into a domain is not known, i.e., one can only assign a determinable property to the individual, e.g., one knows it is red, but not the exact shade of red; (*ii*) one does not have any information about a given property, one does not even know if an individual is located in a given domain, e.g., if it has a color or not. Firstly, note that in (*i*) one may consider the maximal region of a domain. That means, for instance, that one only knows that the individual is colored. Secondly, (*ii*) contemplates the case of individuals that lack some properties, i.e., individuals are not necessarily located in all the domains. For instance, abstract individuals are not in space, while holes do not weight. However, we do not represent the impossibility to be located in a domain[4] but only the lack of information (see below).

The assumption that the conditions of individuation of objects are purely conceptual has been criticized by Pylyshyn. In [16] he explores the idea that "[p]art of what it means to individuate something is to be able to keep track of its identity despite changes in its properties and location" ([16], p.33). The initial individuation and tracking of objects is not conceptual, i.e., it is not based on the classification under concepts, it is based on a lower level mechanism built into the visual system called FINST. We cannot enter here into the details of the approach. What is interesting for us is the link, provided by Pylyshyn, with the theory of *object files* [11]. One "can think of an object file as a way for informa-

[2] Consequently, the structural relations of spaces, e.g., distances or orders, are here only used to build the taxonomy of properties. As discussed in Section 6, this structural information could be also used to represent relations among objects.

[3] From here we use 'object' and 'individual' as synonymous.

[4] That could be useful for approaching the semantics of negation.

tion to be associated with objects that are selected and indexed by the FINST mechanism. When an object first appears in view (...) a file is established for that object. Each object file has a FINST reference to the particular individual to which the information refers." ([16], p.38) The file allows us to group and maintain all the informations associated to the same individual (maybe acquired or updated through time), in particular "the one-place predicates that pertain to that object" ([16], p.39). An object file may be seen as an updatable frame-based description of an individual.[5]

Following this idea, we assume a fixed set OB of objects that are described by objects files defined as tuples $\langle a, R_1, \ldots, R_n \rangle$ where $a \in OB$ and $R_i \subseteq \mathbf{D}_i$ is a set of regions of $\mathbf{D}_i$. Firstly, object files are contextual, they depend on the chosen sets of domains and objects. Secondly, they collect all the known properties of a given object, i.e., all their known locations inside the domains. Intuitively, an object file represents the whole information about an object one has at a given stage, i.e., in an ontological perspective, the collection of *states of affairs* [3] relative to the same object. Thirdly, the R_i are sets of regions rather than simply regions. This extension of the original notion of location into domains is required to represent the process of making the acquired knowledge about an object explicit. As an illustrative example, assume that the color domain contains three subregions such that: $\mathtt{scarlet} \subset \mathtt{red} \subset \mathtt{colored}$. In $f = \langle a, \{\mathtt{scarlet}\} \rangle$ the only explicit knowledge is the scarletness of a, whereas $f' = \langle a, \{\mathtt{scarlet}, \mathtt{red}\} \rangle$ adds the redness of a. By looking at the structure of the color domain, the knowledge in f' was already present in f, but only in an implicit form, i.e., f' is the result of an inference process, a cognitive abstraction activity. In mathematics, one can see this situation as the introduction of a new theorem. The theorem was implicit in the theory but, by making it explicit, we add, in some sense, information.[6] Fouthly, we need to guarantee that object files contain consistent information, e.g., it is possible to have $\langle a, \{\mathtt{scarlet}, \mathtt{red}\} \rangle$ but not $\langle a, \{\mathtt{red}, \mathtt{blue}\} \rangle$ (if 'being red' and 'being blue' are disjoint). Finally, $R_i = \emptyset$ represents the total lack of information, discussed above, about the ith domain. In particular, $f = \langle a, \emptyset, \ldots, \emptyset \rangle$ represents just the existence of $a \in OB$.

A situation can be seen as a set of object files for the objects OB with respect to the domains $\mathbf{D}_1, \ldots, \mathbf{D}_{\mathrm{N}}$, i.e., as a the collection of states of affairs relative to the objects OB expressible with the same set of properties. Because the R_i in the object files may be the empty set or may represent determinable properties, in general the situations capture partial information about the objects. In particular, the situation 1 is the situation where all the object files have the form $\langle a, \emptyset, \ldots, \emptyset \rangle$, i.e., the situation 1 represents only the *terminological* knowledge.

[5] Note that we do not consider time, updating must be intended in terms of knowledge or information acquisition steps.

[6] In an empirical scenario where one disposes of instruments with different resolutions, the previous situation could be seen as the acquisition of a new measure with a coarser resolution. We do not consider this interesting observational perspective where one could also acquire new measures with identical resolution, e.g., one would be able to distinguish $\langle a, \{\mathtt{scarlet}, \mathtt{scarlet}\} \rangle$ from $\langle a, \{\mathtt{scarlet}\} \rangle$.

Then, we model the reachability relation between situations in terms of updates of the information contained in a situation. Intuitively, given the situation s, t, and u, $Rstu$ holds when the object files in u can be obtained by means of the ones in s and t through two possible types of updating: *abstraction* and *correlation*. Abstraction generalizes conceptualization within the same domain (e.g. from scarlet to red), i.e., it relies on the $\subseteq$-structure of domains. Vice versa correlation individuates dependencies between distinct domains, for instance, it may relate colors and shapes. *Induction*, as understood by Gärdenfors, is an example of correlation: "[t]he essential role of induction is to establish *connections* among concepts or properties *from different domains*" ([9], p.211). More specifically, the "inductive process corresponds to determining *mappings* between the different domains of a space. Using such a mapping, one can then determine correlations between the regions of different domains. The correlation between two properties F and G, expressed on the symbolic level by a universal statement of the form "all Fs are Gs," would then just be a special case" ([9], p.228). We represent only the simple correlation between two properties by a pair of regions, the regions that represent these properties.

Finally, following the Routley-Meyer Semantics, the function of valuation v assigns to any atomic proposition a set of situations.

5 Conceptual spaces and relevant logic

We formally define the notions introduced in the previous section. A domain $\mathbf{D}$ is given by the set of all regions over a set of values $D=\{p_1,\dots,p_l\}$: $\mathbf{D}=\mathcal{P}^{\star}(D)=\mathcal{P}(D)\setminus\emptyset$, where we exclude $\emptyset$ to avoid counterintuitive "null properties". In what follows, we fix a set of N domains $\mathbf{D}_1,\dots,\mathbf{D}_{\mathrm{N}}$, denoted by $\bar{\mathbf{D}}$. We denote by r_i^1, ..., r_i^n the elements of a domain $\mathbf{D}_i$. Elements r_i^j are called *regions* of the domain. We sometimes use names for labeling regions. For instance, let $D=\{p_1,p_2,p_3\}$, then $\mathbf{D}$ has as elements regions such as $\{p_1\}$, $\{p_2\}$, and $\{p_1,p_2\}$. We may then label $\texttt{scarlet}=\{p_1\}$, $\texttt{crimson}=\{p_2\}$ and $\texttt{red}=\{p_1,p_2\}$.

Given a domain $\mathbf{D}_i$, we denote by $R_i\subseteq\mathbf{D}_i$ a set of regions in $\mathbf{D}_i$.

Definition 3 (Consistency). *We say that R_i is* consistent *iff if $R_i\neq\emptyset$, then $(\bigcap_{r\in R_i} r)\neq\emptyset$.*

Intuitively, as we will see, consistent sets of regions can be intended as non-exclusive properties that can in principle be ascribed to an object. In case the set of regions is empty, it represents the absence of information of type $\mathbf{D}_i$ concerning that object. For instance, $R_i=\{\texttt{scarlet}=\{p_1\},\texttt{red}=\{p_1,p_2\}\}$ is consistent, since the intersection of the regions in R_i is not empty, whereas $R_i'=\{\texttt{scarlet}=\{p_1\},\texttt{crimson}=\{p_2\}\}$ is not. That is, we can say that an object is both scarlet and red, as for instance $\texttt{scarlet}\subseteq\texttt{red}$, but we cannot say that it is both scarlet and crimson.

Moreover, we fix a set $OB=\{a_1,\dots,a_l\}$ of objects.

Definition 4 (Object files). *An object file f_a is a vector $\langle a,R_1,\dots,R_n\rangle$, where $a\in OB$, $R_i\subseteq\boldsymbol{D}_i$, such that each R_i is consistent.*

The set of all object files depends on the choice of $\bar{\mathbf{D}}$ and OB, so we denoted by $OBF^{OB}_{\bar{\mathbf{D}}}$.We can now introduce the definition of situation.

Definition 5 (Situation). *A situation s is a set of object files $s \subseteq OBF^{OB}_{\bar{\boldsymbol{D}}}$ such that, for every object $a \in OB$, there exist a unique object file $f_a \in s$.*

Then, we assume a number of *correlations* relating regions in different domains.

Definition 6 (Correlations). *A set of correlations COR is a set of pairs of regions (r^l_i, r^m_j), where $r^l_i \in \boldsymbol{D}_i$ and $r^m_j \in \boldsymbol{D}_j$, $i, j \in \{1, \ldots, \mathrm{N}\}$ and $i \neq j$. Moreover correlations satisfy the following conditions:*

Restricted transitivity *if $(r^l_i, r^m_j) \in COR$, $(r^m_j, r^n_h) \in COR$, and $h \neq j$, then $(r^l_i, r^n_h) \in COR$.*

Correlation composition *if $(r^l_i, r^m_j) \in COR$ and $r^h_i \subseteq r^l_i$, then $(r^h_i, r^m_j) \in COR$;*
if $(r^l_i, r^m_j) \in COR$ and $r^m_j \subseteq r^h_j$, then $(r^l_i, r^h_j) \in COR$.

Restricted transitivity states that if we can connect two regions in a number of steps, we can also connect them by composing the correlations in one single step. The condition $h \neq j$ in the restricted transitivity excludes that we end up relating regions of the same domain. For instance, it prevents passing from $(\texttt{scarlet}, \texttt{round})$ and $(\texttt{round}, \texttt{crimson})$ to $(\texttt{scarlet}, \texttt{crimson})$. The rules for correlation composition state that if we correlate a concept with another, the correlation applies also to the subconcept of the first one and to super-concept of the second one. For instance, if we say that red things are round, we also say that scarlet things are round. We do not put any further consistency constraint on correlations. The reason is that correlations are intended to represent factual, but not necessarily correct, mappings between concepts. For instance, we do not exclude from COR correlations that can end up in inconsistent outcomes, e.g. $(\texttt{round}, \texttt{scarlet})$ and $(\texttt{round}, \texttt{crimson})$. The point is that correlations express matters of fact, thus they are falsifiable and in principle revisable. By contrast, conceptual information is fixed and non-revisable.

We turn now to the interpretation of the ternary relation R in our setting. Intuitively, situations are related if they are reachable by means of an abstraction move or by means of a correlation link. Denote by f^s_a the (unique) object file f_a in situation s. Moreover, denote by $R^s_{a,i}$ the set of regions of $\mathbf{D}_i$ that in situation s are associated to object a. We are ready now to present our interpretation of the ternary relation in terms of reachability of situations.

Definition 7 (Reachability of situations). *Let u, t and s situations in $OBF^{OB}_{\bar{\boldsymbol{D}}}$. The situation u is reachable from t given s, i.e., $Rstu$, iff:*

R1. for all $a \in OB$, for all $R^u_{a,i}$ then $R^u_{a,i} \supseteq (R^s_{a,i} \cup R^t_{a,i})$;
i.e., all the data in s and t are imported in u;

R2. for all $a \in OB$, for all $r \in R^u_{a,i} \backslash (R^s_{a,i} \cup R^t_{a,i})$, r is obtained in one of the two following ways:

Abstraction *there exists $r' \in R^s_{a,i} \cup R^t_{a,i}$ such that $r' \subseteq r$;*

Correlation *there exists* $r' \in R^s_{a,j} \cup R^t_{a,h}$ *such that* $(r', r) \in COR$.

$Rstu$ imposes that the whole information in u is derived (by using conceptual knowledge or correlations) from the one in s and the one in t. R1 entails that the regions in s and t are preserved in u. R2 shows that all the new regions in u are derived from the ones in s and t by abstraction or by correlation. Note that, in principle, a situation could be updated through abstraction and correlation into something that is not a situation, i.e., into a set of inconsistent object files. For instance, suppose that $(\texttt{round}, \texttt{crimson}) \in COR$ and that `scarlet` and `crimson` are disjoint. Suppose $\bar{\mathbf{D}}$ contains just two domains, e.g. colors and shapes. Thus, a situation s that contains $\langle a, \{\texttt{scarlet}, \texttt{red}\}, \{\texttt{round}\}\rangle$ can be updated, by means of the correlation $(\texttt{round}, \texttt{crimson})$, to a set of object files that contains $\langle a, \{\texttt{scarlet}, \texttt{red}, \texttt{crimson}\}, \{\texttt{round}\}\rangle$, which violates consistency of the sets of regions that is required for object files. Since we are assuming that R is defined on situations, i.e. sets of object files with consistent R_i-sets, the case above is excluded. This point shows a significant difference between abstraction and correlation: abstraction guarantees consistency of the update, whereas correlation does not. This reflects the distinction between conceptual and factual knowledge. Once the conceptual relations are set and we have assumed that they are consistent, by abstraction we can only generalize on given data. By contrast, correlations introduce new data that may be inconsistent with previous ones.

We define the following relation of *consistency* between situations (Cst)

Definition 8 (Consistent situations Cst). *The two situations s and t are consistent, noted by Cst, iff:*

C1. for $i \in \{1, \ldots, \mathrm{N}\}$, $R^s_{a,i} \cup R^t_{a,i}$ *is consistent* (*cf. Definition 3*)

By means of Definition 7, we can infer that, if a situation u is reachable from t given s, then u is consistent both with s and with t and s is consistent with t.

Proposition 1. *If $Rstu$, then Csu, Ctu, and Cst.*

Proof. Assume $Rstu$, that entails by R1 that for every i and every object a, $R^s_{a,i} \cup R^t_{a,i} \subseteq R^u_{a,i}$. Thus, since $R^u_{a,i}$ is consistent by definition, then $R^s_{a,i} \cup R^t_{a,i}$ is consistent, so Cst. The other cases follows by noticing that $R^s_{a,i} \subseteq R^u_{a,i}$ and $R^t_{a,i} \subseteq R^u_{a,i}$. □

We conclude this paragraph by providing an interpretation of the element 1 of the substructural model. We define 1 as the situation in which we have no information about any object, i.e., $1 := \{\langle a, \emptyset, \ldots, \emptyset\rangle \mid a \in OB\}$. Every $\langle a, \emptyset, \ldots, \emptyset\rangle$ is an object file, that is, it satisfies consistency. Moreover 1 is a situation, since for every object, there exist a unique object file in 1.

5.1 Conceptual spaces as models of R

We can now show that our view of situations provides a model of relevant logic.

Definition 9 (Conceptual substructural model). *A conceptual substructural model is given by* $(\langle \mathcal{S}, COR, R, 1\rangle, v)$*, where* $\langle \mathcal{S}, COR, R, 1\rangle$ *is a conceptual substructural frame:* $\mathcal{S}$ *is a set of situations defined wrt. a domain* $\bar{\boldsymbol{D}}$ *and a set of objects* OB*,* COR *is a set of correlation between regions of* $\bar{\boldsymbol{D}}$*,* $R \subseteq \mathcal{S}^3$ *is a reachability relation, and* $1 := \{\langle a, \emptyset, \ldots, \emptyset\rangle \mid a \in OB\}$*. Moreover,* v *is a valuation that associates to atoms sets of situations, i.e.,* $v : Atom \to \mathcal{P}(\mathcal{S})$ *such that heredity holds.*

We only need to show that R and 1 satisfy the axioms of Definition 1.

Proposition 2. *The reachability of situations R satisfies axioms A1, A2, A3, A4, and A5 of Definition 1.*

Proof. We only show the details of the representative cases.
A1: $R1ss$. R1 trivially holds. R2 holds because $R^s_{a,i} \setminus (R^1_{a,i} \cup R^s_{a,i}) = \emptyset$.
A2: If $Rstu$, then $Rtsu$. It is sufficient to notice that the definition of R is symmetric wrt. $R^s_{a,i}$ and $R^t_{a,i}$.
A3: If $R^2(st)uw$, then $R^2(su)tw$. We need to show that if there exists an x such that $Rstx$ and $Rxuw$, then there exists a y such that $Rsuy$ and $Rytw$. Assume that there exists an x such that $Rstx$ and $Rxuw$.
We show that there is a y such that $Rsuy$ and $Rytw$. We set for every a and i, $R^y_{a,i} = R^s_{a,i} \cup R^u_{a,i}$.
Firstly, we show that $Rsuy$. We have that $R^s_{a,i} \cup R^u_{a,i} \subseteq R^y_{a,i} = R^s_{a,i} \cup R^u_{a,i}$, thus R1 is fine. Since there is no other regions in $R^y_{a,i}$, we can conclude that R2 is also satisfied. Hence, $Rsuy$.
Then, we have to show $Rytw$. By assumption, $R^x_{a,i} \cup R^u_{a,i} \subseteq R^w_{a,i}$, thus we can deduce $R^y_{a,i} \cup R^t_{a,i} \subseteq R^w_{a,i}$. So R1 is satisfied.
Suppose now that there is an $r \in R^w_{a,i} \setminus R^y_{a,i} \cup R^t_{a,i}$, that is $r \in R^w_{a,i} \setminus R^s_{a,i} \cup R^u_{a.i} \cup R^t_{a,i}$. Since by assumption $Rxuw$, every region r in w is obtained by abstraction or correlation from regions in x or u. If r is obtained by abstraction or correlation from a region in $R^u_{a,i}$, then we are done, since $R^u_{a,i} \subseteq R^y_{a,i}$. If r is obtained from regions in x, then, by assumption $Rstx$, so r is obtained from regions that are either in s or t. We approach the following cases:
(i) r is obtained by correlation from an $r' \in R^x_{a,i}$ and r' is obtained from correlation from an $r'' \in R^s_{a,i}$. This means that $(r', r), (r'', r') \in COR$, thus, by restricted transitivity, $(r'', r) \in COR$. Therefore, if $r \in R^w_{a,i} \setminus R^s_{a,i} \cup R^u_{a.i} \cup R^t_{a,i}$, then there is an $r'' \in R^s_{a,i}$ such that $(r'', r) \in COR$, thus we conclude;
(ii) r is obtained by correlation from an $r' \in R^x_{a,i}$ and r' is obtained by abstraction from an $r'' \in R^s_{a,i}$. This means that $r'' \subseteq r'$ and $(r', r) \in COR$, thus, by the first rule of correlation compoistion, $(r'', r) \in COR$. Therefore, for $r \in R^w_{a,i} \setminus R^s_{a,i} \cup R^u_{a.i} \cup R^t_{a,i}$, there is an $r'' \in R^s_{a,i}$ such that $(r'', r) \in COR$ and we conclude again;
(iii) r is obtained by abstraction from r' in x and r' is obtained by abstraction from r'' in s. Then r can be obtained by abstraction from r'' and we are done;
(iv) r is obtained by abstraction from r' in x and r' is obtained by correlation $(r'', r') \in COR$ from r'' in s. In this case, $r' \subset r$ and $(r'', r') \in COR$, thus by the second rule of correlation composition we infer $(r'', r) \in COR$ and we conclude.

Therefore, also R2 is satisfied, therefore $Rytw$.
A4: $Rsss$ holds, since R1 trivially holds and $R^s_{a,i} \setminus (R^s_{a,i} \cup R^s_{a,i}) = \emptyset$.
A5: If $Rstu$ and $w \leq s$, then $Rwtu$. Recall that $w \leq s$ is defined by $R1ws$. We show only the following case. Suppose $r \in R^u_{a,i} \setminus R^w_{a,i} \cup R^t_{a,i}$ and that r is obtained by means of correlation from r'. In case $r' \in R^w_{a,i} \cup R^t_{a,i}$, we are done. Otherwise, by assumption $r' \in R^s_{a,i}$ and $(r, r') \in COR$. Since $R1ws$, there are two cases. Firstly, there exists r'' such that $(r'', r') \in COR$. By restricted transitivity, we conclude that for $r \in R^u_{a,i} \setminus R^w_{a,i} \cup R^t_{a,i}$, there exists an r'' in $R^w_{a,i}$ such that $(r'', r) \in COR$. Secondly, r' is obtained by abstraction from r'' in $R^w_{a,i}$, in this case by correlation composition, we conclude. $\square$

It is important to notice that the provided interpretation in terms of situations does not trivialize the substructural model, namely R does not provide a model of intuitionistic or classical implication. To see that, we show that monotonicity does not hold in conceptual substructural models. In axiomatic terms, monotonicity corresponds to the validity of $A \rightarrow (B \rightarrow A)$. In semantic terms, it corresponds to the following constraint on the ternary relation [7]:

f1 $\quad Rstu \Rightarrow R1su$

Consider a simple example where $s = \{\langle a, \emptyset\rangle, \langle b, \{\texttt{scarlet}\}\rangle\}$, $t = \{\langle a, \{\texttt{scarlet}\}\rangle, \langle b, \emptyset\rangle\}$, and $u = \{\langle a, \{\texttt{scarlet}, \texttt{red}\}\rangle, \langle b, \{\texttt{scarlet}\}\rangle\}$. In this case, although $Rstu$, neither $R1su$ nor $R1tu$ hold, i.e., both the information in s and t is needed for u. Therefore, (f1) does not hold in every conceptual substrctural model, thus $A \rightarrow (B \rightarrow A)$ is not valid.

6 Conclusions and future work

We presented a concrete instantiation of the ternary relation model of the relevant logic **R** that is grounded on the framework of conceptual spaces.Our instantiation of the Routley-Meyer semantics provides a number of reasons to interpret relevant implication in terms of cognitively aware updates of knowledge. Besides the logical contribution, we believe that both the notion of situation and the one of reachability between situations provide a useful framework for separating conceptual and factual knowledge and for modeling knowledge acquisition. However, the cognitive plausibility of the interpretation of the inferential mechanism we proposed still lacks an empirical assessment.

Future work concerns two directions. Firstly, notice that the proposed framework provides an interpretation only to atomic propositions that reduce to the assignment of a (unary) property to an object, it does not consider relations among objects. The extension to relations definable in terms of relations among intrinsic properties of the relata is quite trivial.[7] Using conceptual spaces, this kind of relations can be represented by means of *higher level properties* (see [9,

[7] Even though one has to decide whether relational information is encoded in the objects files—e.g., if $REL(a, b)$ holds then one needs to add this information in both the object-file relative to a and b—or outside them.

sect.3.10.1]). For instance, suppose to have the dimension *length* structured by the order relation $\leq$. The relation *shorter than* can be represented by a region in the space of the pairs of length-values, i.e. the region of all pairs (l_1, l_2) such that $l_1 \leq l_2$. Thus, an object x is shorter than an object y if the pair (length of x, length of y) belongs to this region. Gärdenfors seems to suggest that this approach is general enough to represent all the (binary) relations: "[a] relation between two objects can be seen as a simple case of a *pattern* of the location of the objects along a particular quality dimension" [9, p.93]. However, some structural relations, e.g., part-whole relations, seem to require really complex spaces founded on several quality dimensions (see [17]). More importantly, it is not clear to us how some relations like *eat* or *married to* can be reduced to intrinsic properties of relata. Similarly for *relational categories* [10], i.e., properties that are defined in relational terms, e.g., a carnivore is an animal that eats meat.

Secondly, in Definition 7, we have distinguished two possible ways of updating the information contained in a situation: abstraction and correlation. We have suggested that, intuitively, they correspond to two distinct types of processes: the first abstracts from already given data, the second allows to *indirectly* discover new data, a sort of indirect measurements. Contrast the following sentences:

i. "If a is scarlet, then a is red" and

ii. "If a is scarlet, then a is round."

In our model, (i) updates a situation s that contains, let say, $\langle a, \{\mathtt{scarlet}\}, \emptyset\rangle$ into a situation t that contains $\langle a, \{\mathtt{scarlet}, \mathtt{red}\}, \emptyset\rangle$ whereas (ii) is an update from s to a situation t' that contains $\langle a, \{\mathtt{scarlet}\}, \{\mathtt{round}\}\rangle$. Both these updates add information that was implicit, but they qualitatively differ because the update in (i) impacts the same domain while the one in (ii) impacts a different domain. Furthermore, our intuition is that (i) holds just in virtue of 'the scarletness of a', while (ii) holds in virtue of both 'the scarletness of a' and 'the roundness of a' (assuming that 'being scarlet' and 'being round' are both fully determinate properties). In terms of truth-makers (see [4]) this means that the two propositions 'a is scarlet' and 'a is red' share the same truth-maker ('the scarletness of a'). By contrast, the two propositions 'a is scarlet' and 'a is round' need two different truth-makers, i.e., only the second inference reveals the existence of an implicit truth-maker. This would suggest that the first kind of reasoning, the abstraction, is a purely mental process that does not need verification. By contrast, the second kind of reasoning, the correlation, needs additional validation in terms of truth-makers. Actually this provides a partial justification of the asymmetry between the required consistency of the conceptual knowledge vs. the possibility to have inconsistent correlations. An interesting question is whether it is possible to distinguish the two process in terms of inferential patterns, that is, we ask whether it is meaningful to define two kinds of implications, one corresponding to the sole updating by abstraction, and one corresponding to updating by correlation. We leave for future work the axiomatization of these two types of implications that, in our framework, can be characterized by two distinct reachability relations: one that only permits updates by abstraction, the other that only permits updates by correlations.

References

1. Allo, P., Mares, E.: Informational semantics as a third alternative? Erkenntnis 77(2), 167–185 (2012)
2. Anderson, A.R., Belnap Jr, N.D., Dunn, J.M., et al.: Entailment. The Logic of Relevance and Necessity, vol. II. Princeton University Press, Princeton (1992)
3. Armstrong, D.M.: A World of States of Affairs. Cambridge University Press (1997)
4. Armstrong, D.M.: Truth and Truthmakers. Cambridge University Press (2004)
5. Beall, J., Brady, R., Dunn, J.M., Hazen, A., Mares, E., Meyer, R.K., Priest, G., Restall, G., Ripley, D., Slaney, J., et al.: On the ternary relation and conditionality. Journal of philosophical logic 41(3), 595–612 (2012)
6. Borgo, S., Masolo, C.: Foundational choices in dolce. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. Springer, second edn. (2009)
7. Dunn, J.M., Restall, G.: Relevance logic. In: Handbook of philosophical logic, pp. 1–128. Springer (2002)
8. Fittingelvin: The logic of proofs, semantically. Annals of Pure and Applied Logic 132(1), 1–25 (2005)
9. Gärdenfors, P.: Conceptual spaces. The geometry of thought. MIT Press (2000)
10. Gentner, D., Kurtz, K.J.: Learning and using relational categories. In: Ahn, W.K., Goldstone, R.L., Love, B.C., Markman, A.B., Wolff, P.W. (eds.) Categorization inside and outside the laboratory, pp. 151–175. APA (2005)
11. Kahneman, D., Treisman, A., Gibbs, B.J.: The reviewing of object files: Object-specific integration of information. Cognitive psychology 24(2), 175–219 (1992)
12. Mares, E.: The nature of information: a relevant approach. Synthese 175(1), 111–132 (2010)
13. Mares, E.D.: Relevant logic: A philosophical interpretation. Cambridge University Press Press (2004)
14. Matthen, M.: Seeing, Doing, and Knowing. A Philosophical Theory of Sense Perce-tion. Oxford Univeristy Press (2005)
15. Meyer, R.K., Routley, R.: Classical relevant logics. Studia Logica 32(1), 51–66 (1973)
16. Pylyshyn, Z.W.: Things and Places. How the Mind Connects with the World. MIT Press (2007)
17. Rama Fiorini, S.: Similarity, Structure and Spaces: Representation of Part-Whole Relations in Conceptual Spaces. Ph.D. thesis, Universidade Federal do Rio Grande do Sul (2014)
18. Restall, G.: An introduction to substructural logics. Psychology Press (2000)
19. Sanford, D.H.: Determinates vs. determinables. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (Spring 2013 Edition)
20. Van Dalen, D.: Intuitionistic logic. In: Handbook of philosophical logic, pp. 225–339. Springer (1986)
21. Wansing, H.: The idea of a proof-theoretic semantics and the meaning of the logical operations. Studia Logica 64(1), 3–20 (2000)

Information-Theoretic Segmentation of Natural Language

Sascha S. Griffiths, Mariano Mora McGinity, Jamie Forth, Matthew Purver, and Geraint A. Wiggins

Cognitive Science Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, 10 Godward Square, London E1 4FZ United Kingdom
sascha.griffiths@qmul.ac.uk,
WWW home page: http://compling.eecs.qmul.ac.uk/

Abstract. We present computational experiments on language segmentation using a general information-theoretic cognitive model. We present a method which uses the statistical regularities of language to segment a continuous stream of symbols into "meaningful units" at a range of levels. Given a string of symbols—in the present approach, textual representations of phonemes—we attempt to find the syllables such as *grea* and *sy* (in the word *greasy*); words such as *in*, *greasy*, *wash*, and *water*; and phrases such as *in greasy wash water*. The approach is entirely information-theoretic, and requires no knowledge of the units themselves; it is thus assumed to require only general cognitive abilities, and has previously been applied to music. We tested our approach on two spoken language corpora, and we discuss our results in the context of learning as a statistical processes.

Keywords: Artificial Intelligence; Language Acquisition; Learning; Language Segmentation; Information Content

1 Introduction

The question which we address in this paper is whether language learning can be considered to be a statistical process. This has been an ongoing and fundamentally dividing issue in fields which consider language learning their subject matter.

We assume that language has several layers of structure. At the bottom we find the smallest units; in this paper, we start with phonemes, though our method is not restricted to this level of granularity. These smallest units build larger items of language structure:

- Phonemes build larger units such as syllables and morphemes.
- Morphemes and syllables build words. Words build phrases and phrases are the building blocks of sentences (or spoken utterances).
- These sentences or utterances go in turn to make up larger units such as paragraphs in text or speaker turns in speech.

We must assume some smallest unit such as phonemes in speech or graphemes in text as an entry point into the language system. The question which arises is how one can tell where one such unit above the phoneme or grapheme ends and another one begins. In natural language processing this task is generally called text segmentation for written language and speech segmentation for spoken language.

In the current paper, we assume that the phonemes are presented as one continuous stream - roughly equivalent to removing the white space from sentences in written text - and define our task as determining where a word or other linguistic unit begins or ends. This is similar to the task infants face when learning their first language, itself an open research question. Taking the title as an example, we need to identify that the word *segmentation* is composed of syllables, which are *seg*, *men*, *ta*, and *tion*, and morphemes, which are *segment* and *ation*. From there larger units need to be distinguished such as the words *segmentation*, *of* and *natural*. Longer utterances need to be split up into phrases, perhaps at various levels of granularity e.g. *natural language* and *segmentation of natural language*.

We present a computational approach to the segmentation problem [1] in which we rely entirely on the *information content* of a symbol within a language dataset. Our prediction is that information content will rise at the beginning of a segment and fall at the end of a segment. A similar assumption was used by Harris [2] for finding morpheme boundaries. We assume that this assumption should hold for segments at all levels of the linguistic hierarchy. However, for each level the nature and extent of this fall and rise will vary; but parameters of the model will vary predictably with levels of segmentation. In the following, we will present computational experiments which test this prediction on two datasets of natural language.

Our approach to the cognitive task of language processing therefore places emphasis on domain-independent principles, rather than taking a domain-specific approach as has been argued as appropriate for the case of language.

We outline our information-theoretic approach in the next section; we then present the methods used in this paper in detail. Our results and the discussion of these for computational experiments on language segmentation are presented in the following sections. In our conclusion we return to the question outlined above.

2 Information-Theoretic Speech Segmentation

Applying machine learning and pattern recognition methods to natural language has become a rich source of insights into language structure and theoretical issues of linguistics [3] and the learning of language [4]. While many linguists take the view that natural language requires domain-specific, innate structures (see e.g.[5]), this is debated; one of the tenets of cognitive linguistics is that language processing by humans is domain general and not domain specific. As Geoffrey Sampson puts it, language learning depends on ‘general human intelligence and abilities' [6, p. IX]. Using information theory as a framework for such an approach has been argued to be both cognitively and biologically plausible [7, 8].

It has often been proposed that language and music share certain properties. One can ask the same questions regarding the structure and processing of language as one can ask about music [9]. Indeed, there a number of objective similarities and differences between these two domains [10]. There seem to be shared resources in structural processing of language and music [10], in addition to the conceptual similarity which is that the building blocks of the structures are "cognitive objects" – i.e. percepts . In this paper, we assume that percepts in general can be processed via their statistical regularities in a given corpus. The computational model used here was created for purposes of melodic grouping [11].

The current research is situated within the wider context of the IDyOM and IDyOT frameworks. IDyOM [12] stands for Information Dynamics Of Music. It was developed on the basis of natural language processing methods and can divide melodies into perceptually correct segments using the statistical regularities in a corpus. Generally, however, we argue here that the framework can also still be used to segment a corpus of natural language data into syllables, words and phrases.

In previous work [13, 14] a cognitive architecture called IDyOT is outlined which builds on the principles of IDyOM. IDyOT [14] stands for Information Dynamics Of Thinking. The premise of both of these different incarnations of the underlying research framework is that grouping and boundary perception are central to cognitive science [15] and that the most cognitively plausible way of approaching this task is using Shannon's [16] information theory. Especially, we employ information content as introduced by MacKay [17].

IDyOT is based on the Global Workspace Theory [18]. In the long term it is predicted that IDyOT provides the basis for modelling creativity and eventually aspects of consciousness [14, 19]. In order of testing certain claims about the domain generality of the information dynamics approach embodied by IDyOM and IDyOT, we look at language segmentation to see whether the approach shown to be useful in music segmentation can be transferred back to language.

The IDyOM model [20] and corresponding software[1] were developed for the statistical modelling of music in the context of music perception and cognition research. However, one of the central features of the model is that it can also be used for other types of sequential data, as the principles on which it is based are cognitively inspired and meant to be general rather than domain specific [13]. The model presented in IDyOM relies on a pattern recognition theory of mind [21] which suggests that languages are learned by processing the underlying statistics of the positive data contained in stimuli. Although, at the present moment only representations of auditory stimuli have been studied, our conjecture is that any kind of perceptual data can be processed in this way. Wiggins [22]

[1] The software can be found at https://code.soundsoftware.ac.uk/projects/idyom-project/files.

gives initial indications that the model can extend to language segmentation, and we pursue that idea in more depth here.

As aforementioned, we take an information-theoretic approach here [16]. Predicting the next element in a sequence given the previous element is often called a Shannon Game [23, p. 191]. Here, we assume that both music and language can be modelled as a sequence of elements e from an alphabet $\mathcal{E}$. For each element e_i in e one can calculate its probability given the context – more specifically the preceding context e_1^{i-1}:

$$p(e_i|e_1^{i-1}) \tag{1}$$

There is good evidence that children use transition probabilities during language acquisition [24, p. 33], and this probability can be calculated by approximating on the basis of a context subsequence of finite length n, i.e. by using an n-gram model [25, pp. 845–847].

2.1 The IDyOM Model

IDyOM is a multidimensional variable-order Markov model. The multidimensionality within IDyOM is formalised as a multiple viewpoints system [26], where viewpoints can be either given basic types, or derived and combined from existing viewpoints to form new viewpoints revealing more abstract levels of structure. Predictions from individual variable-order viewpoint models are combined using an entropy-weighting strategy [20].

Two basic information-theoretic measures are central to IDyOM. *Information content* is the measure of unexpectedness—or surprisal to use the terminology of [27]—and *entropy* a measure of uncertainty.

1. information content (h) is *a measure of how unpredictable a [given unit] is given its context* [27];
2. entropy (H) is *the expected information content of an unseen event in a given context.*

More formally, in IDyOM these concepts are modelled as (1) and (2) below:

$$h(e_i|e_1^{i-1}) = log_2 \frac{1}{p(e_i|e_1^{i-1})} \tag{2}$$

$$H(e_1^{i-1}) = \sum_{e \in \mathcal{E}} p(e_i|e_1^{i-1}) h(e_i|e_1^{i-1}) \tag{3}$$

Entropy-based models such as these have been used in natural language learning in the past [28, pp. 21–37]. Given an n-gram model of $p(e_i|e_1^{i-1})$ which characterises the dataset in question, we can calculate h and H at all points in a sequence, and thereby find local falls and rises. Such falls and rises have previously been shown to correlate with the ending and beginning of structural units in music [11, 12] and language [22]. For the case of music it has also been demonstrated that this model outperforms rule-based approaches [29].

2.2 Segmentation of Natural Language

Segmentation of natural language has been a topic for computational psycholinguistics at least since 1990 [30]. However, it can still be regarded as a current problem in computational approaches to language learning (see for example [31–34]). Brent [35] classifies a number of approaches to natural language segmentation into three types of strategies. These are the *utterance-boundary strategy*, the *predictability strategy* and the *recognition strategy*. Our approach employs elements of the predictability strategy: we attempt to detect boundaries based on changes in the information-theoretic properties of the symbol sequences in question. In this way it is similar to, but simpler and more general than, methods such as that of Cohen and Adams [36], who use *boundary entropy* but combine it with other frequency measures via voting experts to segment words in a range of languages, or Sun, Shen and Tsou [37], who use *mutual information* but combine it with other statistical measures to segment Chinese characters into words.

This contrasts with approaches in which one tries to build grammars (or probabilistic models) of likely segment sequences (the predictability strategy), (e.g. for Finnish morphemes [38]), and with those in which one matches patterns of known words against the stream (the recognition strategy); in those approaches one needs to build up a lexicon first, either from external knowledge (e.g. [39]) or from incremental clustering (e.g. [40]). Our boundary detection strategy needs no knowledge of the lexicon or even of the fact that there are such concepts as syllables, words or phrases.

3 Methodology

Our experimental method requires two steps: firstly, building a statistical n-gram (IDyOM) model on the basis of which to calculate information content (entropy is left for future work); secondly, hypothesising boundaries based on local drops and rises in information content.

3.1 Calculating the Information Content

IDyOM has a range of model configurations intended to simulate different aspects of musical listening behaviour. The basic distinction concerns the data used to train an n-gram model: a model can be trained from a large dataset, modelling the learned experience of a listener and termed the Long Term model (LTM) in IDyOM's terminology; or from only the current sequence under consideration, trained incrementally for each utterance being predicted [26, 20, 41], modelling a listening experience in a specific context, and termed the Short Term model (STM). However, variations are possible: the LTM approach can be made dynamic by adjusting its probabilities based on the current sequence as it is observed (termed LTM+); and the STM and LTM models can be combined. This results in a total of five models:

STM model trained on stimuli only in a local context (i.e. notes of the melody or phonemes in the utterance currently being predicted);

LTM model trained on a large training corpus;
LTM+ as LTM, but model also learns from the current example;
Both combination of STM and LTM;
Both+ combination of STM and LTM+.

3.2 Segmentation

Our overall approach is to look for characteristic local contours in information content [22]—what Pearce et al. [15] call 'peak picking'. Rises in information content are signals of unexpectedness, and Wiggins [14] hypothesises that these should correlate with the beginnings of new segments; conversely, falls in information content are signals of predictability, which we expect to correlate with the endings of segments.

Our current method is extremely simple, checking only for a simple rise between successive data-points: the value at the current symbol e_i must exceed that at its immediate predecessor e_{i-1} by some specified amount. This amount is our only parameter, d; thus, a new segment begins if $h(e_i) - h(e_{i-1}) > d$. We evaluate performance using the κ statistic [42, 43], and set d to give the maximal value for κ (for a specific segment type) by testing all d over the interval $[0, 10]$.

3.3 Data

We test this method on two language corpora. The first dataset is a derived corpus[2] of the CHILDES corpus of child-directed adult English speech [44], collated and transcribed at the phoneme level for word segmentation experiments [45]. It contains 93,555 phoneme tokens which make up 33,377 words and 9,790 utterances; average utterance length is 3.4 words. A single viewpoint with phonemes as observed variables, denoted {phonemes}, is used as the basic IDyOM representation.

The second corpus is the TIMIT transcriptions [46], a dataset of spoken English sentences obtained for the purpose of automatic speech recognition model training, and transcribed at the level of sentences, words and phonemes. It contains 81,533 phoneme tokens which make up 20,756 words and 2,342 utterances; average utterance length is therefore 8.9 words. Again we use a simple phoneme viewpoint; as TIMIT also contains stress annotations (represented as primary, secondary, and no stress), this also allows us to construct a linked viewpoint formed of the cross-product of phonemes and stress {phonemes $\otimes$ stress}, and a two-viewpoint system combining both viewpoints {phonemes, phonemes $\otimes$ stress}.

To evaluate phrase-level segmentation, we used the Pattern parser [47] – neither TIMIT nor CHILDES contains phrase structure information. Automatic parses are noisy: we excluded cases where Pattern produced a parse which could not be mapped back onto the phonetic form of the utterance. Thus, our analysis on the phrase level only considers approximately half of the data for both corpora.

[2] `http://www.ling.ohio-state.edu/~melsner/resources/acl12data.tgz`

4 Results

We evaluate our segmentation model in terms of accuracy of boundary placement against the ground truth for each level—syllables, words and phrases—with accuracy assessed via both Kappa values (κ) and the F1-score (harmonic mean of precision and recall). Both κ and F1 are calculated on the basis of individual phoneme tokens, with the gold-standard annotation classifying only the first token in each segment as a boundary. We also examine the mean information content ($\bar{h}$), and optimum value of our segmentation parameter (d). $\bar{h}$ is the same in across all, as it is a property of the (phoneme-based) corpus and model and not of the evaluation.

4.1 CHILDES

The results for the CHILDES corpus segmentation into words and phrases is summarised in Table 1. Lower $\bar{h}$ values mean better predictability, as high $\bar{h}$ signifies more "surprisal" by a new element.

Table 1. Results for the CHILDES corpus for words (left) and phrases (right) using all five IDyOM configurations.

CHILDES							
		WORDS			PHRASES		
		{phonemes}			{phonemes}		
Model	$\bar{h}$	d	κ	F1	d	κ	F1
STM	5.74	5.39	0.39	0.46	6.06	0.52	0.57
LTM	3.42	1.59	0.58	0.71	2.87	0.54	0.63
LTM+	3.42	1.57	0.58	0.71	2.87	0.54	0.63
Both	3.67	1.21	0.54	0.7	3.02	0.55	0.65
Both+	3.66	1.8	0.54	0.7	3.05	0.56	0.65

Performance is reasonable at word level, with F1 around 0.7 and κ approaching 0.6. The performance of the STM is considerably lower than other models, as might be expected; we note that $\bar{h}$ is considerably higher for the STM, indicating worse fit. The d parameter is therefore also correspondingly higher – and takes longer to find – for the STM. The lowest d for words is found in the Both model and LTM and LTM+ for the phrase segmentation.

In terms of both F1 and κ, the LTM and LTM+ are the best models for the word discovery task. In the phrase segmentation task, we find that the Both and Both+ models do marginally better than in the word segmentation task with respect to κ, but with worse performance with respect to F1. The apparent improvement of results for the STM may be due to the lower number of segments which need to be predicted. The improvement in performance by the short term model also leads to improvements in the Both and Both+ configuration as these are combinations of STM and LTM.

In comparison to previous work on the same dataset, our best configuration (LTM) still performs slightly worse with respect to F1-scores in the word segmentation task. While Elsner et al. [45] obtained an F1-score of 0.8, our best F1 score was 0.71. We also checked the baselines with respect to a random segmentation, a segmentation which assumes every symbol is a boundary and a segmentation which assumes no boundaries. In each case, the κ will be 0 as expected with low F1-scores.

4.2 TIMIT

Table 2 shows the results for the TIMIT corpus. The results for the syllable segmentation task are very comparable for all measures with those reported by Wiggins [22]. As with the CHILDES dataset, the STM shows higher values for $\bar{h}$, with the LTM and Both models showing better performance. κ and F1 are almost the same for the STM for all configurations. Therefore, the STM, for which the $\bar{h}$ is determined based on isolated utterances, seems not to be a good model for this task.

Table 2. Summary of results for the TIMIT corpus for words (left) and phrases (right) using all five configurations of IDyOM.

TIMIT										
		SYLLABLES			WORDS			Phrases		
Model	$\bar{h}$	{phonemes}			{phonemes}			{phonemes}		
		d	κ	F1	d	κ	F1	d	κ	F1
STM	5.46	2.43	0.11	0.26	3.95	0.17	0.24	6.96	0.39	0.42
LTM	3.55	1.29	0.47	0.65	1.96	0.58	0.69	4.50	0.41	0.47
LTM+	3.54	1.15	0.47	0.66	1.95	0.56	0.69	4.40	0.41	0.47
Both	3.68	1.26	0.45	0.64	1.65	0.55	0.67	4.44	0.42	0.48
Both+	3.67	1.05	0.45	0.65	1.94	0.56	0.69	4.52	0.42	0.48
Model	$\bar{h}$	{phonemes $\otimes$ stress}			{phonemes $\otimes$ stress}			{phonemes $\otimes$ stress}		
		d	κ	F1	d	κ	F1	d	κ	F1
STM	6.04	3.09	0.11	0.22	3.72	0.18	0.26	7.36	0.39	0.42
LTM	3.73	1.08	0.48	0.67	2.17	0.60	0.70	4.48	0.42	0.48
LTM+	3.72	1.10	0.48	0.67	2.05	0.60	0.71	4.84	0.42	0.48
Both	3.85	1.2	0.46	0.66	2.16	0.58	0.68	4.11	0.42	0.49
Both+	3.84	1.27	0.47	0.65	2.15	0.58	0.69	4.09	0.42	0.49
Model	$\bar{h}$	{phonemes, phonemes $\otimes$ stress}			{phonemes, phonemes $\otimes$ stress}			{phonemes, phonemes $\otimes$ stress}		
		d	κ	F1	d	κ	F1	d	κ	F1
STM	6.01	2.96	0.11	0.23	3.64	0.18	0.26	7.03	0.39	0.42
LTM	3.72	1.10	0.49	0.67	2.12	0.61	0.71	3.93	0.42	0.49
LTM+	3.71	1.14	0.49	0.67	2.13	0.61	0.71	4.49	0.42	0.48
Both	3.85	1.07	0.47	0.66	2.01	0.58	0.69	4.12	0.43	0.49
Both+	3.84	1.10	0.47	0.65	1.92	0.58	0.69	4.02	0.42	0.49

Generally, one can see a trend that the LTM, LTM+, Both and Both+ models perform better if they receive more information, i.e. in the {phonemes $\otimes$ stress} and {phonemes, phonemes $\otimes$ stress} show marginally better performance. The smallest values for d are found in the viewpoints {phonemes$\otimes$ stress}.

There seems no improvement in performance when moving from the LTM to LTM+ or the Both model variants (in all configurations except the {phonemes} condition, where the LTM+ shows slightly higher F1-score). Overall, the best configuration for the word segmentation task is the LTM in the {phonemes, phonemes $\otimes$ stress} condition.

Segmentation at Different Levels For the word segmentation task, the optimal setting of d is higher than that for the syllable segmentation task, for all configurations. This corresponds with intuitive expectation, as one needs to predict fewer segments. In all configurations, the LTM and LTM+ still show better performance than the STM, Both and Both+; overall accuracy is slightly improved over the syllable segmentation task, with F1 scores over 0.7. The LTM+ is the best configuration overall in the {phonemes, phonemes$\otimes$ stress} condition. The STM perhaps also shows some improvement here, with κ values marginally higher.

In the phrase segmentation task, again, the optimum d increases relative to word and syllable tasks, as even fewer segments need to be predicted. Performance in terms of κ and F1-scores is, however, much lower for phrase discovery than for syllables and words. Thus, with regard to this measure the performance on the TIMIT data is less effective.

The κ values and F1 scores for the STM, however, are considerably higher for this task. The STM does, however, not benefit from the additional information which it gets in the {phonemes $\otimes$ stress} and {phonemes, phonemes $\otimes$ stress} conditions.

In all configurations, the LTM, LTM+, Both and Both+ models show worse performance in the phrase segmentation task with respect to κ and F1. Also, there is little difference in the performance of these four models. The Both is the best configuration overall in the {phonemes, phonemes $\otimes$ stress} condition with respect to the κ value.

As expected our results are very similar to those reported in Wiggins [22] for syllable segmentation. We also checked the baselines with respect to a random segmentation, a segmentation which assumes every symbol is a boundary and a segmentation which assumes no boundaries. In each case, the κ will be 0 as expected with low F1-scores.

4.3 Overall Segmentation Performance

Figures 1 and 2 show the variation of κ with the information content threshold parameter d for each corpus, illustrating the process of determining optimum d values.

The LTM and Both model variants show a general pattern for syllables and words: a gradual improvement leading up to a peak in performance (defining

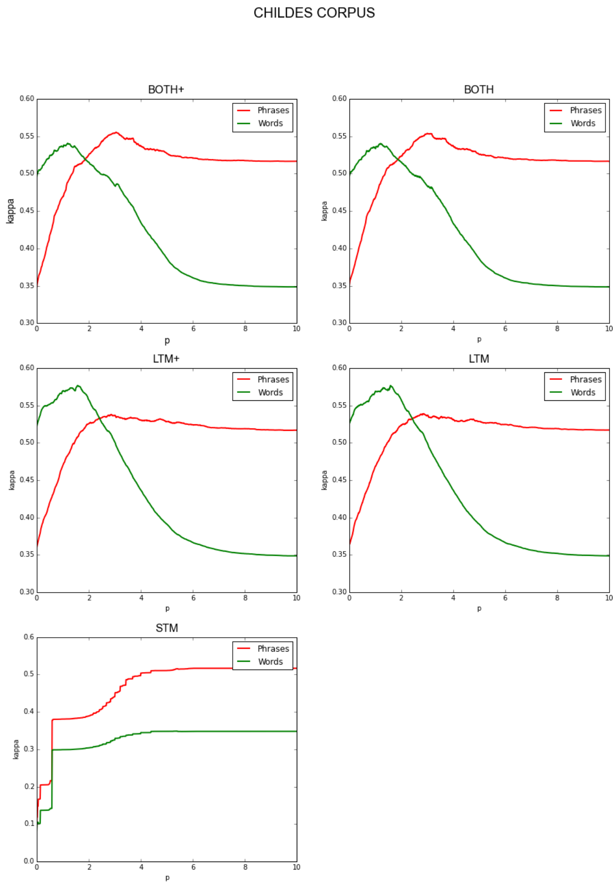


Fig. 1. CHILDES corpus κ vs parameter d for words and phrases.

optimum d), after which performance drops off again. This optimum value of d is higher as we move to longer, higher-level segments (from syllables to words, and from words to phrases): larger changes in information content correspond to segment boundaries at different levels. However, the phrase segmentation curve shows less of a peak: performance reaches a level at which it stays. This suggests that as long as d is large enough one finds segments which have a high probability of coinciding with phrase boundaries. The plateau in the curve after the peak can be explained as an effect of our segmentation method coding the beginning of an utterance as a given start symbol. This is similar to the approach of Elsner et al [45]. Thus, once the method stops oversegmenting at low d it finds the optimum d and afterwards continues to agree on those given symbols at higher d values.

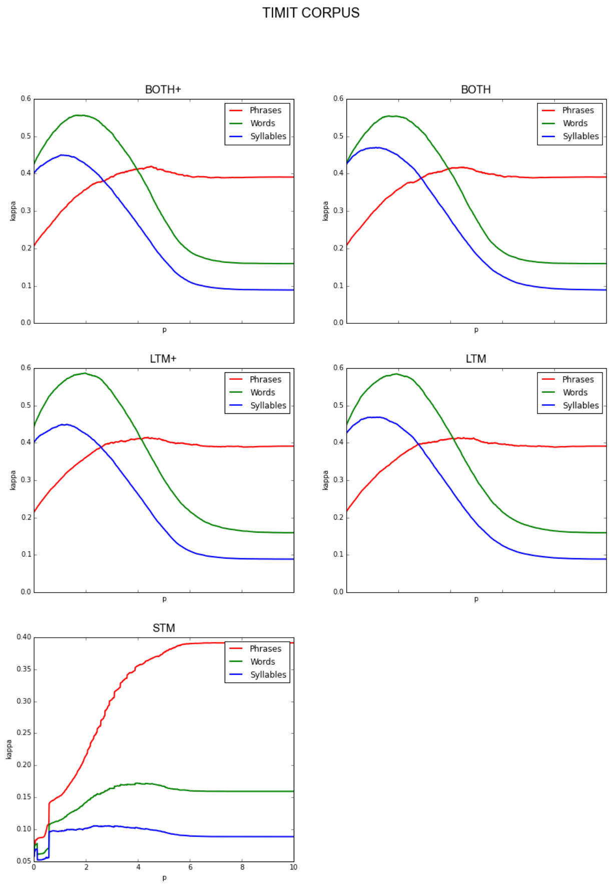


Fig. 2. TIMIT corpus κ vs parameter d for syllables, words and phrases.

The STM shows particularly bad performance initially but then the plots show a sudden leap in performance. This is true for all configurations on both corpora. Thus, short term segmentation seems to require a certain threshold to show any noticeable segmentation performance. Exposure to isolated utterances is insufficient to learn the distributional regularities of language.

5 Discussion & Conclusion

Landis and Koch [48] characterise a $\kappa \in [0.4, 0.6]$ as "moderate". Thus, most of the results reported here show a moderate success. The results reported for the CHILDES corpus with respect to phrases is slightly higher and thus falls into the "substantial" category. However, one has to again note, that results regarding the syntactic units are to be taken with caution. There is less to predict and less to agree on. Therefore, one would expect a higher agreement between ground-truth and segmentation.

The long term model shows better performance than the short term model. In effect, these two model a listeners knowledge of language (LTM) and a current listening experience (STM). It is to be expected that there is little result to be expected from learning from a single listening experience. Thus, the results with respect to the differences in LTM and STM show that a long term learning from raw stimulus is possible.

The TIMIT data also indicates that learning is improved if stress information can be included. Though, the differences are small, the inclusion of stress in the viewpoints selected for predicting the next phoneme do improve the results. The differences reported here are minor, though.

The present contribution took a strong view of statistical language learning. We claimed that it would be possible to predict syllable, word and phrase boundaries from a raw stimulus without having explicit information about these units encoded in the method. We succeeded in the sense that our results indicate that this is indeed possible. In future work, we plan to explore further in what way the inclusion of different viewpoints improves the results.

6 Acknowledgments

The research reported in this is supported by ConCreTe: the project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

1. Elman, J.L.: Computational approaches to language acquisition. In Brown, K., ed.: Encyclopedia of Language and Linguistics. Volume 2. Second edn. Elsevier, Oxford (2006)
2. Harris, Z.S.: From phoneme to morpheme. Language **31**(2) (1955) pp. 190–222
3. Lappin, S., Shieber, S.M.: Machine learning theory and practice as a source of insight into universal grammar. Journal of Linguistics **43**(2) (2007) 393–427
4. Clark, A., Lappin, S.: Linguistic Nativism and the Poverty of the Stimulus. Wiley-Blackwell, Oxford (2011)
5. Chomsky, N.: New Horizons in the Study of Language and Mind. Cambridge University Press, Cambridge (2000)
6. Sampson, G.R.: The Language Instinct Debate. Continuum, London (2005)
7. Wallace, R.: Cognition and biology: perspectives from information theory. Cognitive Processing **15**(1) (February 2014) 1–12
8. Rohrmeier, M., Zuidema, W., Wiggins, G.A., Scharff, C.: Principles of structure building in music, language and animal song. Philosophical Transactions of the Royal Society of London B: Biological Sciences **370**(1664) (2015) 20140097
9. Jackendoff, R., Lerdahl, F.: The capacity for music: What is it, and what's special about it? Cognition **100**(1) (2006) 33–72

10. Patel, A.D.: Music, Language, and the Brain. Oxford University Press, Oxford (2008)
11. Pearce, M.T., Wiggins, G.A.: The information dynamics of melodic boundary detection. In: Proceedings of the Ninth International Conference on Music Perception and Cognition, Bologna (2006) 860–867
12. Pearce, M.T., Müllensiefen, D., Wiggins, G.A.: Melodic grouping in music information retrieval: New methods and applications. In: Advances in music information retrieval. Springer, Berlin (2010) 364–388
13. Wiggins, G.A.: The mind's chorus: creativity before consciousness. Cognitive Computation **4**(3) (2012) 306–319
14. Wiggins, G.A., Forth, J.: IDyOT: A computational theory of creativity as everyday reasoning from learned information. In Besold, T.R., Schorlemmer, M., Smaill, A., eds.: Computational Creativity Research: Towards Creative Machines. Volume 7 of Atlantis Thinking Machines. Atlantis Press (2015) 127–148
15. Pearce, M.T., Müllensiefen, D., Wiggins, G.A.: The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. Perception **39**(10) (2010) 1365–1389
16. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review **5**(1) (1948) 3–55
17. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)
18. Baars, B.J.: A cognitive theory of consciousness. Cambridge University Press, Cambridge (1993)
19. Wiggins, G.A., Tyack, P., Scharff, C., Rohrmeier, M.: The evolutionary roots of creativity: Mechanisms and motivations. Philosophical Transactions of the Royal Society of London B: Biological Sciences **370**(1664) (2015) 20140099
20. Pearce, M.T.: The construction and evaluation of statistical models of melodic structure in music perception and composition. PhD thesis, City University London (2005)
21. Kurzweil, R.: How to create a mind: The secret of human thought revealed. Penguin, London (2012)
22. Wiggins, G.A.: "I let the music speak": Cross-domain application of a cognitive model of musical learning. In Rebuschat, P., Williams, J., eds.: Statistical Learning and Language Acquisition. Mouton de Gruyter, Amsterdam, NL (2012) 463 – 494
23. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge, MA (1999)
24. Ambridge, B., Lieven, E.V.M.: Child Language Acquisition: Contrasting Theoretical Approaches. Cambridge University Press, Cambridge (2011)
25. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Third edn. Prentice Hall International (2013)
26. Conklin, D., Witten, I.H.: Multiple viewpoint systems for music prediction. Journal of New Music Research **24**(1) (March 1995) 51–73
27. Mahowald, K., Fedorenko, E., Piantadosi, S.T., Gibson, E.: Info/information theory: Speakers choose shorter words in predictive contexts. Cognition **126**(2) (2013) 313–318
28. Charniak, E.: Statistical Language Learning. MIT Press, Cambridge, MA (1996)
29. Pearce, M.T., Wiggins, G.A.: Expectation in Melody: The Influence of Context and Learning. Music Perception **23**(5) (2006) 377–405
30. Elman, J.L.: Finding structure in time. Cognitive Science **14**(2) (June 1990) 179–211

31. Elsner, M., Goldwater, S., Feldman, N., Wood, F.: A joint learning model of word segmentation, lexical acquisition, and phonetic variability. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2013)
32. Fourtassi, A., Börschinger, B., Johnson, M., Dupoux, E.: Why is English so easy to segment? In: Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL), Sofia, Bulgaria, Association for Computational Linguistics (August 2013) 1–10
33. Pate, J.K., Johnson, M.: Syllable weight encodes mostly the same information for english word segmentation as dictionary stress. In: EMNLP, Doha, Qatar (2010) 844–853
34. Çöltekin, Ç.: Units in segmentation: A computational investigation. In: Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning, Lisbon, Portugal, Association for Computational Linguistics (September 2015) 55–64
35. Brent, M.R.: Speech segmentation and word discovery: a computational perspective. Trends in Cognitive Sciences **3**(8) (August 1999) 294–301
36. Cohen, P., Adams, N.: An algorithm for segmenting categorical time series into meaningful episodes. In: Advances in Intelligent Data Analysis, 4th International Conference, Cascais, Portugal (2001) 198–207
37. Sun, M., Shen, D., Tsou, B.K.: Chinese word segmentation without using lexicon and hand-crafted training data. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics. (1998) 1265–1271
38. Virpioja, S., Turunen, V.T., Spiegler, S., Kohonen, O., Kurimo, M.: Empirical comparison of evaluation methods for unsupervised learning of morphology. Traitement Automatique des Langues **52**(2) (2011) 45–90
39. Sproat, R., Shih, C., Gale, W., Chang, N.: A stochastic finite-state word-segmentation algorithm for Chinese. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics. (1994) 66–73
40. Gold, K., Scassellati, B.: Audio speech segmentation without language-specific knowledge. In: Proceedings of the 28th Annual Conference of the Cognitive Science Society, Vancouver (2006) 1370–1375
41. Wiggins, G.A., Pearce, M.T., Müllensiefen, D.: Computational modelling of music cognition and musical creativity. In Dean, R., ed.: The Oxford Handbook of Computer Music. Oxford University Press (2009) 383–420
42. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement **20**(1) (April 1960) 37–46
43. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics **22**(2) (1996) 249–254
44. MacWhinney, B.: CHILDES Project: Tools for analyzing talk. 3rd Edition. Vol. 2: The Database. Lawrence Erlbaum Associates, Mahwah, NJ (2000)
45. Elsner, M., Goldwater, S., Eisenstein, J.: Bootstrapping a unified model of lexical and phonetic acquisition. In: Proceedings of the 50th Annual Meeting of the Association of Computational Linguistics. (2012)
46. Zue, V., Seneff, S., Glass, J.: Speech database development at MIT: TIMIT and beyond. Speech Communication **9** (1990) 351–356
47. De Smedt, T., Daelemans, W.: Pattern for Python. Journal of Machine Learning Research **13**(1) (2012) 2063–2067
48. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1) (1977) 159–174

Pattern-Recognition: a Foundational Approach

Agnese Augello[3], Salvatore Gaglio[1,3], Gianluigi Oliveri[2,3], and Giovanni Pilato[3]

[1] DICGIM - Università di Palermo
Viale delle Scienze, Edificio 6 - 90128, Palermo - ITALY
[2] Dipartimento di Scienze Umanistiche - Università di Palermo
Viale delle Scienze, Edificio 12 - 90128, Palermo - ITALY
[3] ICAR - Italian National Research Council
Viale delle Scienze - Edificio 11 - 90128 Palermo, Italy
{gaglio,oliveri}@unipa.it
{agnese.augello,giovanni.pilato}@cnr.it

Abstract. This paper aims at giving a contribution to the ongoing attempt to turn the theory of pattern-recognition into a rigorous science. In this article we address two problems which lie at the foundations of pattern-recognition theory: (i) What is a pattern? and (ii) How do we come to know patterns? In so doing much attention will be paid to tracing a non-arbitrary connection between (i) and (ii), a connection which will be ultimately based on considerations relating to Darwin's theory of evolution.

1 Introduction

As is well known, the main aim of pattern-recognition theory is to determine whether, and to what extent, what we call 'pattern-recognition' can be accounted for in terms of automatic processes. From this it follows that two of its central problems are how to: (i) describe and explain the way humans, and other biological systems, produce/discover and characterize patterns; and how to (ii) develop automatic systems capable of performing pattern recognition behaviour.

Having stated these important facts, we need to point out that at the foundations of pattern-recognition theory there are two more basic questions which we can formulate in the following way: (a) what is a pattern? (b) how do we come to know patterns? And it is clear that, if we intend to develop a science of pattern recognition able to provide a rigorous way of achieving its main aim, and of pursuing its central objects of study, it is very important to answer questions (a) and (b).

After having addressed the problem of providing a definition of the concept of pattern in §2, a case-study of a particular type of finite geometry is discussed in §3 in the hope that by so doing we might obtain a rigorous characterization of the concept of mathematical pattern.

Section 4 is then dedicated to the examination of some of the interesting lessons that can be learned from the case-study in §3. In particular, one of these has to do with the characterization of the concept of mathematical pattern in terms of mathematical structure; and another concerns the possibility of generalizing the view of mathematical patterns as structures to patterns belonging to fields different from mathematics.

Finally, sections 5 and 6, armed with the notion of pattern developed so far, bring the paper to a close by addressing question (b) above: how do we come to know patterns?

2 Searching for a definition

A potentially fruitful approach to the problem 'What is a pattern?' is that of Daniel Dennett. For Dennett, who in his discussion of the concept of pattern is concerned with issues belonging to the philosophy of mind and action,

> [W]e are to understand the pattern to be what Anscombe called the "order which is there" in the rational coherence of a person's set of beliefs, desires, and intentions. [[6], §IV, p. 47.]

However, although taking into account final causes, beliefs and intentions often can both reveal an order existing among a certain individual's actions and explain his behaviour in terms of giving an account not only of how, but also of why he did what he did, it must be admitted that talking about 'the order which is there in the rational coherence of a person's set of beliefs, desires, and intentions' is too vague to shed light on the notion of pattern. This is, in particular, the case when the accounts of the order which is there ... etc. are several, radically differ from one another, and all seem to agree with the facts.

Moreover, since patterns do not occur only within the context of human actions and beliefs, what happens when we are dealing with patterns displayed by crystals of snowflakes? Of course, also in the case of crystals of snowflakes (see Fig. 1)

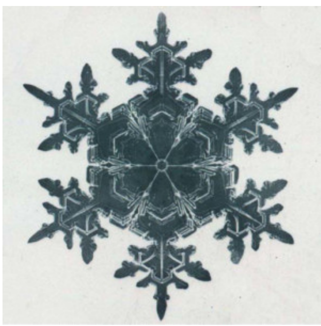


Fig. 1. A crystal of a snowflake

the patterns they display are related to the order in which the components of the crystals of snowflakes are to one another. But, whereas in the case of the crystals of snowflakes, if we use a microscope, we can actually see them, when we turn to actions the verb 'seeing' appears to let us down. For an action, in contrast to the crystal of a snowflake, is not just a brute physical fact and, therefore, the order manifested by a sequence of actions 'which is there in the rational coherence of a person's set of beliefs, desires, and intentions' is not something we can perceive by simply keeping our eyes wide open, and using instruments of observation.

This is an important point, because, if there is some truth in Dennett's definition of pattern, it means that what we might call 'brute seeing', that is, the mere act of representing within visual perceptual space a given input — like what happens with a photo-camera when we take a picture — cannot provide a satisfactory account of what happens when we perceive a pattern.

Therefore, if we intend to give an account of perceiving a pattern which is in accord with Dennett's definition, we should appeal to a concept of seeing which is much richer than brute seeing. A good candidate for such a concept of seeing is the concept that in the *Philosophical Investigations* Wittgenstein famously called 'seeing something as' or 'aspect seeing'.[4]

Notice, for example, that in seeing something as a square the perception of the square-pattern is not brute, because it presupposes, among other things, that the observer has a grasp of the concept of square.

However, independently of questions relating to the nature of the 'order which is there . . . ' in different contexts, and any consideration concerning what we must mean by 'seeing an aspect' or 'perceiving a pattern', Dennett proposes a very interesting general test for the existence of patterns. Basing himself on Chaitin's definition of randomness:

> A series of numbers is random if the smallest algorithm capable of specifying it to a computer has about the same number of bits of information as the series itself. [[2], p. 48.]

Dennett asserts that:

> A pattern exists in some data — is real — if *there is* a description of the data that is more efficient than the bit map, whether or not anyone can concoct it. [[6], §II, p. 34.]

Although that offered by Dennett is a very plausible criterion which, in some cases, reveals the presence of patterns in a data-set, it is not specific to them. To see this consider the definite description 'The satellite of the Earth'. Such a definite description certainly provides an enormous compression of data with respect to the bit map of a computer visual representation of the Moon. But, it is a description which uniquely identifies an object not a pattern/structure.

Lastly, the phenomenon of seeing something as a square appears to hint at a structural feature of perception, where the pattern perceived is that of a square. In fact if, by zooming in or out on the object we perceive as a square, we change

[4] See on this [17], Part II, §XI, pp. 213^e–214^e.

(within a certain range) the magnitude of the picture of the object, we would still see the object as a square.

The structural character of the pattern perceived is particularly evident in the case of the crystal of a snowflake. Indeed, when we observe a crystal of a snowflake through a microscope or when we look at a photograph or at an artist's accurate impression of that very crystal of a snowflake, etc. in spite of being presented in each single case with a different object — the actual crystal, the photograph of the crystal, and the artist's accurate impression of the crystal — we recognize the presence of the same pattern *in* all these objects. Of course, the next question is 'What is a structural feature of an object?' or, in more general terms, 'What is a structure?' The latter is, indeed, the problem which is going to be at the heart of the next section.

3 Mathematical Patterns. A case study

If we are presented with objects **a** and **b** (see Figures 2 and 3), it is very difficult to see what interesting mathematical feature they might have in common, if any, let alone that they exemplify the same mathematical pattern.

A	B	C	D	E	F	G
B	C	D	E	F	G	A
D	E	F	G	A	B	C

Fig. 2. Object **a**

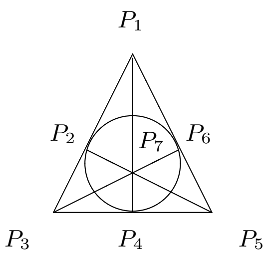


Fig. 3. Object **b**

Indeed, whereas object **a** is a 3×7 matrix whose elements are the first seven letters of the Italian alphabet, object **b** is a geometrical entity consisting of 7 lines and 7 points. The lines of object **b** are: the sides of the triangle drawn in Figure 3, the bisecting segments, and the inscribed circle. On the other hand, the 7 points are the points of intersection of three lines.

However, the situation radically changes if we introduce the following formal system T with the appropriate interpretations.

Let a formal system T be given such that the language of T contains a primitive binary relation 'x belongs to a set X' ($x \in X$), and its inverse 'X contains an element x' ($X \ni x$).

Furthermore, let us assume that D is a set of countably many undefined elements $a_1, a_2, \ldots$; call 'm-set' a subset X of D; and consider the following as the axioms of T:

Axiom 1 If x and y are distinct elements of D there is at least one m-set containing x and y;
Axiom 2 If x and y are distinct elements of D there is not more than one m-set containing x and y;
Axiom 3 Any two m-sets have at least one element of D in common;
Axiom 4 There exists at least one m-set.
Axiom 5 Every m-set contains at least three elements of D;
Axiom 6 All the elements of D do not belong to the same m-set;
Axiom 7 No m-set contains more than three elements of D.[5]

Now, the language of T contains two different sorts of variables: $x, y, \ldots$ and $X, Y, \ldots$ Let us assume that the variables $x, y, \ldots$ range over $\mathrm{D}_1 = \{\mathrm{A}, \ldots, \mathrm{G}\}$; and that the variables $X, Y, \ldots$ range over D_1^*, where D_1^* is a set whose elements are the subsets of D_1 the elements of which appear in the columns of the matrix in Figure 1, that is:

$$\mathrm{D}_1^* = \{\{\mathrm{A},\mathrm{B},\mathrm{D}\},\{\mathrm{B},\mathrm{C},\mathrm{E}\},\{\mathrm{C},\mathrm{D},\mathrm{F}\},\{\mathrm{D},\mathrm{E},\mathrm{G}\},\{\mathrm{E},\mathrm{F},\mathrm{A}\},\{\mathrm{F},\mathrm{G},\mathrm{B}\},\{\mathrm{G},\mathrm{A},\mathrm{C}\}\}.$$

(The elements of D_1^* are the m_1-sets.)

It turns out that $\mathrm{D}_1 \cup \mathrm{D}_1^*$ is the domain of the model of T represented in Figure 2. To see this, using the interpretation suggested above, it is sufficient to verify that **Axioms 1 − 7** are true of the matrix in Figure 2. We call such a model '$\mathcal{M}_1(\mathrm{T})$.'

On the other hand, if we change interpretation making: (a) the variables $x, y, \ldots$ range over $\mathrm{D}_2 = \{\mathrm{P}_1, \ldots, \mathrm{P}_7\}$, where $P_1, \ldots, P_7$ are the 7 distinct points indicated in Figure 3; and (b) the variables $X, Y, \ldots$ range over D_2^* whose elements are the m_2-sets, that is, the sets of three P_i points, for $1 \leqslant i \leqslant 7$, lying on the sides, the bisectrices, and the circle inscribed in the triangle represented in Figure 3: $D_2^* = \{\{P_6, P_2, P_4\}, \{P_2, P_7, P_5\}, \{P_5, P_4, P_3\}$, $\{P_4, P_7, P_1\}, \{P_3, P_7, P_6\}, \{P_3, P_2, P_1\}, \{P_1, P_6, P_5\}\}$; we have that $\mathrm{D}_2 \cup \mathrm{D}_2^*$ is also the domain of a model of T, a model represented in Figure 3. We call such a model '$\mathcal{M}_2(\mathrm{T})$.'

To show that $\mathcal{M}_2(\mathrm{T})$ is a model of T, it is sufficient, using the interpretation just provided, to check that **Axioms 1 − 7** are true of the object represented in Figure 3.

If we, now, compare $\mathcal{M}_1(\mathrm{T})$ with $\mathcal{M}_2(\mathrm{T})$, we realize that, among other things: (1) $(\mathrm{D}_1 \cup \mathrm{D}_1^*) \cap (\mathrm{D}_2 \cup \mathrm{D}_2^*) = \emptyset$; (2) the elements of $\mathrm{D}_1 \cup \mathrm{D}_1^*$ are not homogeneous

[5] These axioms have been taken, with some minor alterations, from [16], §2.10, p. 30.

with the elements of $\mathrm{D}_2 \cup \mathrm{D}_2^*$; and that (3) $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$, are isomorphic to each other.

With regard to point (3) above, we notice that if f is the function $f : D_1 \to D_2$ such that:

$$\begin{aligned} f(A) &= P_6, \\ f(B) &= P_2, \\ f(C) &= P_5, \\ f(D) &= P_4, \\ f(E) &= P_7, \\ f(F) &= P_3, \\ f(G) &= P_1; \end{aligned}$$

and g is the function $g : D_1^* \to D_2^*$ such that:

$$\begin{aligned} g(X) &= g(\{x_i, x_j, x_k\}) \\ &= \{f(x_i), f(x_j), f(x_k)\} \end{aligned}$$

for $1 \leqslant i \leqslant j \leqslant k \leqslant 7$, then f induces a bi-univocal correspondence between D_1 and D_2, whereas g induces a bi-univocal correspondence between the set D_1^* (of m_1-sets) and the set D_2^* (of m_2-sets).

Now, it is clear that the function ψ, where $\psi : D_1 \bigcup D_1^* \to D_2 \bigcup D_2^*$ such that:

$$\psi(\lambda) = \begin{cases} f(x) & \text{if} \quad \lambda = x \\ g(X) & \text{if} \quad \lambda = X \end{cases}$$

shows that $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$ are isomorphic to one another. In fact, ψ induces a bi-univocal correspondence between $D_1 \bigcup D_1^*$ and $D_2 \bigcup D_2^*$ preserving the two (primitive) relations $\in$ and $\ni$, that is:

$$\begin{aligned} x \in X &\text{ iff } \psi(x) \in \psi(X) \\ X \ni x &\text{ iff } \psi(X) \ni \psi(x). \end{aligned}$$

The case relative to the existence of two isomorphic models $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$ of T brings out very clearly that the pattern described by the axioms and theorems of T is independent of the nature of the objects present in $\mathrm{D}_1 \cup \mathrm{D}_1^*$ (the first seven letters of the alphabet plus ...), and in $\mathrm{D}_2 \cup \mathrm{D}_2^*$ (the seven distinct points highlighted in Figure 3 plus ...). The pattern described by the axioms and theorems of T is an abstract mathematical structure realized by/present in both $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$.

At this point a legitimate problem that might arise is ‘How is the structure common to $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$ given to us?’ and another is ‘What sort of thing is this structure?’ Let us address the second question first.

A structure/pattern is an ordered pair the first element of which is the domain of the structure — in our case $\mathrm{D}_1 \cup \mathrm{D}_1^*$ or $\mathrm{D}_2 \cup \mathrm{D}_2^*$ — and whose second element is a set of relations defined on this domain — in our case the relations are $\in$ and $\ni$ — relations the basic properties of which are implicitly defined by the axioms.

With regard to the question concerning the reality of the structure instantiated by $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$, consider that if objects **a** and **b** exist and, therefore, are real then also the structure they realize exists and, therefore, is real.

The answer to the first question is more complicated, because there is no unique way in which a pattern, even a mathematical one, becomes salient to an observer. However, it is certainly the case that necessary conditions for seeing a certain object as the realization of the pattern/mathematical structure we have been talking about in this paper are: (1) the observer's acquaintance with object **a** or with object **b**, (2) the observer's knowledge of T, and (3) the observer's knowledge of the appropriate interpretation of T.

Another non-psychological way of addressing the question 'How is the structure common to $\mathcal{M}_1(\mathrm{T})$ and $\mathcal{M}_2(\mathrm{T})$ given to us?' consists in transforming object **b** into an object **c** isomorphic to object **b** such that object **c** is clearly isomorphic to object **a** (see on this Figures 4-6). For, since isomorphism is a transitive relation this would show that object **b** is isomorphic to object **a**.

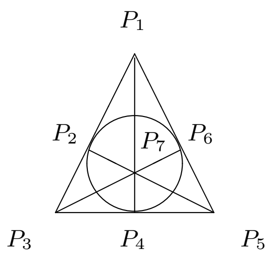


Fig. 4. Object **b**

$$\begin{array}{ccccccc} P_6 & P_2 & P_5 & P_4 & P_3 & P_3 & P_1 \\ P_2 & P_7 & P_4 & P_7 & P_7 & P_2 & P_6 \\ P_4 & P_5 & P_3 & P_1 & P_6 & P_1 & P_5 \end{array}$$

Fig. 5. Object **c**

$$\begin{array}{ccccccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} \\ \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} & \mathrm{A} \\ \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} & \mathrm{A} & \mathrm{B} & \mathrm{C} \end{array}$$

Fig. 6. Object **a**

Notice that the procedure illustrated above is non-psychological, because, although we always assume that the observer finds himself in 'normal conditions', the procedure acts on the objects observed and not on the observer. Indeed, in constructing object **c**, we have simply 'opened' **b** in such a way as to obtain a 3×7 matrix which has as columns the sets of points contained in each line. (The order in which the points actually occur in the respective lines is not relevant for our purposes.)

Several are the things that interest us in this example. We shall briefly comment on some of them in the next section.[6]

4 Some comments on the case study

Among the necessary conditions for 'seeing a certain object as ... ' that we have mentioned in the previous section the first is the observer's acquaintance with object **a** and/or with object **b**. Now, the possibility for an observer of being acquainted with **a** and/or **b** depends, among other things, on:

> [T]he particular pattern-recognition machinery hard-wired in our visual systems — edge detectors, luminance detectors, and the like ... [T]he very same data (the very same streams of bits) presented in some other format might well yield no hint of pattern to us ([6], p. 33).

Other important conditions upon which the possibility of an observer being acquainted with **a** and **b** depends are the size and position of objects **a** and **b** relative to the observer. To see this, immagine that objects **a** and **b** are microscopic and the observer is an average human being without any support provided by technology; or that **a** and **b** are too far from the observer to be surveyable by him, etc.

Secondly, in the absence of the formal system T and of the relevant interpretations of T, the observer cannot see the pattern/structure instantiated by **a** and **b**. This is because, in the absence of the formal system T and of the relevant interpretations of T, he is in no position for making the observations concerning the salient features of the pattern/structure in question, observations such as those which have to do with the part/whole distinction, etc. This shows that T, together with the relevant interpretations, does not simply power a deductive engine, but is also a system of representation.

From the considerations above, we can conclude that necessary conditions for pattern recognition in mathematics are the existence of: (1) an observer O; (2) a domain of objects D; and of (3) a system of representation Σ, i.e. $(\mathrm{O}, \mathrm{D}, \Sigma)$.[7]

Thirdly, the mathematical structure which becomes salient when we observe objects **a** and **b** *through* T depends not only on T, but also on **a** and **b** — this is where the realism concerning mathematical structures comes in. In fact,

[6] A discussion of whether mathematics as a whole is conceivable as a science of patterns/structures can be found in: [14], [10], [11], [15], [Resnik, 2001], [12], [13], [1]

[7] Actually, the system of representation Σ is an ordered pair $\Sigma = (\mathrm{T}, I)$, where T is a set containing (as a subset) a recursive set of axioms $\mathcal{A}$ and all the logical consequences of $\mathcal{A}$, and I is an interpretation of T on to D.

given that we can prove in T that there exist exactly seven elements in D and seven m-sets, if, for instance, the number of letters of the Italian alphabet we considered as elements of our matrix were different from seven, the matrix could not be a model of T (the same applies *mutatis mutandis* to the number of points of intersection of three lines in **b**).

Fourthly, we have a criterion of identity for the structure/pattern described by T, criterion of identity represented by model isomorphism, i.e., **a** and **b** instantiate the same structure, because they are isomorphic models of T. This is a very important condition, because it guarantees that the concept of structure is well defined.

Fifthly, we should notice that the definition of structure we offered in §4 — a structure $\mathcal{S}$ is an ordered pair whose first element is a domain of objects D, and second element is a set $\Re$ of relations defined on D — together with the criterion of identity for structures (isomorphism) provide both a rigorous characterization of what falls under the concept of pattern in mathematics, and the possibility of operating a natural generalization of this concept to fields different from mathematics.

With regard to the second point above, notice that both the examples of patterns examined in §3 can be accounted for in terms of structures. In the philosophy of mind and action case, the structure $S_1 = (\mathrm{D}_1, \Re_1)$ is such that D_1 contains beliefs, whereas $\Re_1$ contains relations defined on D_1 such as $\models_{\mathrm{pd}}$ — the plausible deontic consequence relation, where $B_1, \ldots, B_n \models_{\mathrm{pd}} B$ means: someone who believes $B_1, \ldots, B_n$ plausibly ought to believe B. (The turnstile $\models_{\mathrm{pd}}$ is typical of a non-monotonic logic.)

The case of a structuralist account of patterns displayed by crystals of snowflakes (see Fig. 1) is even simpler than that discussed above. The pattern/structure $S_2 = (\mathrm{D}_2, \Re_2)$ of a crystal of a snowflake consists of a domain D_2, the elements of which are the molecules of water contained in the snowflake, and of a set $\Re_2$ whose elements are the physical laws determining how the molecules of water in D_2 are related to one another in the crystal.

But, of course, if the definition of structure we offered in §4 is applicable to both the examples of patterns examined in §3, so does also the identity condition for structure: structure isomorphism.

From here on, as a consequence of what we have been arguing so far, we are going to consider the two words ‘pattern’ and ‘structure’ as synonyms.

5 Patterns’ morphogenesis and cognitive architectures

If we consider the pattern/structure instantiated in object **a** (Fig. 2), we realize that this is a complex entity composed out of simpler entities. The simplest, or atomic entities, are the first 7 letters of the Italian alphabet $\mathrm{A}, \mathrm{B}, \ldots, \mathrm{G}$, and then we have the molecular entities represented by the subsets of three elements of the set $\{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{G}\}$ which appear as the columns of the 3×7 matrix in Fig. 2.

Notice that the atomic entities mentioned above can be thought as patterns/structures of points, as is shown by observing the obvious isomorphism existing among any two of the following different objects:

$$\text{A}, \text{A}, \text{A}, \text{A}, \text{A}.$$

Moreover, molecular expressions such as $\{\mathrm{A},\mathrm{B},\mathrm{D}\},\{\mathrm{B},\mathrm{C},\mathrm{E}\},\ldots,\{\mathrm{G},\mathrm{A},\mathrm{C}\}$ (the columns of the matrix) can also be seen as patterns of patterns. Indeed, the structural rôle of these three-element sets (of patterns) is revealed by the fact that they are obviously isomorphic to the following three-element sets of patterns: $\{a,b,d\},\{b,c,e\},\ldots,\{g,a,c\}$.

All these considerations lead us, in a very natural way, to speak of a morphogenetic process which, starting from atomic patterns $\mathrm{A},\mathrm{B},\ldots,\mathrm{G}$ (patterns of type 0), produces molecular patterns $\{\mathrm{A},\mathrm{B},\mathrm{D}\},\{\mathrm{B},\mathrm{C},\mathrm{E}\},\ldots,\{\mathrm{G},\mathrm{A},\mathrm{C}\}$ (these are patterns of type 1, because their elements are patterns of type 0), molecular patterns which then give origin to the pattern realized in object **a** (Fig. 2). (The latter is a pattern of type 2, because its elements are patterns of type 1).

Now, from the brief account of the patterns' morphogenetic process described above, it should be clear that such a process is capable of generating patterns of arbitrarily large complexity. Therefore, to answer the problem 'How do we come to know patterns?' on the part of a finite cognitive system which is dependent on a limited amount of resources, resources for which he is in competition with other finite cognitive agents, we are going to suggest that such an agent must be endowed with a biologically inspired cognitive architecture (described in §6) which consists of different systems for the representation and the manipulation of information.

To see this, let $\mathcal{A},\mathcal{B},\ldots,\mathcal{G}$ be the shortest neural network algorithms for the recognition of $\mathrm{A},\mathrm{B},\ldots,\mathrm{G}$, within the set of the alphabet letters $\{\mathrm{A},\mathrm{B},\ldots,\mathrm{Z}\}$.[8] The shortest neural network algorithm for the recognition of $\{\mathrm{A},\mathrm{B},\mathrm{D}\}$ will have a length much longer than the sum of the lengths of $\mathcal{A},\mathcal{B}$ and $\mathcal{D}$, because, among other things, lacking a concept of set, our neural network will have to treat $\{\mathrm{A},\mathrm{B},\mathrm{D}\}$ as a plurality of individual patterns and, if we exclude pluralities containing repetitions of letters such as $\{\mathrm{A},\mathrm{A},\mathrm{B}\}$, etc., our algorithm will have to deal with a domain D represented by the power set of $\{\mathrm{A},\mathrm{B},\ldots,\mathrm{G}\}$ which contains 2^7 elements.

Furthermore, the next step, that is, the recognition of **a**, becomes already computationally onerous. For, if $\mathcal{ABD},\mathcal{BCE},\ldots,\mathcal{GAC}$ are the shortest neural network algorithms for the recognition of, respectively, the following patterns: $\{\mathrm{A},\mathrm{B},\mathrm{D}\},\{\mathrm{B},\mathrm{C},\mathrm{E}\},\ldots,\{\mathrm{G},\mathrm{A},\mathrm{C}\}$, the length k of the shortest neural network algorithm for the recognition of **a** will be quite formidable, because, having to recognize **a** out of 7! possible 3×7 matrices the columns of which are the possible permutations of $\{\mathrm{A},\mathrm{B},\mathrm{D}\},\{\mathrm{B},\mathrm{C},\mathrm{E}\},\ldots,\{\mathrm{G},\mathrm{A},\mathrm{C}\}$, k will be much greater than the sum of the lengths of $\mathcal{ABD},\mathcal{BCE},\ldots,\mathcal{GAC}$.

[8] We mention here neural network algorithms, because such algorithms are so far the most basic biologically inspired general procedures for pattern-recognition.

But, of course, in order to individuate the relevant structure realized in $\mathbf{a}$, we should now concatenate to our neural network algorithm for the recognition of $\mathbf{a}$ another neural network algorithm of length k^* for the individuation of the isomorphism inducing function $\psi : D_1 \bigcup D_1^* \to D_2 \bigcup D_2^*$ (see §3). And, since both $D_1 \bigcup D_1^*$ and $D_2 \bigcup D_2^*$ contain 14 elements each, our algorithm will have to recognize ψ out of a set of 14^{14} functions. A tall order indeed!

All these considerations make us suspect that if a finite cognitive agent dependent on a limited amount of resources, resources for which he is in competition with other finite cognitive agents, has in its cognitive architecture systems for the representation of information which use only neural networks, it could not go very far in its pattern recognition activity. And this would not be a consequence of the fact that there are certain patterns for which in principle there is no neural network based algorithm capable of recognizing them, but of the consideration that these algorithms, if they exist, would have to be unfeasibly long, given the computational limitations of our agent.

6 The cognitive architecture. An evolutionary account.

Given what we said in the previous section about the connection existing between patterns' morphogenesis and the cognitive architecture of a finite cognitive agent who is dependent on a limited amount of resources, resources for which he is in competition with other finite cognitive agents, in what follows in this section we are going to illustrate a cognitive architecture (see figure 7) consisting of three levels of information-representation: a *subconceptual level*, in which data coming from the environment (sensory input) are processed by means of a neural network based system; a *conceptual level*, where data are represented and conceptualized independently of language; and, finally, a *symbolic level* which makes it possible to manage the information through symbolic/linguistic representations and computations.
Notice that all three levels for the representation and processing of information mentioned above are present in humans, and that the first two levels may be found in most higher animals, etc.

We have already come across the sub-conceptual level of representation (the Sub-conceptual Tier) in §5 when we discussed the possibility of recognizing type 0 patterns (atomic patterns) by means of algorithms based on neural networks. What we need to do now is providing a brief description of the conceptual and symbolic levels of representation of the cognitive architecture sketched in Figure 7.

The conceptual level of the cognitive architecture of our agent consists of the so-called 'Gärdenfors conceptual spaces'. According to Gärdenfors, conceptual spaces are metric spaces which represent information exploiting geometrical structures rather than symbols or connections between neurons. This geometrical representation is based on the existence/construction of a space endowed with a number of what Gärdenfors calls 'quality dimensions' whose main func-

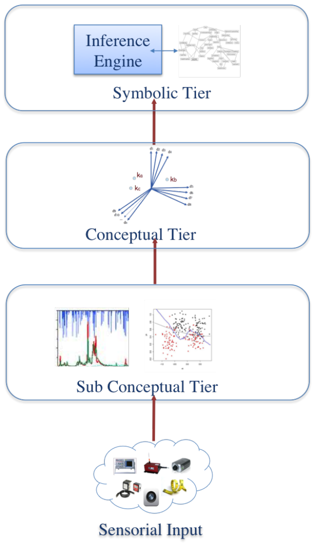


Fig. 7. A sketch of the cognitive architecture

tion is to represent different qualities of objects such as brightness, temperature, height, width, depth.

Moreover, for Gärdenfors, judgments of similarity play a crucial role in cognitive processes and, according to him, the smaller is the distance between the representations of two given objects (in a conceptual space) the more similar to each other the objects represented are.

For Gärdenfors, objects can be represented as points in a conceptual space, points which we are going to call 'knoxels',[9] and concepts as regions (in a conceptual space). These regions may have various shapes, although to some concepts — those which refer to natural kinds or natural properties — correspond regions which are characterized by convexity.[10] According to Gärdenfors, this latter type of region is strictly related to the notion of prototype, i.e., to those entities that may be regarded as the archetypal representatives of a given category of objects (the centroids of the convex regions).

Finally, the symbolic level (the Symbolic Tier) of the cognitive architecture consists, instead, of language-based systems of information representation and computation.

[9] The term 'knoxel' originates from [7] by the analogy with "pixel". A knoxel k is a point in Conceptual Space and it represents the epistemologically primitive element at the considered level of analysis.

[10] A set S is *convex* if and only if whenever $a, b \in S$ and c is between a and b then $c \in S$.

To see the three levels of the cognitive architecture at work, and assess their relative merits, consider the following problem: to recognize the pattern exemplified by object **A**.

If we assume that the algorithms that follow can all be expressed in a given language $\mathfrak{L}$, then the advantage of using algorithms based on tools characteristic of the sub-conceptual level (neural networks) to solve the problem above is that ... an algorithm is better than nothing! On the other hand, the obvious disadvantage is that neural network based algorithms can be relatively long.

Imagine now a 2-d Gärdenfors conceptual space, CSA, related to the letters of the alphabet. This is a CSA tessellated by means of prototypes of such letters using the well-known Voronoi's procedure. The pattern-recognition algorithm relating to **A** is quite simple: determine to which of the finitely many points belonging to CSA which represent the prototypes of the letters of the alphabet the point representing **A** in CSA is nearest.

Although the use of conceptual spaces is able to produce pattern recognition algorithms much more compressed than neural network based algorithms for the recognition of the same patterns, it has a serious defect: conceptual spaces are 'in the head' in the sense that they ultimately have perceptual space as a 'vehicle'. And, therefore, a finite cognitive agent dependent on a limited amount of resources, resources for which he is in competition with other finite cognitive agents, will have difficulties in exploiting the full potential of conceptual spaces.

However, the following 'symbolic algorithm': (1) list the letters of the alphabet; (2) check whether **A** is an instance of the first letter; if yes (3) stop; if no (4) check whether **A** is an instance of the second letter; ... (n) stop; is certainly shorter (and safer) than the CSA-algorithm mentioned above.

Other advantages of stepping up to the symbolic level are that:

1. language enables many minds to be connected in what we might call a 'world wide web' overcoming in this way the computational limitations of every single mind;
2. language is not 'in the head', in the sense that language allows:
   2.1 the storing of portable information in the form of articles, books, inscriptions, etc. information which, among other things, no longer needs to occupy storing space in individuals' minds;
   2.2 objectivity in the treatment of information, because in language information is conveyed by assertions for which there exist public criteria of correctness which we all learn when we learn the language;
3. language extends our representational and computational capabilities. To see this consider the natural number $10^{10^{10}}$. There is no chance that we are able to represent within our visual perceptual space such a multiplicity and distinguish it, for example, from a multiplicity of $10^{10^{10}} \pm 7$ elements. And yet, within number theory, not only there are many things we can prove about such multiplicities, but we can also use their cardinal numbers in our ordinary arithmetical computations. These considerations apply even more so to transfinite cardinal numbers such as $\aleph_0, \aleph_1, \ldots$ and their arithmetic.

Many more are the things that could be said in favour of the great importance of language for pattern-recognition. However, those which have already been mentioned in this section are sufficient to show the crucial rôle that the symbolic level has in the cognitive architecture of a finite cognitive agent who is dependent on a limited amount of resources, resources for which he is in competition with other cognitive agents.

But, before ending this section and the paper, we need to spend a few words to justify the cognitive architecture here presented. To this end, let us consider, as we have repeatedly said, that our cognitive agent is finite, dependent on a limited amount of resources, and engaged in a constant struggle for life with nature and other cognitive agents, and that:

> Owing to this struggle for life, any variation, however slight and from whatever cause proceeding, if it be in any degree profitable to an individual of any species, in its infinitely complex relations to other organic beings and to external nature, will tend to the preservation of that individual, and will generally be inherited by its offspring. ([5], Chapter III, p. 40.)

From this we have that, as a consequence of natural selection,[11] our cognitive agent not only develops a hard-wired pattern-recognition machinery in his visual system — edge detectors, luminance detectors, and the like (see on this the quotation from [6] on p. 7 of this article) — but also a multi-level cognitive architecture for the representation and manipulation of information.

At this point it is clear that questions like 'Why does the cognitive architecture have three different levels?', 'How do conceptual spaces come about in the cognitive architecture?', etc. can only be give an 'evolutionary answer', that is, the cognitve architecture we have illustrated above is the consequence of variations which come about in the system of representation and manipulation of information of human beings. These are variations which have been preserved as a consequence of their being greatly profitable for the crucially important pattern-recognition activity of humans.

7 Conclusions

In this paper we intended to give a contribution to the foundations of pattern-recognition theory; and, to do so, we decided to address two central questions: (a) 'What is a pattern?' and (b) 'How do we come to know patterns?'

Dealing with question (a), we produced a definition of mathematical pattern which we then generalized to fields different from mathematics (philosophy of mind and action, physics). But, when it came to answering question (b), we thought of presenting a cognitive architecture for a finite cognitive agent who is dependent on a limited amount of resources. This is a cognitive architecture

[11] 'This preservation of favourable variations and the rejection of injurious variations, I call Natural Selection.' ([5], Chapter IV, p. 51).

which is, in principle, able to cope with some of the basic demands posed by the process of pattern-recognition; and has developed as a consequence of Darwinian natural selection.

References

1. Bombieri, E.: 2013, 'The shifting aspects of truth in mathematics', *Euresis*, **vol. 5**, pp. 249–272.
2. Chaitin, G.: 1975, 'Randomness and Mathematical Proof', *Scientific American*, Vol. CCXXXII, pp. 47-52.
3. Chella, A., M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. Artif. Intell., 89:73111, 1997.
4. Dales, H.G. & Oliveri, G. (eds.): 1998, *Truth in Mathematics*, Oxford University Press, Oxford.
5. Darwin, C.: 1859, *On the Origin of Species*, edited by M. T. Ghiselin, Dover Publications, 2006, Mineola, New York.
6. Dennett, D.: 1991, 'Real Patterns', *The Journal of Philosophy*, Vol. 88, No. 1, pp. 27-51.
7. Gaglio, S., P. P. Puliafito, M. Paolucci, and P. P. Perotto. 1988. Some problems on uncertain knowledge acquisition for rule based systems. Decis. Support Syst. 4, 3 (September 1988), 307-312. DOI=10.1016/0167-9236(88)90018-8 http://dx.doi.org/10.1016/0167-9236(88)90018-8
8. Gärdenfors, P.: 2004, *Conceptual Spaces: The Geometry of Thought*, MIT Press, Cambridge, Massachusetts.
9. Gärdenfors, P.: 2004. 'Conceptual spaces as a framework for knowledge representation'. *Mind and Matter* 2 (2):9-27.
10. Oliveri, G.: 1997, 'Mathematics. A Science of Patterns?', *Synthese*, **vol. 112**, issue 3, pp. 379–402.
11. Oliveri, G.: 1998, 'True to the Pattern', in: [4], pp. 253–269.
12. Oliveri, G.: 2007, *A Realist Philosophy of Mathematics*, College Publications, London.
13. Oliveri, G.: 2012, 'Object, Structure, and Form', *Logique & Analyse*, **vol. 219**, pp. 401-442.
14. Resnik, M.D.: 1981, 'Mathematics as a Science of Patterns: Ontology and Reference', *Noûs* **XV**, pp. 529-550.

[Resnik, 2001] Resnik, M.D.: 2001, *Mathematics as a Science of Patterns*, Clarendon Press, Oxford.

15. Shapiro, S.: 2000, *Philosophy of Mathematics. Structure and Ontology*, Oxford University Press, Oxford.
16. Tuller, A.: 1967, *A Modern Introduction to Geometries*, D. Van Nostrand Company, Inc., Princeton, New Jersey.
17. Wittgenstein, L.: 1983, *Philosophical Investigations*, Second Edition, transl. by G. E. M. Anscombe, B. Blackwell, Oxford.

World Modeling for Tabletop Object Construction

Arda Inceoglu, Melodi Deniz Ozturk, Mustafa Ersen, and Sanem Sariel
{inceoglua,ozturkm,ersenm,sariel}@itu.edu.tr

Artificial Intelligence and Robotics Laboratory
Istanbul Technical University, Istanbul, Turkey

Abstract. In tabletop construction scenarios, robots work with vertically or horizontally stacked object structures. In order to form such structures, they need to recognize and correctly model closely placed objects in such structures. Depending on the robot's point of view and the objects' positions, it is likely that objects closely located or in contact partially occlude each other, and as a result it is not always possible to model object stacks by relying only on object recognition. However, if the objects are added to the construction consecutively, it becomes possible to sequentially build the model of object stacks. In this work, we propose a scene interpretation system to build and maintain a consistent world model for tabletop construction scenarios. To overcome the challenge of modeling object stacks, we extend our previous scene interpretation system with a semi-closed world assumption and by preserving the models of objects in the formed structures even when they are out of sight. Our extension includes the use of spatial object relations, as well as depth-based segmentation results to model not only single objects, but object combinations. In our system, the LINE-MOD algorithm and an enhanced version with HS histograms are used for recognizing objects along with depth-based segmentation for detecting novel objects. We run numerous construction scenarios using building blocks and show that our system can be successfully used for modeling constructed objects.

Keywords. Scene interpretation, Tabletop object construction, Object manipulation, World modeling for tabletop manipulation

Introduction

In order to achieve given goals, robots often need to interact with various objects. For successful interaction, before anything else, they need to collect correct information about the objects in their environment. The required data includes the accurate properties of objects, such as their size, shape and color, their locations in the world and necessary inter-object relations. For this purpose, robots use their sensors to gather observations from the world, which sometimes do not overlap, are not complete, and sometimes even contradict with each other. Our previous work presents a scene interpretation system to cope with these challenges for a ground robot [1], [2]. In this work, we focus on tabletop object construction scenarios and extend our previous work for modeling stacked objects during task execution. This is mainly important for continually monitoring execution against anomalies (e.g., effects of external interventions) or unexpected

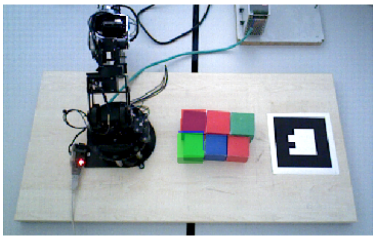


Fig. 1. A rectangular structure is to be built from six cubical blocks by a robot arm. The blocks are from different colors and sizes. When all these blocks are attached to each other, it may not be possible to recognize all of them at once.

outcomes (e.g., inherent failures). New objects should be correctly localized with their properties interpreted and information on these objects should be maintained against any changes (e.g., after disappearing from the scene or displacement in any way). Besides, when symbol grounding is needed for further cognitive skills such as reasoning and learning, correct identification of objects is a prerequisite.

World modeling is especially challenging when objects are in direct contact with each other either horizontally or vertically (i.e., when they are on top of each other). In these scenarios, it is likely that they partially occlude one another from the robot's point of view or the vision algorithms may fail in recognizing all objects. For example, consider the scenario where the robot is tasked to build a rectangular prism structure from a set of cubical blocks in different colors and sizes as in Figure 1. In this figure, the LINE-MOD algorithm [3] is used to recognize textureless objects in 3D by considering their surface normals and matching existing measurements with their previously registered templates. However, since the objects are attached to each other, their boundaries and some of their surfaces can not be distinguished well which results in errors in recognition of some of the blocks (e.g., only two blocks on the leftmost column, indicated with the corresponding markers in the figure, are recognized for this scenario). This problem can be alleviated by reducing the similarity threshold used for matching templates in the algorithm. However, this may result in false positives. Humans, on the other hand, intrinsically use their background and default knowledge when faced with similar problems, incorparating the recent history of events that have led to the current situation. In this study, we are inspired by this cognitive skill and propose a system to reach logical conclusions, similarly to humans, about the robots environment.

Given the requirements in modeling objects in construction scenarios, we propose new advancements over our previous scene interpretation system. The contributions of this work are three fold. First, the scene interpretation system is made capable of using observations taken during execution and building the model of a structure incrementally using both temporal and spatial relations extracted during runtime and prior semantic rules for handling occlusions. Second, a truth maintenance mechanism is applied to store the models of occluded objects even if they can not be recognized but to remove their models when they are believed to disappear from the scene. Third, for the identification of objects a semi closed-world assumption is applied for symbol grounding.

The rest of the paper is organized as follows. First, we mention studies related to world modeling. Then, we describe our scene interpretation system for consistent world modeling in tabletop block construction scenarios. We then give empirical results of our system followed by the conclusions.

Related Work

Several recent studies address the issue of maintaining a world model from the robot's visual observations. Nyga, Balint-Benczedi and Beetz (2014) proposed an ensembles of experts approach based on Markov Logic Networks [4] for fusing different aspects of information coming from different object recognition methods (e.g., LINE-MOD [3], Google Goggles etc.) enabling robots to answer logical queries about different aspects of recognized objects [5]. WIRE [6] is a system based on multiple hypothesis anchoring for robots to maintain semantically rich world models in unstructured and dynamically changing environments. It relies on multiple model tracking for incorporating prior knowledge and multiple hypothesis tracking-based data association for consistently updating the world model using new observations. Another similar study addresses the data association problem from a different perspective by using clustering-based approaches instead of multiple hypothesis tracking [7]. In our previous work, we presented a temporal scene interpretation system for maintaining a consistent world model relying on noisy perception outcomes [1]. Our system uses segmentation outcomes as well as object recognition outcomes to be able to detect objects without previously generated recognition models as unknown object candidates, updates the world model by evaluating these perceptual outcomes temporally, and takes the robot's field of view into account during these updates. In this paper, we enhance our scene interpretation system in the following directions. First, we replace the 2D model of the robot's field of view we used for our ground robot with a 3D model which is necessary for tabletop object manipulation scenarios. Second, we incorporate a semi-closed world assumption for keeping track of previously encountered objects. Finally, we present enhancements for the block construction domain.

Perception Sources

The first step in object manipulation by autonomous robots is maintaining a consistent and up-to-date world model about their environment. For this task, the robot has to collect visual recognition and detection data to filter out and to reach conclusions. Our perception system uses LINE-MOD [3], LINE-MOD&HS [8] and 3D segmentation [9] algorithms as means for processing 3D sensory data obtained from an on-board ASUS Xtion Pro RGB-D camera. LINE-MOD is an object recognition algorithm that uses surface normals of the objects, calculated from the Point Cloud [10] data regarding the object, to extract object templates. The algorithm then uses these templates with the sliding windows approach to detect the modelled objects in new scenes. The LINE-MOD&HS algorithm, in turn, augments LINE-MOD to use the HSV histograms of the objects in order to integrate the use of color information of the objects in recognition.

Additionally, 3D segmentation is used for detecting objects that are either not previously modelled, or otherwise cannot be recognized in the current scene.

The perception sources are implemented as separate processes, where their recognition/detection results are asynchronous. The Scene Interpreter system combines these results to create an accurate representation of the world [1].

Scene Interpretation for Tabletop Manipulation

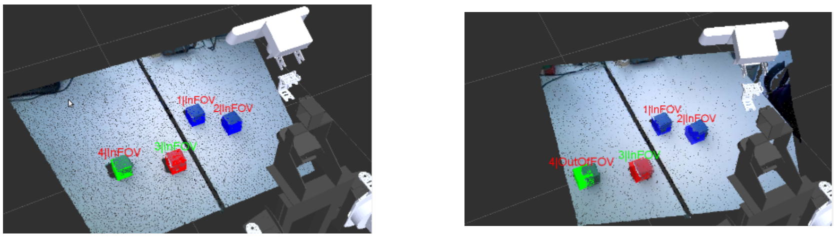
Object recognition is not reliable alone for robotic manipulation tasks, since failures in recognition or detection occur due to noisy sensor measurements, illumination changes, dynamic environments or other agents and sensors. In order to automatically build a consistent and up-to-date model about the environment, visual recognition and detection outcomes should be filtered out and logical conclusions should be reached in the face of contradictory outputs. Previous work by the same research team includes a Scene Interpretation system for ground robots working with objects clearly separated in the horizontal plane [1], which forms the foundation of the proposed system. Necessary deductions about a robot's environment include a unique id for each object in the environment, their type, color, size, shape and location properties, as well as the confidence of the system about these object's existence in the environment.

The confidence is represented with a value varying between 0 and 100, proportional to the degree of belief on the corresponding object's existence. Confidence values are updated with every new perception outcome. An object's confidence value increases as more consistent recognition or detection results arrive regarding the object.

The observed facts are kept in the Knowledge base (KB) of the robot, which can be defined as a collection of reached conclusions about objects, their properties, and inter-object relations. The robot's KB is initialized as empty. During run time, recognized objects are inserted into the KB and their corresponding confidence values, as well as properties, are updated with each newly received recognition message. If an object in the KB does not receive any corresponding recognition message for a period of time, even though this object is in the robot's field of view and should be recognized, the confidence value regarding the object is gradually decreased. If this value reaches zero, it is believed that the object is no longer in the robot's environment, and thus it is removed from the KB.

Most humanoid robots have the capability of moving their heads around, making it possible for them to observe more about their environment. As a result, their visual field of view (FOV) is bounded by the limitations of their cameras. A robot can receive reliable information about objects only within its FOV and the field of view constraints should be taken into account when updating object properties. Extending the 2D definition in the base system, 3D boundaries are empirically determined for an RGB-D camera where the objects within are expected to be recognized reliably. An example scenario regarding FOV calculations can be seen in Figure 2.

Objects in the environment are considered depending on whether they are inside the camera's FOV or not. Objects outside the FOV are not expected to be recognized, and any data regarding them in the KB are kept static until they re-enter the FOV of the robot, and new perception data are available.



(a) (b)

Fig. 2. A scenario demonstrating FOV calculations. In (a) all objects are in FOV. In (b) the robot's head is slightly rotated to right. This time the green block becomes out of FOV, yet it is kept in the KB. (Note that FOV area is determined tighter than the actual physical limits of the camera for more reliable object recognition, and it can be adjusted easily.)

Spatial Relations

After recognition and localization of the objects in the scene, their spatial relations are determined as in [1]. These relations are represented as unary or binary predicates such as $onTable(obj1)$, $near(obj1, obj2)$ and $on(obj2, obj1)$. Consider a scenario where three blocks are stacked on top each other, assigned ids 1 to 3 from bottom to top. There are two on relations expected such that $on(3, 2)$ and $on(2, 1)$. Objects in the bottom are considered as out of field of view and thus they are not expected to be detected. We make use of this for modeling objects in block construction as visualized in Figure 3.

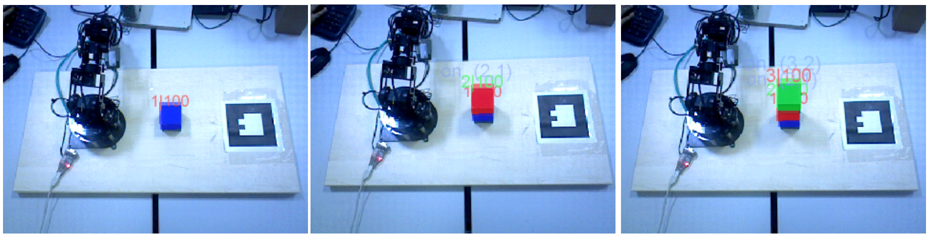


Fig. 3. The phases of a block stacking scenario in the real world (from left to right). The recognized objects along with their ids and confidence values, and the relations among them (e.g.,$on(obj_2, obj_1)$) are marked on the original RGB image in Rviz.

Symbolic Models of Tabletop Objects

The first part of the study focuses on symbol grounding problem for objects ids. Ambiguities in determination of ids can arise in dynamic scenes. Objects may be displaced, removed from and put back into the scene, or the robot could be mobile and have localization problems. As a result, an object might be registered with different ids over time, which prevents creating and executing plans including object manipulation successfully.

After successfully detecting objects, the world is assumed to be closed (closed world assumption) for identity resolution tasks. Closed world assumption [11] can be defined as having complete knowledge about the world, that is, the numbers and the attributes of all objects are known apriori. However, robots often have partial information about the world. Even though object attributes are known, objects' locations may be dynamic or unknown which requires obtaining extra information from the environment [12]. Whereas, in an open world assumption, no prior knowledge about the world is given, and every object entering to the scene is assumed to be encountered for the first time. In contrast, we define a semi-closed world assumption in which the robot builds its KB itself at runtime and does not use any prior knowledge about the scene contents. At each object detection, the attributes of the object are compared with that of previously registered objects in the KB. If an object is believed to have been encountered before, its previous id is used, otherwise a new id is generated. The corresponding algorithm is given in Algorithm 1.

Data: Detected object attributes
Result: Object Id
foreach $object$ **in** KB **do**
    if *attributes match* **and** *object is not in the scene* **then**
        return $object.id$;
    else
        $newId \longleftarrow$ generate new id ;
        return $newId$
    end
end

Algorithm 1: Algorithm for Semi-Closed World Assumption

Symbolic Relations among Tabletop Objects

The second focus of the study is to correctly model closely located objects. Object recognition in cluttered scenes is still a challenging problem. Object detection success is low in such scenes due to their placements. An example scenario with six cubical blocks is given in Figure 4. The system can not distinguish between objects and only some of the objects can be added to KB. This is the natural result of assumptions of vision algorithms. LINE-MOD extracts surface normals on visible surfaces and color gradients around borders. Furthermore, the 3D segmentation algorithm assumes objects are clearly separable on a supporting plane.

The first solution attempt to this problem was decreasing the similarity threshold of the LINE-MOD algorithm between the object templates and real time detections (See Figure 5). The threshold is set to 95% by default, and it is decreased gradually. As a result, the system was able to detect the objects and register them into the KB. The main drawback of the approach is, as the threshold is lowered, the number of false positives, i.e. the number of misdetections increase. As the threshold reaches 80% and below it becomes harder to maintain the number of objects in the KB.

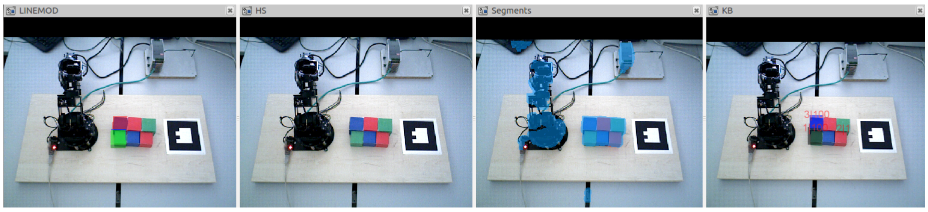
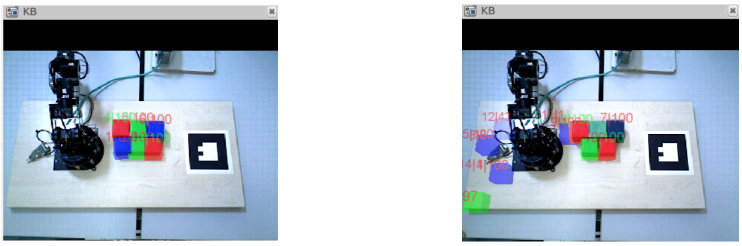


Fig. 4. The first three frames represent the outputs of algorithms LINEMOD, LINEMOD&HS and Segmentation, respectively. Rightmost frame represents the state of the KB, where recognized objects are marked with their ids and a confidence values. The six cubes are placed into the scene initially, only 3 of them are recognized and added to KB.



(a) (b)

Fig. 5. Scenarios with different similarity thresholds for the object recognition algorithm. Thresholds are set to 90% in (a) and 85% in (b). Comparing with Figure 1, in (a) objects are recognized and added to the KB. For the case of (b) false positives are introduced by decreasing the threshold.

For a similar scenario where the threshold is set to 95%, even though objects are placed into the scene one by one but very closely, some of the previously recognized objects are removed from the KB due to lack of recognition messages after some point. The second proposed approach utilizes the 3D segmentation algorithm. We can rely on detections of the 3D segmentation algorithm in terms of the existence of an object in the scene. If the objects are placed into the scene one by one and clear enough to be recognized, after successfully being added to the KB, the objects can be marked as detected, if their centroid lies in one of the last segmented point clouds produced by the 3D segmentation algorithm. Thus, the objects are exempted from being removed from the KB. The proposed algorithm is given in Algorithm 2.

The segmentation algorithm is used to maintain object stacks of the same level. In order to increase the level -the height- of the structure, spatial relations among objects, namely on relations, are employed. When objects are stacked on top of each other, corresponding on relations are detected between object pairs. In a pair, the bottommost object is partially occluded, so it is not expected to be detected, which avoids the update operation on the object and thus the deletion from the KB. In addition, if the topmost object of a pair is removed from the scene, the corresponding object model and the on relation are also removed from the KB, and the bottommost object is expected to be detected again.

```
KB ⟵ initialize empty knowledge base;
upon receive Objects:;
/* Objects : Recognized objects via LINEMOD and LINEMOD&HS */
foreach object in Objects do
    if object in KB then
        update object;
    else
        KB ⟵ add object ;
    end
end
upon receive Segments:;
/* Segments : Segmented point cloud clusters */
foreach object in Objects do
    if objectcentroid in Segments then
        object ⟵ mark object as detected
    end
end
```

Algorithm 2: Maintaining Closely Located Objects

Experiments

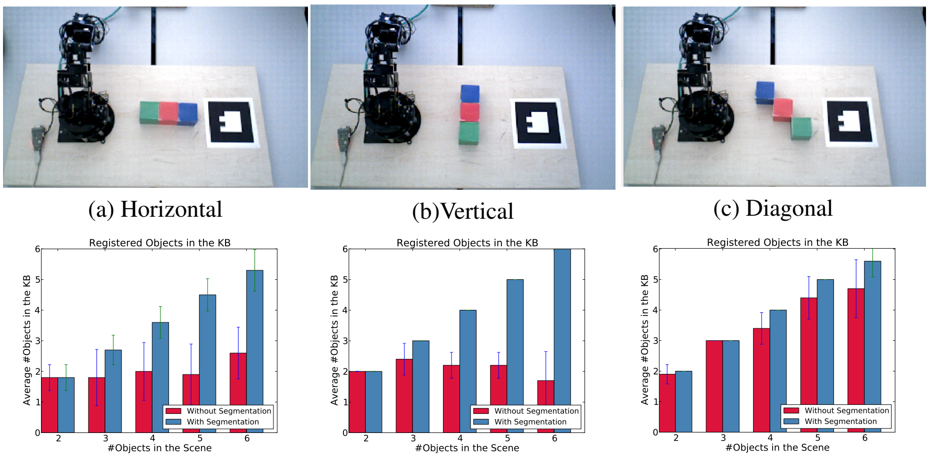
This section describes the experimental setup and presents the obtained results. First, we present object recognition and registration to KB during run time. Then, id tracking capabilities of our system under semi-closed world assumption is demonstrated.

Object Registration to the Knowledge Base

For the first part, block construction scenario is considered. Red, green and blue colored blocks are placed into the scene sequentially to form horizontal, vertical and diagonal structures on the same plane. For comparison purposes, experiments are repeated with and without employing the proposed segmentation based approach. Each time, after a block is placed, the number of objects registered to the KB is recorded. Each case is repeated 10 times, and the mean is calculated. Figure 6 shows the comparison for horizontal, vertical and diagonal structures. Note that, since the results are recorded in a sequential manner, errors in the previous steps accumulated to oncoming steps.

The following conclusions can be drawn from the analysis given in Figure 6. Employing segmentation based approach fairly increases the number of objects registered to the KB. The best performance is obtained from vertical placement scenario due to the fact that the last placed object can be correctly isolated from its surroundings and thus, it is easier for the vision algorithm to recognize. However, in the diagonal placement scenario using segmentation does not provide much improvement since objects are in less contact with each other. Horizontal scenario is the most complicated one in terms of distinguishing between objects, since objects have more contact with each other. Improvements become clear when the number of objects in the structure is increased.

In another experiment, blocks are stacked on top of each other one by one to measure *on* relation detection success when new layers are introduced. This time, after each



(a) Horizontal (b)Vertical (c) Diagonal

(d) Horizontal (e)Vertical (f) Diagonal

Fig. 6. Comparison of the number of registered objects into the KB.

stacking operation, the number of *on* relations is recorded. It is expected to detect one *on* relation with a stack of two objects, two *on* relations with a stack of 3 objects and so on. Success rates of detecting *on* relations are 100%, 100% and 93.3% for the number of layers 2,3 and 4 respectively. Due to object recognition failures, success rate is decreased in the 4th layer and above.

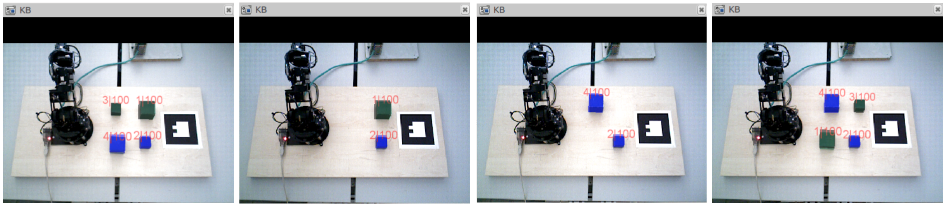
Semi-Closed World Assumption

The goal of this experiment is to illustrate id tracking capabilities of the system. The object set contains red, green and blue colored, medium sized cylinders, and small and large sized blocks. In the model; type, size and color attributes are taken into account. An example scenario is visualized in Figure 7.

The KB is initialized by putting all target objects into the scene, and each object is assigned a unique id. Then, the objects are removed from the scene. Each object is put back and id assignments are observed. A confusion matrix based on id assignments is given in Figure 8. Whenever an object could not be matched with the previously encountered objects registered to KB, a new id is generated for the object. The reason of mismatches are originated from errors in recognizing objects due to illumination conditions. It is observed that if an object is failed to match with an object in the initial object set, and thus attached a new id, the consecutive recognitions are also matched to this id. Whereas, some objects could not be recognized at all, which are denoted as not detected in Figure 8.

Conclusion

We have presented enhancements for our scene interpretation system in order for it to be used in tabletop manipulation and construction scenarios for cognitive robots. First,



(a) (b) (c) (d)

Fig. 7. A scenario demonstrating the id tracking performance of the system.(a) contains small and large sized blue and green cubes. Object are added to the KB and each is attached a unique id. In (b) the small green cube and the large blue cube are removed. In (c) the large green cube is removed and then the large blue cube is placed again. In (d) all objects are put back into the scene. By using the size and color attributes, the system is able to remember the objects and attach their previous ids.

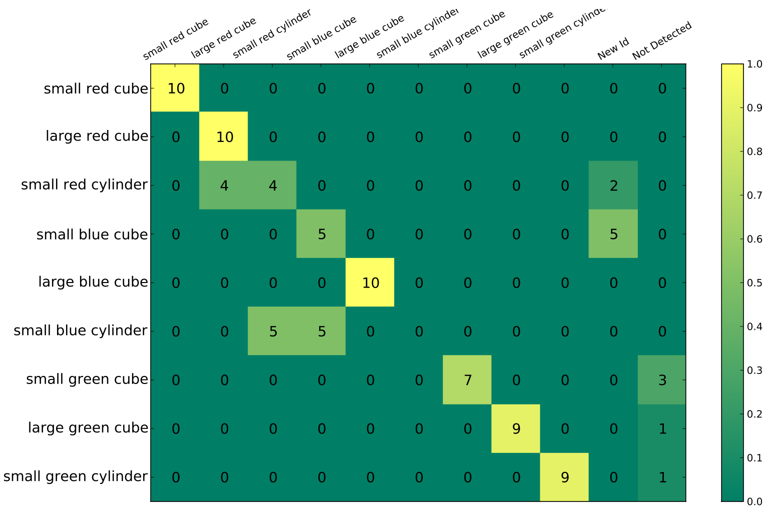


Fig. 8. Confusion matrix for semi-closed world assumption. After each object is assigned a unique id, objects are removed and put back. Matrix shows original versus newly assigned ids. A new id is assigned if object is recognized yet could not be matched with previous objects. Not detected denotes, object could not be recognized at all.

the 2D model of a ground robot's field of view was extended to 3D for a humanoid robot with a moveable head. Then, we introduced a hybrid model of open and closed world assumptions for keeping track of object ids in case of dislocations and disappearances & reappearances. This hybrid model is able to keep track of lost objects, while still allowing new objects to enter the scene. Deductions about object ids are made based on physical attributes of the objects and without using any kind of prior knowledge. Finally, for block construction scenarios, we proposed utilizing 3D segmentation on top of object recognition to maintain objects in the KB when they are in direct contact with each other and cannot be recognized. Furthermore, we employed spatial relations to maintain objects that have other objects on top of them and thus to model higher level structures. Future work includes improving the system to keep track of more complicated scenarios that include unknown objects, and modifying the system to operate on a probabilistic framework.

Acknowledgments

This research is partly funded by a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Grant No. 111E-286.

References

1. M. D. Ozturk, M. Ersen, M. Kapotoglu, C. Koc, S. Sariel-Talay, and H. Yalcin, "Scene interpretation for self-aware cognitive robots," in *Proceedings of the 2014 AAAI Spring Symposium: Qualitative Representations for Robots*, pp. 89–96, AAAI Press, 2014.
2. M. Ersen, M. D. Ozturk, M. Biberci, S. Sariel, and H. Yalcin, "Scene interpretation for lifelong robot learning," in *The 9th International Workshop on Cognitive Robotics (CogRob 2014) held in conjunction with ECAI-2014*, (Prague, Czech Republic), 2014.
3. S. Hinterstoisser, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012.
4. M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
5. D. Nyga, F. Balint-Benczedi, and M. Beetz, "PR2 looking at things - Ensemble learning for unstructured information processing with markov logic networks," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3916–3923, IEEE Press, 2014.
6. J. Elfring, S. van den Dries, M. J. G. van de Molengraft, and M. Steinbuch, "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, 2013.
7. L. L. S. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Data association for semantic world modeling from partial views," *International Journal of Robotics Research*, Accepted for publication.
8. M. Ersen, S. Sariel-Talay, and H. Yalcin, "Extracting spatial relations among objects for failure detection," in *Proceedings of the KI 2013 Workshop on Visual and Spatial Cognition*, pp. 13–20, 2013.
9. A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration (SPME)*, 2013.
10. A. Aldoma, Z. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.
11. R. Reiter, "Readings in nonmonotonic reasoning," ch. On Closed World Data Bases, pp. 300–310, Morgan Kaufmann Publishers Inc., 1987.
12. O. Etzioni, K. Golden, and D. S. Weld, "Sound and efficient closed-world reasoning for planning," *Artificial Intelligence*, vol. 89, no. 12, pp. 113 – 148, 1997.

A Network-based Communication Platform for a Cognitive Computer

Mostafa W. Numan, Jesse Frost, Braden J. Phillips, and Michael Liebelt

Centre for High Performance Integrated Technologies and Systems (CHiPTec)
School of Electrical and Electronic Engineering
The University of Adelaide
Adelaide, Australia
{mostafa.numan,jesse.frost,braden.phillips,michael.liebelt}@adelaide.edu.au

Abstract. *Street* is a reconfigurable parallel computer architecture. It executes a production language directly in hardware with the aim of realising advanced cognitive agents in a more energy efficient manner than conventional computers. Street requires frequent communication between many processing elements and to make this communication more energy efficient, a network-based communication platform, *StreetNet*, is proposed in this paper. It maps the processing elements onto a 2D mesh architecture optimized according to the data dependencies between them. A deadlock-free deterministic routing function is considered for this platform along with the concept of sleep period, analogous to human sleeping, to reorganize the placements of processing elements based on runtime traffic statistics. These mechanisms serve to reduce total network traffic and hence minimise energy consumption.

Keywords: Cognitive computer, computer architecture, networks-on-chip, mapping

1 Introduction

Street is a reconfigurable, flat, parallel architecture designed for symbolic cognitive workloads [6]. The goal of Street is to find a new computer architecture that can take advantage of the huge number of transistors in modern integrated circuits to achieve advanced cognitive computation in real time, but with much lower power consumption than current computers. It is designed to use in real time embedded implementations of artificial general intelligence, exemplified by the plethora of potential autonomous robotics applications. The new machine is very different from conventional computers, consisting of many simple processing elements executing and communicating in parallel. A bus-based interconnect performs well in production systems with a small number of processing elements, or when groups of dependent productions are mapped to the same processor [1], however it does not scale well for more frequently interacting processing elements. For chips with a large number of processing elements, network based communication provides better scalability, and is seen as the most efficient solution [13]. In this paper, a network-based communication platform, *StreetNet*, is proposed for efficient communication among the processing elements of Street.

2 Street

Street executes a parallel production language directly in hardware. This language, which we call Street Language, is inspired by Forgy's OPS5 [5] and the languages used in the Soar [10] and ACT-R [3] cognitive architectures. However Street Language is different from all of these. Street is asynchronous, with no global match-select-act cycle as found in traditional production systems. This asynchronous model provides the best opportunity to parallelise traditional production systems in application level [2].

2.1 Street Language

An intelligent system is implemented using a set of production rules written in Street Language [6]. Each production rule is an *if-then* statement: *if* a specified pattern exists in working memory, *then* the rule makes some changes to working memory. Working memory is a set of tuples called working memory elements (WMEs). Each WME has one or more elements called attributes. For instance, the WME (ID17 source ID2) has 3 attributes: ID17, source, and ID2. Here is a simple example of working memory of just 3 WMEs:

{(ID17 name Torrens), (ID17 source ID2), (isCounted ID5)}

Each production rule consists of a left hand side (LHS) of one or more condition elements (CEs), and a right hand side (RHS) of one or more actions. In the example in Fig. 1, (<p> type dog) is a CE and (<p> isOld) is an action. A complex cognitive agent would consist of thousands of production rules operating on symbolic and numeric data in working memory.

```
st {oldDogs
    (<p> type dog)      // condition elements
    (<p> age (<a> > 7))
-->
    (<p> isOld)         // actions
}
```

Fig. 1. A Street Language production rule

A subset of working memory that satisfies all of the CEs in a production rule with consistent variable assignments is called an instantiation of the rule. So instantiations of the rule above will be pairs of WMEs in working memory such as: {(pet1 type dog), (pet1 age 8)}. Note that the first attributes must be

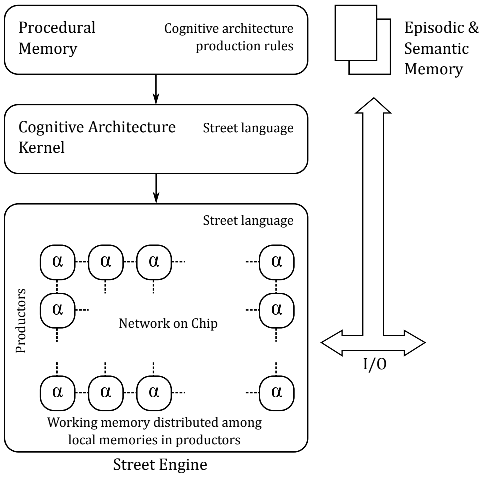


Fig. 2. Hardware/software stack on *Street* (adopted from [6])

the same as they were specified by the same variable `<p>`. The WMEs must join on any shared variables. The actions of a production rule are performed for each new instantiation of the rule. An action such as `(<p> isOld)` adds a WME to working memory. Actions can also remove WMEs from working memory. This change in working memory may cause other production rules to instantiate, and all new instantiations are executed.

2.2 Street Architecture

The Street architecture consists of a large number of identical and simple microcoded processing elements (PEs) with a single production rule assigned to each. A PE contains a controller, a block of content-addressable memories (CAMs) and an Arithmetic Logic Unit (ALU). The PEs communicate using tokens to notify each other of changes to working memory. The local memory of a PE stores just the subset of working memory that may lead to instantiations of its rule. Each PE matches the associated production rule against its own local memory with an algorithm similar to TREAT [12]. For each incoming token, a PE updates the contents of its local memory, finds new instantiations (*match*), and outputs tokens corresponding to the rule's actions (*act*). The controller coordinates the PE's match-act cycle. Fig. 2 shows Street architecture executing an agent using a symbolic cognitive architecture.

3 StreetNet: The Communication Platform

When a PE produces tokens these are transmitted to other PEs and may cause new rule instantiations and yet more tokens. There may be a large amount of token traffic between PEs or small clusters of PEs so efficient data interconnect is

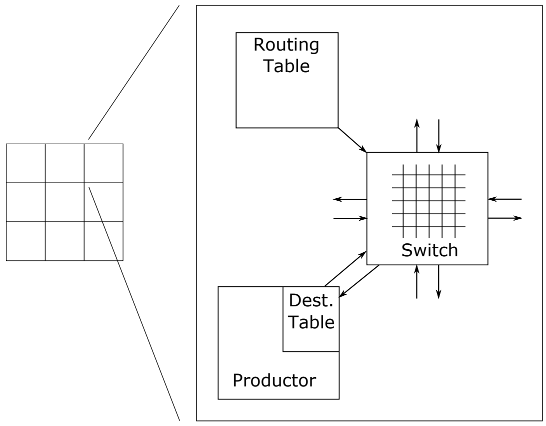


Fig. 3. A 3×3 StreetNet structure

required. For a device with a large number of PEs a network based interconnect, named the *StreetNet*, is proposed in this paper.

3.1 Network Architecture

A regular tile-based 2D network architecture is considered for StreetNet. Each tile contains a single PE and router, however adjacent tiles may be linked to share memory resources (discussed below in 3.6). Every PE has a *destination table* generated from the dependency graph. It lists the desired destinations and corresponding tokens to those destinations. Each router is connected to its local PE and four neighbouring tiles. Each router also has a routing table that is checked for destination. The tokens are broken into packets and forwarded to the neighboring tile towards the destination. A crossbar switch is used as the switching fabric in the router. Fig. 3 shows the structure of a StreetNet.

3.2 Dependency Graph

The mapping of PEs onto network architecture is based on a *dependency graph*. A dependency graph is a directed graph, where each vertex p_i represents a PE. Directed arcs represent non-zero communication paths between two PEs and are assigned a weight characterising the communication rate between the PEs. Fig. 4 shows an example of a dependency graph of nine PEs. This graph is used to map the PEs onto the network so that the most dependent PEs are placed close together in the expectation this will reduce communication latency and power consumption. The PEs are sorted by total incoming and outgoing traffic that was recorded during runtime, and mapped in this order. This is useful since the positions of the PEs with high traffic requirement have higher impact on the overall energy consumption. This dependency graph is updated during a sleep period (described in subsection 3.5) depending on runtime traffic statistics.

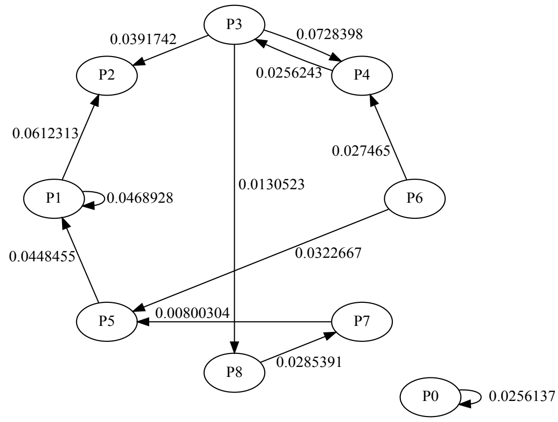


Fig. 4. An example of dependency graph with nine PEs

3.3 Mapping Techniques

In [8], an energy aware mapping technique is proposed for networks-on-chip (NoC) with a regular architecture. This technique is adopted in the StreetNet. The average energy consumed in sending one bit of data from a tile to a neighboring tile is calculated as

$$E = E_S + E_B + E_L \tag{1}$$

where E_S, E_B and E_L are the energy consumed at the switch, buffer and link. Since the energy consumed for buffering is negligible compared to E_L [8], (1) becomes

$$E = E_S + E_L \tag{2}$$

Now, if the bit traverses n hops to reach tile t_j from tile t_i, the average energy consumption is

$$E_{t_i,t_j} = n \times E_S + (n-1) \times E_L \tag{3}$$

For a system that involves a large number of processing elements, it is important to adopt efficient mapping and routing techniques so that total energy consumption is minimized and communication traffic does not exceed available bandwidth. Two different mapping techniques have been considered in this work: one is based on Simulated Annealing (SA) and the second is based on Branch-and-bound (BB) technique.

Simulated Annealing based Mapping Simulated Annealing (SA) [9] is a well known technique for solving optimization problems. It effectively optimizes solutions over large state spaces by making iterative improvement. It has a concept of *temperature* which is initially very high and keeps reducing in every step until it reaches the minimum temperature. For each temperature, it starts with

```
current_temp=MAX_TEMPERATURE;
previous_cost=INITIAL_COST;
current_mapping=randomMapping();
current_cost=cost(current_mapping);
do{
    while(attempts<MAX_ATTEMPTS){
        current_mapping=makeRandomTileSwap(current_mapping);
        new_cost=cost(current_mapping);
        ΔC=new_cost - current_cost;
        if (random(0,1)≤exp(-ΔC/current_cost×current_temp))
            current_cost=new_cost;
        else
            current_mapping=rollbackTileSwap(current_mapping);
    }
    if(toleranceTest(previous_cost,current_cost)||current_temp≤MIN_TEMPERATURE)
        done=1;
    else{
        previous_cost=current_cost;
        current_temp=getNextTemperature(current_temp);
    }
}while(!done);
```

Fig. 5. Simulated annealing algorithm for PE mapping

a random feasible solution and searches for better solutions with lower cost. This is a greedy algorithm. A tolerance test is done in every iteration to check if the cost is changing insignificantly over the last few temperatures or the temperature reaches a certain limit. Eventually, when the temperature goes below the minimum limit, it defaults to the greedy algorithm only. Fig. 5 illustrates the SA algorithm for PE mapping.

Branch-and-bound based Mapping In this mapping technique, a search tree is generated that represents the solution space. The root node corresponds to the state where no PEs are mapped. Each internal node represents a partial mapping and each leaf node is a complete mapping of PEs onto tiles. Fig. 6 shows the search tree of the solution space. For example, the node labelled $t_0 t_{n-1} t_1 ... t_{n-2}$ represents the placement in which PEs P_0, P_1, P_2, ... P_{n-1} are mapped to tiles t_0, t_{n-1}, t_1, ... t_{n-2} respectively.

The branch-and-bound (BB) mapping finds the solution node which has the minimum cost. The cost of mapping is calculated by the total energy consumed by all the PEs that are already mapped. The PEs are initially sorted based on their traffic demand obtained from the dependency graph. As the PEs with higher traffic demands dominate the overall energy consumption, they are mapped first to the unoccupied tiles to generate new child nodes. Each node has a table that stores the routing paths between its occupied tiles. When a child node is generated, the table from its parent node is inherited, and the routing

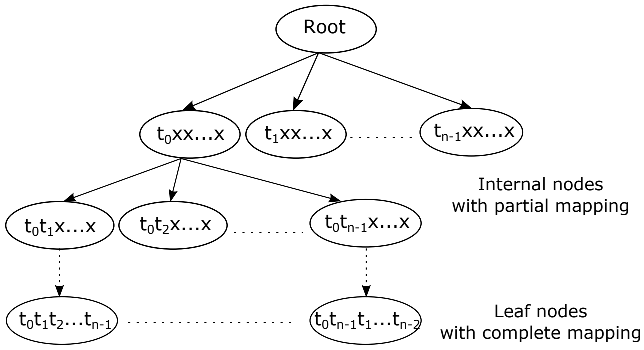


Fig. 6. Search tree of solution space

path to the new tile is added to the table. Then, each of the newly generated child nodes are examined to see if it is possible to generate the best leaf node later. The upper and lower bounds of the nodes are calculated to detect candidate optimal nodes. The upper bound of a node is the value that is no less than the minimum cost of its leaf nodes; the lower bound is defined as the lowest cost that its descendant leaf nodes can possibly achieve. If the cost or lower bound of a node is higher than the lowest upper bound that is already found so far, it is deleted without any expansion, because it is guranteed that the node cannot lead to the best mapping solution. The lower and upper bounds are updated after every step. All the nodes are traversed this way, and finally the node with minimum cost is accepted as the best mapping.

3.4 Routing

Wormhole routing is considered in this work because of limited buffering resources. In wormhole routing, packets are broken down into flow control digits (*flits*) and the flits are routed over the network in a pipelined fashion. The header flit contains routing information and leads the packet to the destination. Deterministic dimension-ordered routing is chosen in this work. In comparison to adaptive routing, deterministic routing requires less buffering space since no ordering is required for received packets [4]. Moreover, deterministic routing algorithms are livelock free. We use the west-first turn model [7] that prohibits north-to-west and south-to-west turns to make it deadlock-free.

3.5 Sleep Period

Street stores traffic statistics during runtime which are used periodically to update the dependency graph. The assignments of rules to PEs and their location within the network are refined based on this so that total traffic energy consumption is minimized. During this period, the execution is paused, analogously to human sleeping [11], and the memory contents as well as destination tables are re-arranged between PEs. However, the network structure and routing table remain unchanged as they are the fixed components of StreetNet. After the sleeping period, it continues to operate as before, but with improved performance.

3.6 Clustered PEs

If memory content exceeds the capacity of a PE, the flat architecture of Street's tiles allows the PE to use the memory resources of adjacent unused tiles. One of the PEs in the group of tiles acts as a *master PE*, and the destination table attached to it remains active. This master PE acts as source or destination of packets, the other routers of the cluster are used to forward the packets only. The PEs inside the cluster communicate through a local bus. If there are no unused adjacent tiles, the rule is moved to a new PE with free adjacent tiles and the contents are transferred during a sleep period.

4 Experiments

StreetNet was tested over network architectures ranging from 4 to 196 processing elements. As we have not yet developed any large-scale agents, dependencies were artificially created. For every architecture, 10 random dependency sets were generated. Each dependency set was used for mapping using both simulated annealing and branch-and-bound techniques. Fig. 7 shows the total energy consumption comparison between the mapping techniques. This shows that SA based mapping performs slightly better than BB based mapping, but when compared in terms of computation time, the latter significantly outperforms the former, as seen in Fig. 8. This indicates that BB mapping works much faster than SA mapping but the energy savings at execution time from the SA mapped solution may warrant the extra mapping time. StreetNet creates routing tables for all the routers as well. Since deterministic dimension-order routing has been considered in this work, routing tables do not change over time. As a result, ordered packet delivery and simplicity are ensured.

5 Conclusion

In this paper, a network-based communication platform, StreetNet, is proposed for the Street cognitive computer in which the PEs are mapped onto a 2D mesh architecture. The mapping is derived from a dependency graph that is obtained from runtime traffic statistics. This work introduces the concept of a periodic sleep period, during which the placement of the PEs is updated to improve overall energy efficiency. Branch-and-bound and simulated annealing based mapping techniques are discussed here. Experiments indicate that branch-and-bound mapping significantly outperforms simulated annealing based mapping when compared in terms of computation time. Clustered PEs are implemented in this work to accommodate large memories. The number and orientation of

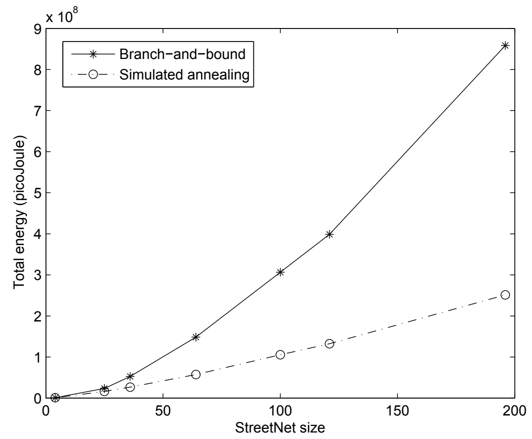


Fig. 7. Energy comparison between two mappings

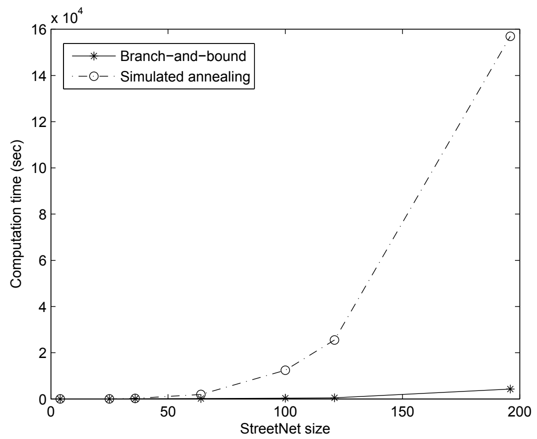


Fig. 8. Computation time comparison between two mappings

the tiles of a clustered PE will affect the overall energy consumption of the system to an extent that is yet to be investigated. Moreover, we plan to implement the mapping algorithm using Street Language so that Street can update PE mapping using its own resources.

References

1. Amaral, J.N.: A parallel architecture for serializable production systems. Ph.D. thesis, Citeseer (1994)

2. Amaral, J.N., Ghosh, J.: Parallel Processing for Artificial Intelligence 1, chap. Speeding up production systems: From concurrent matching to parallel rule firing, pp. 139–160 (1993)
3. Anderson, J.R.: ACT: A simple theory of complex cognition. American Psychologist 51(4), 355 (1996)
4. Bhattacharyya, S.S., Deprettere, E.F., Teich, J.: Domain-Specific Processors: Systems, Architectures, Modeling, and Simulation (2003)
5. Forgy, C.L.: OPS5 user's manual. Tech. rep., Computer Science Department, Carnegie-Mellon University (Jul 1981)
6. Frost, J., Numan, M.W., Liebelt, M., Phillips, B.J.: A new computer for cognitive computing. In: 14th IEEE International Conference on Cognitive Informatics & Cognitive Computing. Beijing, China (July 2015)
7. Glass, C.J., Ni, L.M.: The turn model for adaptive routing. SIGARCH Comput. Archit. News 20(2), 278–287 (Apr 1992), http://doi.acm.org/10.1145/146628.140384
8. Hu, J., Marculescu, R.: Energy- and performance-aware mapping for regular NoC architectures. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 24(4) (April 2005)
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (May 1983)
10. Laird, J.E.: The Soar Cognitive Architectures. The MIT Press, Cambridge (2012)
11. Landmann, N., Kuhn, M., Piosczyk, H., Feige, B., Baglioni, C., Spiegelhalder, K., Frase, L., Riemann, D., Sterr, A., Nissen, C.: The reorganisation of memory during sleep. Sleep Medicine Reviews (0), – (2014), http://www.sciencedirect.com/science/article/pii/S1087079214000264
12. Miranker, D.P.: TREAT: A New and Efficient Match Algorithm for AI Production Systems. Morgan Kaufmann (2014)
13. Richardson, T., Nicopoulos, C., Park, D., Narayanan, V., Xie, Y., Das, C., Degalahal, V.: A hybrid SoC interconnect with dynamic TDMA-based transactionless buses and on-chip networks. In: VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on (Jan 2006)

Developing Initial State Fuzzy Cognitive Maps with Self-Organizing Maps

Marcel Wehrle, Edy Portmann, Alexander Denzler, and Andreas Meier

University of Fribourg
Boulevard de Perolles 90, 1700 Fribourg, Switzerland
University of Bern
Engehaldenstrasse 8, 3012 Bern, Switzerland
{marcel.wehrle,alex.denzler,andreas.meier}@unifr.ch
edy.portmann@iwi.unibe.ch

Abstract. Using soft computing methods, the authors collect and process relevant user-generated information from the web. Through the use of self-organizing maps, fuzzy cognitive maps are constructed. The fuzzy cognitive map is a generated representation of the emergent web semantics of the dataset. In the next step, the fuzzy cognitive maps are enriched with related lexical content and stored in a graph database. This makes it possible for a human user to explore the maps in a visual way. Following a design science research approach, a prototype has been implemented as a proof of concept.

Keywords: Emergent web semantics, Fuzzy cognitive maps, Self-organizing maps

1 Introduction

Todays existing social web contains a set of relations that connect users through the Internet [1]. It primarily consists of human-understandable information (i.e., semantics); however, this information most often comes in unstructured form and is thus not straightforward to interpret by computers. However, the semantic web should enable computers to understand and respond to complex user queries based on their meaning [2]. Such an understanding, however, requires relevant information (i.e., data that have been given formal meaning by way of relational connection through some meaning negotiation process [3]) to be semantically structured. To this end, emergent web semantics is a possible answer to enhance the interaction between humans and machines.

This field consists of a set of methods and techniques for analysing the evolution of decentralized semantic structures in large-scale distributed information systems [4]. As inevitably required in the semantic web (i.e., to adaptively and dynamically address todays information explosion as naturally as possible), emergent semantics adopts a complex systems approach of addressing meaning by automatically creating semantics in a distributed system as an ensemble of relationships between syntactic structures [5]. Both the discovery of the proper interpretation of symbols (e.g., as a result of a self-organizing process performed by distributed agents) and the representation of the thus created semantics are taken into account.

To team up humans and computers, both need the ability to learn from each other. This may progress quite naturally because over time, our ability to process information and communicate it to others improves [6]. For example, through dynamic query interfaces [7], humans are able to adapt to computers, and through machine learning, computers can (better) adapt to humans. This computational intelligence consists of a toolbox of nature-inspired methods of computation to address the real worlds complexity to which conventional approaches (i.e., first principles modelling or statistical modelling) are ineffectual or impracticable [8]. It also embraces biologically-inspired algorithms (i.e., swarm intelligence and artificial immune systems), which can be seen as part of evolutionary computation, and includes broader fields such as natural language processing and data mining.

To support people in their searches, dynamic query interfaces should fit into a single users knowledge. For this purpose, the computer should rely on learning to defer to the average user, who uses language in a natural way. Through dynamic interfaces that integrate digital content into a humans life in seamless ways, a computer should even become adaptable to each individual user. With an automatically built-in knowledge graph (a fuzzy cognitive map (FCM)), computers may become (more) responsive to humans. These FCMs may be created automatically on the basis of self-organizing data mining algorithms (i.e., agents that crawl and aggregate social web data). A similar approach was already introduced by Jazzar & Jatan [9] by using FCMs in SOM-based intrusion systems. Normally, FCMs are constructed out of human knowledge [10]; consequently, they strongly depend on the subjective beliefs of the expert(s). Furthermore, the map itself is limited to be relatively simple and small and domain specific [11]. To overcome this, several efforts have been conducted by introducing algorithms to learn the FCM model structure. In general, there are two main proposed paradigms: Hebbian and genetic algorithms [12–16]. However, with the help of SOM, the aim of this framework is to find an initial state vector, which is constructed out of the collected data, that leads to a predefined FCM, similar to Kahn & Chong [17].

Section 2 introduces fuzzy logic, cognitive maps and FCMs. Section 3 describes self-organizing maps (SOMs), their functionality and how they can be used to construct initial FCMs. Based on SOMs, Section 4 presents graph databases as possible stores and query engines for emergent semantics [4]. In Section 5, the authors' framework is presented, and its architecture and single components are illustrated. Section 6 presents the prototype with all components. Finally, in Section 7, we draw conclusions and indicate possibilities for future research.

2 Fuzzy Cognitive Maps

The problem with today's semantic web is that the more complex it becomes, the less precise the statements (i.e., exact statements formulated in two-valued predicate logic) that can be made about it become. This is the message of the principle of incompatibility, which roughly states that high complexity is incompatible with high precision [18]. Therefore, anticipating an enhancement to two-valued logic, Zadeh introduced fuzzy logic as a tool for formalizing and representing the reasoning process and fuzzy logic systems, which are based on fuzzy logic and possess many characteristics attributed to intelligence [19]. Fuzzy logic effectively handles uncertainty, which is common in human reasoning, perception and inference and, contrary to some misconceptions, has a very formal and strict mathematical backbone (i.e., it is deterministic in itself, yet it allows uncertainties to be effectively represented and manipulated).

In more detail, fuzzy sets are graduated in the sense that membership in a fuzzy set is a matter of degree. A fuzzy set A, in a universe of discourse U, is defined by a membership function u_A that associates with each object u in U, the degree to which u is a member of A. A fuzzy set is basic if its membership function takes values in the unit interval $[0, 1]$. More generally, the membership function may take values in a partially ordered set.

FCMs are fuzzy structures that strongly resemble neural networks, and they have powerful and far-reaching consequences as a mathematical tool for modelling complex systems. Kosko [20] introduced FCMs as a fuzzy extension of the cognitive map pioneered by Axelrod [21], who used the map to represent knowledge as an interconnected, directed, bi-level logic graph. The underlying model behind FCMs is simple and effective because it can analyse the data using directed graphs and connection matrices [11].

FCMs are fuzzy signed directed graphs with feedback. The directed edge e_ij from causal concept C_i to concept C_j measures how much C_i causes C_j. The time-varying concept function $C_i(t)$ measures the non-negative occurrence of some fuzzy event (e.g., the strength of a sentiment, strategy or historical trend). The edges e_ij take values in the fuzzy causal interval [1,1]; $e_ij = 0$ indicates no causality, $e_ij > 0$ indicates a causal increase, and C_j increases as C_i increases (or C_j decreases as C_i decreases).

Finally, $e_ij < 0$ indicates a causal decrease or negative causality, and C_j decreases as C_i increases (and/or C_j increases as C_i decreases). Simple FCMs have edge values of $1, 0, 1$. Thus, if causality occurs, it occurs to a maximal positive or negative degree. As a direct consequence, the values provide a quick first approximation to an expert's causal knowledge.

A solution for integrating computer-understandable meaning in todays semantic web is often programmed from above, such as a creator of an ontology makes something and imparts it with his or her intelligence. Another approach uses a more bottom-up, decentralized method; bio-inspired techniques often involve the method of specifying a set of simple rules, a set of simple organisms that adhere to those rules and a method of iteratively applying those rules [1]. Because the FCMs are generated from social web information, only allowing FCM patterns that are stabilized over time, it involves continuous optimization, and its algorithms can be considered global optimization methods.

After several generations of rule application, it is usually the case that some forms of complex behaviour arise. Complexity gets built upon complexity until the end result is something markedly complex and, quite often, completely counterintuitive to what the original rules would be expected to produce. Evolutionary computation uses iterative progress, just like the permanent aggregating of information by web agents yields emergent semantics [4]. This emergent semantics may be managed with FCMs.

A FCM can model the relationships of various factors, depending on what logic the word vectors are constructing (e.g., similarity or semantics). They can have three different characteristics: (1) indicating a positive or negative causality of a relationship, (2) showing the strength of causal relationships with fuzzy values, and (3) dynamic causal links where changes affect concepts [16].

In the following section, we introduce SOMs that help create the emergent semantics underlying FCMs.

3 Generating Fuzzy Cognitive Maps with Self-Organizing Maps

Since the introduction of SOMs by Kohonen in 1982 [22], more than 7700 SOM-related research papers have been published, primarily in the fields of image analysis, speech recognition, signal processing and robotics [23]. A SOM is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. SOMs use a neighbourhood function to preserve the topological properties of the input space (abstraction). This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling [22]. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training constructs the map using input examples (a competitive process, also called vector quantization), whereas mapping automatically classifies a new input vector.

A SOM consists of components called nodes or neurons i. Associated with each node i is a weight vector w_i of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. To overcome the border effect, spherical grids have been introduced [24, 25]. The SOM describes a mapping from a higher-dimensional input space to a lower-dimensional map space (visualization). The procedure for placing a vector from data space onto the map is to find the node with the closest (i.e., smallest distance metric, such as Euclidean distance) weight vector to the data space vector. In general, in the initialization phase, the number of input units and the topology of the output layer are first determined. However, this is difficult because of the increasing amount of available information. To overcome this limitation, several improvements to SOMs have been introduced, mainly by adaption such as growing grid models [26], hierarchical feature maps [27], growing hierarchical maps [28] or tree-structured maps [29].

Large SOMs exhibit emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself [30].

4 Storing Web Semantics in Graph Databases

Graph databases serve as a digital storage medium for graphs. At the same time, they provide users with features and functionalities commonly used in the domain of graph theory. Within the database landscape, they are classified as belonging to the group of non-relational data models [31]. This group of databases is viewed as an extension to traditional relational databases. Four different types of databases are prominent representatives in this group, with key-value stores and big table clones focusing on handling big data sizes and documents and graph databases specializing in complex data [32]. Considering that structured knowledge is viewed as highly interconnected and thus as complex data, this backs the proposal for choosing a graph database for storing from a technical perspective.

Not all graph databases are the same, however. Some are constructed on top of other data models, while others are standalone solutions. Other differences exist based on the purpose and environment that they have been developed for. While web-based solutions focus on maintaining low latency times for queries, others aim at handling large graphs by scaling horizontally, whereas others are specifically designed for processing algorithms as fast as possible by storing the entire graph in memory [33].

Because FCMs are highly interconnected directed graphs, it appears reasonable to represent them in graph databases. This allows optimized queries to the database [31]. Furthermore, most graph databases allow visual exploring of the underlying FCMs through a web interface. This facilitates the interaction between humans and the FCMs. Graph databases are a valuable tool for representing the web semantics of a given dataset. Another advantage lies in the computational possibilities offered by graph algorithms. Computing shortest paths, clusters and recommendations are tasks that graph algorithms are particularly suited [31]. This may allow deeper insights into the FCMs.

5 Architecture

The framework is built upon a 3-layer software architecture. Figure 1 shows the underlying design. As an external data source, semistructured content from the web is accessed through a web crawler, an API or a data dump. The first layer consists of all data processing steps, beginning with the manipulation of the data. Here, various options are possible. The content can be normalized through tokenization, stemming or lemmatizing. If the amount of data should be reduced, stopwords or particular parts of speech (through part-of-speech tagging) can be eliminated. For further manipulation of the data, the concepts (i.e., words or phrases) need to be represented as word vectors (e.g., Mikolov et al. [34] or Maas et al. [35] provide unsupervised vector-based approaches). These word vectors form the input vectors to create the SOM. The topology, which results as an output from the SOM, can already provide an adequate overview of the data. It shows clusters of similar concepts and where the main topics of data are located.

The output of the SOM can be used as an input matrix to generate the FCMs. The matrix already contains the information needed to draw the concepts with the causal relationship among these concepts. For the estimation of the causal relationship strength, depending on the use case, one of the proposed learning algorithms (see section 2) can be used.

The application layer includes a web framework to provide accessibility over a web browser. This facilitates the application being used by a broader audience. Because the framework is built upon a graph database, the only restriction is an available interface to the database. Finally, the user interface is an important component to complete the user experience. This interface should allow querying and exploring the underlying datasets such that the user can enlarge his personal knowledge.

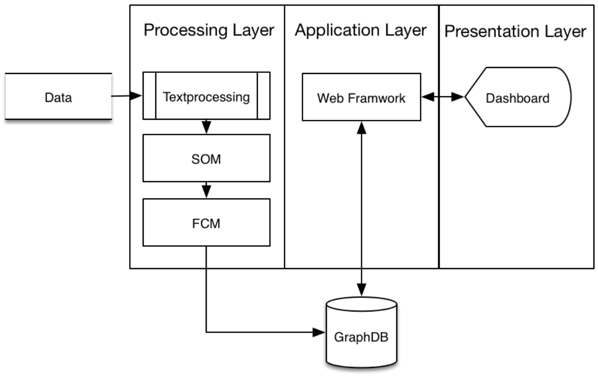


Fig. 1. Framework Architecture

6 Prototype

As a data source, the authors used the dump of a stack exchange on the topic computer science. A primary function of a stack exchange is to share knowledge from user to user. The exchange is performed through questions and answers. Overall, 13492 positively rated posts were considered for the evaluation. Users of the stack had used 463 tags to mark their questions, mostly algorithms (2267), complexity theory (1143) and formal languages (767). Various recent studies have already used the stack dataset [36–38] as a data source, but none considered FCMs.

A framework was established to analyse the data. Followed by the processing steps shown in Figure 2, the data were transformed into FCMs. As soon as the data are normalized and tokenized (1), bigrams are identified and a word vector $\boldsymbol{w_i}$ is constructed as a representation of these bigrams (2). In the prototype,

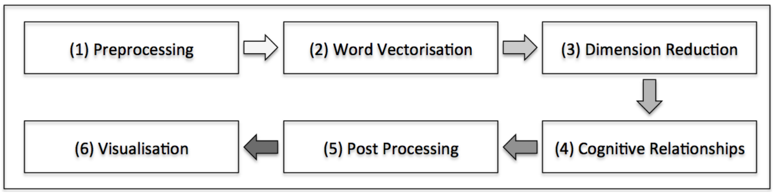


Fig. 2. Processing steps

the authors used a context-prediction vector, CBOW a computationally efficient model by Mikolov [34]. CBOW learns to predict words in the middle of a symmetric window. The window is based on the sum of $\boldsymbol{w}$ of words in the window. However there are various other possibilities to build semantic word vectors [39]. The word vectors $\boldsymbol{w_i}$ are generated for each concept in the vocabulary with a dimensionality of 450.

These vector representations $\boldsymbol{w} = \boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_n}$ are the input source for the SOMs to reduce their dimensionality (3). During the evaluation process, a total of 150 training epochs with a learning rate decreasing from 0.6 to 0.02 have been seen as promising. The size of the map is 225x225 which leads to an output layer of 50625. Figure 3 shows the topology of the explored dataset. The produced distance matrix is the input source for the FCM.

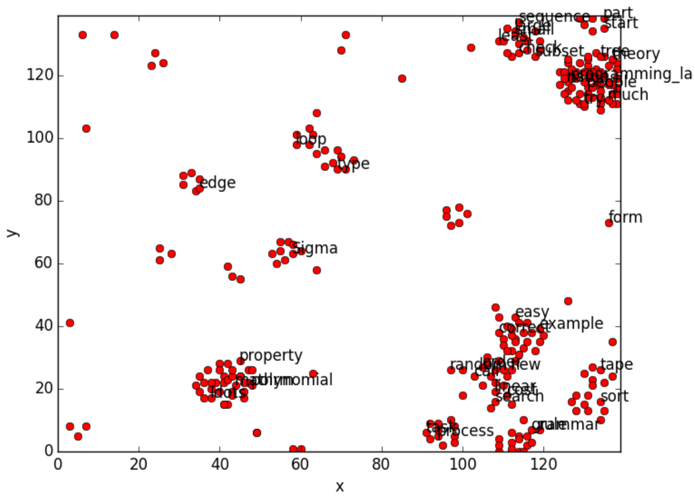


Fig. 3. 2D - Visualization of the SOM of the principal concepts.

The visualization of SOMs also has the advantage of showing centres of concepts where the data are spread. As is often the case in FCMs, weights are determined by human interaction, and learning algorithms are used to eliminate the subjectiveness of weights. In this framework, the weights w_ij are adjusted through the closeness of concepts C_j and C_i, $i = 1, 2, ..., N, j = 1, 2, ..., N$, in the produced distance matrix of the SOM until a stop criterion is reached. This leads to an initial state vector of the FCM. The setup for the value of this weight is between $[0, 1]$ (4). The related pseudocode is shown in Algorithm 1. Then, the fuzzy expressions medium, strong and very strong are mapped to the numerical values of the relationships (Table 1).

Algorithm 1 FCM algorithm

```
procedure CONSTRUCTION FCM
    weight ← inputDistanceSOM
    nodes ← concepts
    minDistanceValue
loop:
    for i, j in nodes do
        if weight(nodes(i, j)) > minDistanceValue then
            Construct E{i, j}
            Label E{i, j}
    close;
```

Algorithm 1 allows the connectivity of the graph to be adjusted by varying the minimal distance value. After the FCM of the dataset is constructed, it is post-processed (5) by adding metadata to each node. These metadata are delivered through an interface to wikidata. All the values, included in the metadata as properties, are stored through a database layer in a graph database. The authors selected OrientDB as a database. OrientDB is a hybrid system that offers a graph database and a document database. It has all the functionality required for this experiment, especially because it possesses a good query system (i.e., it can calculate shortest path operations).

Fuzzy membership functions	Fuzzy regions	Deffuzzified value (weight)
medium	(0.6, 0.8)	0.7
strong	(0.7, 0.9)	0.8
very strong	(0.9, 1)	1

Table 1. Fuzzy expression mapping

The final component of the framework is the user interface. OrientDB already provides a complete interface for querying and analysing graph-related data [40].

It also allows direct interactions, so users can explore and manipulate data. In this way, the data can be visualized in a graph-based environment (6). Currently, the produced output of the processed data can be accessed in [41]

7 Conclusion

The authors propose a framework for analysing data from web sources that develop web semantics represented through fuzzy cognitive maps. The framework includes a mixture of different algorithms and technologies. The FCM allows users to explore underlying web sources enriched with metadata. (Hidden) relationships between concepts can be uncovered. The underlying framework also allows direct user interaction to adjust causal relationships in the graph. In this way, a machine-human interaction is established, where both can learn from each other [5]. Note that the prototype only produces the initial state FCM on a limited dataset.

An advantage of this approach is certainly it simplicity. It can be applied on nearly all user-produced content to generate the related FCMs. This already provides a proper overview on which concepts are important and how they are interconnected. FCMs allow the inclusion of uncertainty and vagueness, both inherent in the human language. Thus, a better interpretation of human-produced content may be possible. They also allow a certain flexibility in the setup of the framework. Through the enrichment with metadata, it may also be used to discover knowledge. Limitations are given by the used technologies and inputs. User-generated input is never perfect and often includes spelling or grammatical mistakes. These mistakes could not always be removed. Reducing the dimensionality through SOM simplifies the data, but at the same time, information is lost. Finally, FCMs with many concepts rapidly become unclear and the user may lose focus.

A related framework was already used for a trend discovery project of the Swiss Commission for Technology and Innovation (CTI). In this project, trends, relevant for the tourism industry, have been identified from social media as an input for performance management. The main difference was the temporal component of the framework to view the transformations over time, which allows the change of concepts relating to their relevance and their relationships to be observed. Furthermore, the dataset grows through continuous crawling, and the weights are adjusted through a learning algorithm.

The basic concept forms the basis for the construction of a granular knowledge cube [42]. This concept combines an amplified multi-source knowledge base together with a knowledge carrier finder system. This approach has the goal of connecting a knowledge seeker and a knowledge carrier in logical and simplified way.

Acknowledgements

Application of Fuzzy Cognitive Map in the Performance Management in the tourism industry of Graubunden (Einsatz von Fuzzy Cognitive Maps im strategischen Performance Management der Tourismusbranche Graubnden). This project has been supported, in part, by the Swiss Commission for Technology and Innovation CTI under grant number 14943.1 PFES-ES

References

1. Portmann, E.: The FORA Framework - A Fuzzy Grassroots Ontology for Online Reputation Management. Eds. Andreas Meier, and Edy Portmann. The FORA Framework - A Fuzzy Grassroots Ontology for Online Reputation Management. Springer, Heidelberg: (2013)
2. Berners-Lee; T., Hendler, J., Lassila, O.: The Semantic Web. In: Scientific American Magazine, 284.5,pp. 28–37. (2001)
3. Rappaport, W. J.: What did you mean by that? Misunderstanding, negotiation and syntactic semantics. In: Journal Minds and Machines. Vol. 13, No. 3. pp. 397–427.(2003)
4. Cudre-Mauroux, P.: Emergent Semantics. In: Encyclopedia of Database Systems. Springer, Heidelberg: (2009)
5. Portmann, E., Kaufmann M., Graf C.: A Distributed, Semiotic-Inductive, and Human-Oriented Approach to Web-Scale Knowledge Retrieval. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012). Hawaii: (2012)
6. Ashby, W.R.: An introduction in cybernetics. Chapman & Hail Ltd., London. (1956)
7. Shneiderman, B., Plaisant, C.: Designing the User Interface (4th Ed.). Person/Addison-Wesley, Boston: (2005).
8. Kacprzyk, J.: Computational Intelligence and Soft Computing: Closely Related but Not the Same. In: Accuracy and Fuzziness. A Life in Science and Politics: A Festschrift book to Enric Trillas Ruiz by R. Seising and L. A. Arglles Mndez (Ed.). Springer, Heidelberg: (2015)
9. Jazzar, M., Bin Jantan, A.: Using Fuzzy Cognitive Maps to Reduce False Alerts in SOM-based Intrusion Detection Sensors. In: Proceedings of the 2nd Asia International Conference on Modelling & Simulation. (2008)
10. Aguilar, J.: A survey about fuzzy cognitive maps papers (Invited Papers). In: International Journal Computing Cognition 3, pp.27–33. (2005)
11. Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Genetic learning of fuzzy cognitive maps. In: Fuzzy Sets and Systems 153. pp.371–401 (2005)
12. Dickerson, J. A., Kosko, B.:Virtual Worlds as fuzzy cognitive maps, Presence 3 (2) pp.173–189. (1994)
13. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P., Vrahatis, M. N.: Fuzzy cognitive maps learning using swarm optimization. In: Journal of intelligent information systems. (2005)
14. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P.: Fuzzy cognitive map learning based on nolinear Hebbian rule. Australian Conference on Artificial Intelligence. pp.256–268. (2003)

15. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P.: Active Hebbian learning algorithm to train fuzzy cognitive maps. In: International Journal in Approx. Reasoning. pp.219–2249. (2004)
16. Papageorgiou, E. I., Stylios, C. D.: Fuzzy cognitive maps. Handbook of Granular Computing. John Wiley & Son Ltd, Publication Atrium, Chichester, England (2008).
17. Khan, M. S., Chong, A.: Fuzzy cognitive maps with genetic algorithm. In: Proceedings of the 1st Indian Conference on Artifical Intelligence. (2003).
18. Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes. In: IEEE Trans. SMC, SMC-3 (1), pp. 28–44. (1973)
19. Zadeh, L. A.: Fuzzy sets. In: Information and Control, vol. 8, issue 3, pp. 338-353. (1965)
20. Kosko, B.: Fuzzy cognitive maps. In: Int. J. Man-Machine Studies, vol. 24, pp. 65-75. (1986)
21. Axelrod, R.: Structure of Decision: the Cognitive Maps of Political Elites. Princeton University Press, Princeton. (1976)
22. Kohonen, T., Honkela, T.: Kohonen network. In: Scholarpedia, 2(1):7421, (2007)
23. Bibliography of SOM Papers, `http://www.cis.hut.fi/research/som-bibl`. Accessed (10-08-2015)
24. Daisuke, N., O. Matashige. Application of spherical SOM in clustering. In: Proceedings of Workshop on Self-Organizing Maps, WSOM03, Kitakyushu, Japan. (2003)
25. Farid, B., Biela, E. P., Jack-Grard P.: Self organizing spherical map architecture for 3d object modeling. In: Proceedings of Workshop on Self-Organizing Maps, WSOM03, Japan. (2003).
26. Fritzke, B.: Growing grid - a self organizing network with constant neighborhood range and adaption strength. In: Neural Processing Letter, vol. 2, no.5, pp.9–13 (1995)
27. Miikkulainen, R.: Script recognition with hierarchical feature maps. In: Connection Science, vol. 2, pp.83–101. (1990)
28. Rauber, A., Dittenbach, M., Merkl, D.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. In: IEEE Transactions on Neural Networks, vol. 13, n. 6, pp. 1331–1341. (2002)
29. Koikkalainen, P.: Tree structured self-organizing maps. In: Kohonen Maps, Oja E., Kaski S. Amsterdam, Netherlands: Elsevier, pp.121–130. (1999)
30. Ultsch, A.: Emergence in Self-Organizing Feature Maps. In: Ritter, H., Haschke, R.: Proceedings of the 6th International Workshop on Self-Organizing Maps. (2007)
31. Robinson, I., Webber J., Eifrem E.: Graph Databases. OReilly Media Inc, pp. 50–60. (2013)
32. Buerli, M.: The Current State of Graph Databases. Departement of Computer Sience, Cal Poly, San Luis Obispo. (2012)
33. Shao, B., Wang, H., Xiao Y.: Managing and mining large graphs: systems and implementations. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD 12, p. 589–592, ACM, New York. (2012)
34. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: arXiv 1301.3781v3, Cornell University Library, New York. (2013)
35. Maas, A., Daly, R., Pham, P., Huang D., Ng, A., Potts C.: Learning Word Vectors for Sentiment Analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol 1, pp. 142–150 (2011)

36. Barua, A., Thomas, S., Hassan, A.E.: What are developers talking about? An analysis of topics and trends in Stack Overflow. In: Empirical Software Engineering , vol 19, issue 3, pp. 619–654. (2014)
37. Vasilescu, B.: StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. In: Social Computing (SocialCom), 2013 International Conference, pp. 188–195. (2013)
38. Asaduzzaman, M. Mashiyat, A., Roy, C., Schneider, K.: Answering questions about unanswered questions of stack overflow. In: MSR '13: Proceedings of the 10th Working Conference on Mining Software Repositories. (2013)
39. Baroni M., Dinu, G., Kruszewski, G. Don't count, predict! A systematic comparison of context-count vs context-predicting semantic vectors. Association of Computer Linguistics. (2014)
40. OrientDB - OrientDB Multi-Model NoSQL Database, `http://orientdb.com/orientdb/`. Accessed (10-08-2015)
41. Prototype FCM on StackOverflow Computer Science Data, `http://knowtology.com/orientdb/`. Accessed (10-08-2015)
42. Denzler, A.; Wehrle, M.; Meier, A.: A granular knowledge cube. World Academy of Science, Engineering and Technology, International Science Index, Computer and Information Engineering. (2015)

Property-based Semantic Similarity: What Counts?

Silvia Likavec and Federica Cena

Università di Torino, Dipartimento di Informatica, Torino, Italy
{likavec,cena}@di.unito.it

Abstract. Similarity, one of six Gestalt principles, is one of the most intuitive ways to perceive the world and categorise the objects surrounding us. The notion of similarity plays an important role in many areas, and it is important to simulate human perception of similarity in order to obtain satisfying results in various applications. We draw our inspiration from Tversky's work on similarity and define property-based similarity for ontological concepts taking into account their common and distinctive features and their values. We also discuss some possible ways to improve the property-based similarity.

Keywords: Similarity, properties, ontology

1 Introduction

It is inherent to human nature to try to categorize objects surrounding us, finding patterns and forms they have in common. One of the most intuitive ways to relate two objects is through their *similarity*. Similarity is one of the six Gestalt principles which guide the human perception of the world, the remaining ones being: Proximity, Closure, Good Continuation, Common Fate, and Good Form.

According to Merriam Webster "similarity" is a quality that makes one person or thing like another and "similar" means having characteristics in common. There are many ways in which objects can be perceived as similar, such as having similar color, shape, size, texture etc. But if we move away from just visual stimuli, we can apply the same principles to define the *semantic similarity* of two objects. This leads to a similarity based on features these two objects have in common, and consequently, the lack of distinctive features characterising each object.

The concept of semantic similarity can be encountered in various fields, from Natural Language Processing (NLP) and Information Retrieval to Semantic Web. In this work we deal with the semantic similarity of concepts in domain ontologies (Gruber, 1993, Guarino and Poli, 1995), where concepts are distinguished by the properties associated to them. The usage of ontologies to represent various domains accounts for both similarities and differences among domain objects as well as generic objects and very specific ones.

Our inspiration comes from Tversky's work on Features of Similarity (Tversky, 1977) and we try to apply his ideas to similarity among ontological objects. More precisely, two objects are similar if they both are defined having the same properties with the same values. In addition to this simple notion of similarity, we explore how this similarity can be improved by considering relevance for properties or relevance for values or hierarchical relationships among values, Throughout this work we use domain of recipes to provide examples and explain our approach and reflections.

The rest of the paper is organised as follows. In Section 2, we provide a brief background on ontologies for knowledge representation and on the treatment of properties in OWL. We give the details of how to calculate the property-based similarity for instances in the domain ontology in Section 3 and then we look into some possible ways to improve the property-based similarity in the ontology in Section 4. We summarise the most relevant related work which regards the semantic similarity in Section 5. Finally, we conclude in Section 6.

2 OWL ontologies and knowledge representation

In various fields, from e-commerce and e-learning to cultural heritage, medicine, digital libraries etc., it is possible to describe the concepts of the domain by using the properties of these concepts and their respective values. The ones that immediately come to mind are ontologies (Antoniou and van Harmelen, 2008, Allemang and Hendler, 2008) and linked open data (Bizer et al., 2009), where properties are prominent elements of the domain and contribute to the description of domain concepts.

In this work we deal with ontologies, powerful and expressive formalisms which make it possible to explicitly specify domain elements and their properties, as well as relationships which exist among domain elements. Also, rigorous reasoning mechanisms are associated with ontologies. One standard formalism for representing ontologies is OWL.[1]

Throughout this work we would use domain of recipes to provide examples and explain our approach and reflections.

2.1 Properties in OWL

In ontologies expressed in OWL *properties* are used to describe domain elements and express their features. There are two kinds of properties in OWL:
(i) *object properties* describing relations among individuals and
(ii) *data type properties* providing relations among individuals and data type values.
Object properties and datatype properties are defined as instances of the built-in OWL classes owl:ObjectProperty and owl:DatatypeProperty, respectively. Both are subclasses of the RDF class rdf:Property. Here, we only consider object properties, and leave the treatment of data type properties (such as literal values) for future analysis, since it is more complex.

The *property axiom* is used to define the characteristics of a property. Usually, it defines its *domain* and *range*. rdfs:domain links a property to a class description, whereas rdfs:range links a property to either a class description or a data range. For example:

```
<owl:ObjectProperty rdf:ID="has_ingredient">
  <rdfs:domain rdf:resource="#Recipet"/>
```

[1] http://www.w3.org/TR/owl-ref

```
  <rdfs:range rdf:resource="#Food"/>
</owl:ObjectProperty>
```

defines a property `has_ingredient` which connects the elements of `Recipe` class to the elements of `Food` class.

Equivalent properties are defined with owl:equivalentProperty.

Properties can be explicitly defined for the classes and can be used to define classes with property restrictions. Our approach to similarity is best illustrated when considering instances in the ontology, hence we will provide here a brief description of properties for instances.

2.2 Instances and their properties

An instance in the ontology are characterised by its class membership, individual identity and property values. An instance inherits its properties from the classes it is an instance of and it has a specific value associated to each property. For example:

```
<Recipe rdf:ID="Herbed_Asparagus">
  <has_ingredient rdf:resource="#Asparagus"/>
  <has_ingredient rdf:resource="#Parmesan"/>
  <has_ingredient rdf:resource="#Herbs"/>
  <has_origin rdf:resource="#Italy"/>
  <suitable_for_diet rdf:resource="#"Vegetarian"/>
</Recipe>
```

defines a recipe `Herbed_Asparagus` which has ingredients: asparagus, parmesan and herbs, originates from Italy and is suitable for vegetarians.

3 Property-based similarity

First of all, let us have a look at an example which should clarify the basics of our approach. We consider the domain of recipes where properties such as `has_ingredient`, `has_origin`, `suitable_for_diet` are defined. These properties have one or more values assigned to them. Intuitively, the similarity among recipes depends on the property-value pairs they have in common. Consider for example the following recipes: `Asparagus_Parmigiana` and `Herbed_Asparagus_With_Parmesan_Cheese`. They both have ingredients: `Asparagus`, `Butter`, `Parmesan`, `Pepper` among others and are both suitable for vegetarian diet. On the other hand, `Indian_Style_Chicken` has only `Butter` in common with any of them and is not suitable for vegetarians. So the asparagus dishes are definitely more similar among themselves than any of them with the chicken dish.

Hence, in order to determine similarity among two objects, we want to consider both, their common features and distinctive features for each of them. To this aim we use Tversky's feature-based model of similarity (Tversky, 1977):

$$\textsc{sim}_T(O_1, O_2) = \frac{\alpha(\psi(O_1) \cap \psi(O_2))}{\beta(\psi(O_1) \setminus \psi(O_2)) + \gamma(\psi(O_2) \setminus \psi(O_1)) + \alpha(\psi(O_1) \cap \psi(O_2))}. \quad (1)$$

where $\psi(O)$ is the function describing all the relevant features of the object O, and $\alpha, \beta, \gamma \in \mathbb{R}$ are constants which permit different treatment of the various components. For $\alpha = 1$ common features of the two objects have maximal importance and for $\beta = \gamma$ non-directional similarity measure is obtained. In our approach we have $\alpha = \beta = \gamma = 1$.

We will be using the following notation:

- *common features of O_1 and O_2:* $\textsc{cf}(O_1, O_2) = \psi(O_1) \cap \psi(O_2)$,
- *distinctive features of O_1*: $\textsc{df}(O_1) = \psi(O_1) \setminus \psi(O_2)$ and
- *distinctive features of O_2*: $\textsc{df}(O_2) = \psi(O_2) \setminus \psi(O_1)$.

Using this notation and setting $\alpha = \beta = \gamma = 1$ the formula (1) becomes:

$$\textsc{sim}_T(O_1, O_2) = \frac{\textsc{cf}(O_1, O_2)}{\textsc{df}(O_1) + \textsc{df}(O_2) + \textsc{cf}(O_1, O_2)}. \tag{2}$$

Since each of the domain objects has a number of property-value pairs describing it, for each property p we will have to calculate how much it is responsible for common features among these objects, as well as for distinctive features of each of them. We denote these values by $\textsc{cf}_p$, $\textsc{df}_p^1$ and $\textsc{df}_p^2$. We consider equal the properties defined with owl:EquivalentProperty.

3.1 Similarity among instances

In this work we present our approach only for instances of classes, although it can be extended to classes defined with their properties and to classes defined as property restrictions (see Cena et al. (2012)). The essence of property-based similarity calculation lies in simple comparison of the property-value pairs for each instance. Let us assume that the property p has h' different values in O_1 and h'' different values in O_2, and k is the number of times O_1 and O_2 have the same value for p, then

$$\textsc{cf}_p = \frac{k^2}{h'h''}, \textsc{df}_p^1 = \frac{h'-k}{h'} \text{ and } \textsc{df}_p^2 = \frac{h''-k}{h''}.$$

Let us assume that the objects O_1 and O_2 have properties $p_1, \ldots, p_n$ in common. We can repeat the above process for each property $p_i, i = 1, \ldots, n$.

Now, there are two possible ways to calculate similarity between O_1 and O_2.

First, we can obtain all common and distinctive features of O_1 and O_2:

$$\textsc{cf}(O_1, O_2) = \Sigma_{i=1}^{n} \textsc{cf}_{p_i} \qquad \textsc{df}(O_1) = \Sigma_{i=1}^{n} \textsc{df}_{p_i}^1 \qquad \textsc{df}(O_2) = \Sigma_{i=1}^{n} \textsc{df}_{p_i}^2$$

where n is the number of properties O_1 and O_2 have in common. The similarity between two instances O_1 and O_2 is then calculated using the formula (2):

$$\textsc{sim}(O_1, O_2) = \frac{\textsc{cf}(O_1, O_2)}{\textsc{df}(O_1) + \textsc{df}(O_2) + \textsc{cf}(O_1, O_2)}.$$

This method for property-based similarity of objects in the ontology was first introduced in (Cena et al., 2012) classes defined with property restrictions but only for value restrictions. It was further developed to include cardinality restrictions and applied to categorization of shapes in (Likavec, 2013).

Second, we can calculate partial similarities w.r.t. each property $p_i, i = 1, \ldots, n$:

$$\textsc{sim}_{p_i} = \frac{\textsc{cf}_{p_i}}{\textsc{df}^1_{p_i} + \textsc{df}^2_{p_i} + \textsc{cf}_{p_i}}$$

and then use these similarities to calculate the total similarity between O_1 and O_2 as:

$$\textsc{sim}(O_1, O_2) = \Sigma^n_{i=1} \textsc{sim}_{p_i}.$$

4 Improving property-based similarity

The above presented base case property-based similarity provides high rates of similarity among objects which can be used in many applications. We still did not perform the thorough evaluation but we evaluated it in the field of user interest propagation and obtained very satisfying results (Cena et al., 2012). But, while performing the second evaluation in this field, we became aware that in certain domains, this property-based similarity of domain objects can be improved w.r.t. various aspects. We will discuss here some of them.

4.1 Relevance of properties

When defining the concepts of a domain, not all the properties play an equal role. Hence, it is possible to introduce the *relevance of properties* and assign different importance to different properties in the domain. Actually, the relevance of a property can be considered as the capacity of the property to determine the similarity between two entities. For example, in the recipe domain, the property `has_ingredient` is far more important than `has_author` and the two recipes with the same ingredients would be considered more similar than the two recipes with the same author. So, the property `has_ingredient` would have a higher relevance factor than prophas_author.

There are various approaches to calculation of property relevance in a domain. It can be declared *a priori* and although effective, this solution may not be very feasible for a huge domain. Also, it is possible to introduce an automatic method to determine the relevance of properties. One possibility is to compute the similarity of concepts and then to calculate the relevance factor for each property as the square of the average similarity between concepts with the same value for that property.

4.2 Property or underlying hierarchy?

First of all, some aspects of the domain can be seen as properties, as well as underlying hierarchy. So the question is, which way of modelling of the domain would provide better similarity with human judgement. For example, in the recipes domain, the concepts corresponding to dish type can be easily organised into a hierarchy and we can have all the instances be instances of certain `Dish_Type` classes. On the other hand, we can simply have a property `dish_type` and have all the recipes be instances of `Recipe`.

4.3 Relevance of values

One of the problems with the approach in which all the values for properties are treated equally is that they might not contribute to the overall similarity with the same degree, since some values might be more important in a certain context than the other. For example, if we consider recipes and the `has_ingredient` the values `beef` or `asparagus` would be more important than `salt` or `pepper`. Hence, we come to the point where we might need to introduce relevance for values, along the lines for relevance for properties. These would have to be proposed by domain experts or calculated by an algorithm designed for this purpose.

4.4 Hierarchy of values

Another possible improvement of property based similarity is to take into account the underlying hierarchy which might exist among the concepts used as values for properties. For example, if we consider recipes and the property `has_ingredient`, one recipe can have ingredient `Fusilli` and the other one `Spaghetti`. Although these two concepts are not equal, they could be considered equal or equal to a certain degree (e.g. 80% equal), since they are both types of pasta, and are descendants of `Pasta` concept. So it might be possible to consider "almost equal" direct descendants of a certain concept and even less equal second degree descendants of a certain concept.

5 Related work

There are various approaches to calculating similarity among concepts, depending on the data structure used to represent the domain and on the amount and type of data available about the concepts of the domain. The principal approaches to similarity calculation are the following: (i) information content-based methods, (ii) distance-based methods and (iii) feature-based methods. Various hybrid methods combine some of the above methods.

In his seminal paper, Resnik (1999) proposes to calculate the semantic similarity of concepts by calculating the information content in an is-a taxonomy of the closest class subsuming both compared concepts. This similarity measure is given by the negative logarithm of the probability of occurrence of the class in a text corpus. Another important information-theoretic definition of similarity is introduced by Lin (1998) where the similarity among concepts is calculated taking into account the shared information for the two concepts and the amount of information needed to fully describe them.

The origins of the distance-based approach go back to Rada et al. (1989) where the ontology graph structure is used to calculate the *distance* between nodes (i.e., the number of edges or the number of nodes between the two nodes) as a measure of their similarity. Leacock and Chodorow (1998) use the normalised path length in WordNet (Fellbaum, 1998) between all the senses of the concepts being compared. The semantic similarity is computed as a negative logarithm of the ratio between the number of nodes in the path which connects the given concepts and the maximum depth of the taxonomy. Wu and Palmer (1994) take into account the depths of the given words in the taxonomy and the depth of their common subsumer in their similarity measure.

Pirrò and Euzenat (2010) introduced a FaITH semantic similarity measure which uses Tversky's feature-based model and calculates the saliency of the features using a new information content approach based on the ontology structure. This new framework permits to calculate semantic similarity, as well as semantic relatedness and can be used to rewrite the existing similarity measures so that they can also compute semantic relatedness.

Smyth (2007) calculates the similarity by taking into account individual features of concepts and by assigning to each feature its own similarity function and the weight which helps distinguish the importance of individual features..

A semantic similarity measure for OWL objects introduced by Hau et al. (2005) is defined as a ratio between the shared and total information content of the two objects. The information content is calculated from the objects' description sets containing all the statements describing the given objects and is based on the number of new RDF statements that can be generated by applying a certain set of inference rules to the predicate.

The similarity measure introduced by Zadeh and Reformat (2013) is similar to ours in the sense that it uses Tversky's feature-based model for calculating similarity and then calculates object's common and distinctive features by observing all the relations the objects have in the given ontology.

In the realm of "Conceptual Spaces" proposed by Gärdenfors (2004) the concepts can be seen as convex regions in a conceptual space, whereas instances correspond to points. The conceptual spaces are constructed using primitive quality dimensions which represent various qualities of objects (e.g., color, shape, size). These dimensions of conceptual spaces provide the means for determining similarity between concepts and instances which can be defined as the inverse of their distance in the space.

Recently, Conceptual Spaces have been integrated with ontological formalisms to form hybrid knowledge bases by Lieto et al. (2015). Since the points are represented as vectors of the point coordinates (representing various object dimensions), their mutual similarity is calculated as cosine similarity.

6 Conclusions and future work

In this work we present an approach to calculate similarity based on properties defined in an ontology, as well as insights on which other factors can be included to improve this similarity in different contexts. We limited ourselves to presenting the approach only for the instances in the ontology, although the approach can be applied to classes and classes defined as property restrictions as well. In addition, this approach can be applied to linked open data Bizer et al. (2009) or any other structure where the objects are described by means of their properties. For example, it would be interesting to apply our measure of similarity to ConceptNet Speer and Havasi (2013), where an edge which connects two nodes can be seen as a property and a target concept as its value.

In the case presented here, the prerequisite is the ontology with explicitly defined properties for classes, rather than only a simple taxonomy of concepts. We only dealt with object type properties in this work, since data type properties, such as literals, require more complex analysis.

One of the limitations of the present approach, known for Tversky's notion of similarity, is that in the case of concepts with few properties defined for them, is is possible that some concepts would be equally similar to the concepts which in reality have different degrees of similarity with them. This problem can be overcome by enlarging the knowledge base with as many properties as possible for each concept. Also, by assigning relevance to certain properties, the more important features would be taken into account.

The evaluation of the approach on different datasets is being carried out and would be published elsewhere.

Bibliography

Allemang, D. and Hendler, J. (2008). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL.* Morgan Kaufmann Publishers.

Antoniou, G. and van Harmelen, F. (2008). *A Semantic Web Primer, second edition*. The MIT Press.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Cena, F., Likavec, S., and Osborne, F. (2012). Property-based interest propagation in ontology-based user model. In *20th Conference on User Modeling, Adaptation, and Personalization, UMAP 2012*, volume 7379 of *LNCS*, pages 38–50. Springer.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT Press.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition Journal*, 5(2):199–220.

Guarino, N. and Poli, R. (1995). Editorial: The role of formal ontology in the information technology. *International Journal of Human-Computer Studies*, 43(5-6):623–624.

Hau, J., Lee, W., and Darlington, J. (2005). A semantic similarity measure for semantic web services. In *Web Service Semantics Workshop at WWW (2005*.

Leacock, C. and Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

Lieto, A., Minieri, A., Piana, A., and Radicioni, D. P. (2015). A knowledge-based system for prototypical reasoning. *Connection Science*, 27(2):137–152.

Likavec, S. (2013). Shapes as property restrictions and property-based similarity. In Kutz, O., Bhatt, M., Borgo, S., and Santos, P., editors, *2nd Interdisciplinary Workshop The Shape of Things*, volume 1007 of *CEUR Workshop Proceedings*, pages 95–105. CEUR-WS.org.

Lin, D. (1998). An information-theoretic definition of similarity. In *15th International Conference on Machine Learning ICML '98*, pages 296–304. Morgan Kaufmann Publishers Inc.

Pirrò, G. and Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. In *9th International Semantic Web Conference, ISWC '10*, volume 6496 of *LNCS*, pages 615–630. Springer.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Trans. on Systems Management and Cybernetics*, 19(1):17–30.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Smyth, B. (2007). Case-based recommendation. In *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *LNCS*, pages 342–376. Springer.

Speer, R. and Havasi, C. (2013). The peoples web meets nlp. In Gurevych, I. and Kim, J., editors, *ConceptNet 5: A large semantic network for relational knowledge*, pages 161–176. Springer Berlin Heidelberg.

Do the Self-knowing Machines Dream of Knowing their Factivity?

Alessandro Aldini[1], Vincenzo Fano[1], and Pierluigi Graziani[2]

[1] Università di Urbino “Carlo Bo”, Italy
[2] Università di Chieti-Pescara, Italy

Abstract. The *Gödelian Arguments* represent the effort done to interpret Gödel's Incompleteness Theorems in order to show that minds cannot be explained in purely mechanist terms. With the purpose of proving the limits of mechanistic theses and investigate aspects of the Church-Turing Thesis, several results obtained in the formal setting of Epistemic Arithmetic (EA) reveal the relation among different properties of knowledge of machines, including self-awareness of knowledge and factivity of knowledge. We discuss the main principles behind the *Gödelian Arguments* and extend the results obtained in EA. In particular, we define a machine that, in a specific case, knows its own code and the factivity of its own knowledge, thus providing new insights for the analysis of the *Gödelian Arguments*.

1 Introduction

In 1951 Gödel held one of the prestigious Gibbs Lectures for the American Mathematical Society. The title of his lecture was *Some basic theorems on the foundations of mathematics and their implications* [7]. The theorems in question were precisely those of Incompleteness and the philosophical implications were concerned with the nature of mathematics and the abilities of the human mind [3]. This was one of the few official occasions in which Gödel expounded his opinion on the philosophical implications of his theorems. Without going into details about Gödel's paper, what is interesting here is the first part, where he derives the thesis of essential incompleteness of mathematics from his famous theorems. Such a thesis was sanctioned by the second theorem. Gödel's idea is that if one perceives with absolute certainty that a certain formal system [4] is correct (sound), s/he will also know the consistency of the system, that is, s/he will know the truth of the statement establishing the consistency of the system itself. But, by Gödel's second theorem, the formal system considered cannot prove its own assertion of consistency, therefore the system does not capture all arithmetical truths, and for this reason “if one makes such a statement he contradicts himself” [7, p. 309].

[3] A very accurate analysis of this work is proposed by Feferman [6], Tieszen [16], and van Atten [17].

[4] In this paper, the expression “formal system” indicates a system that is adequate to derive Incompleteness Theorems.

But what does all of this mean? Does it mean perhaps that a well defined system of correct (sound) axioms cannot contain everything that is strictly mathematical?

In the following, we first recall and discuss Gödel believes about the possible answers to such a question and then analyze the so called *Gödelian Arguments*. In the last decades many scholars dealt with these arguments, which represent the effort done to interpret Gödel's Incompleteness Theorems with the purpose of showing that minds cannot be explained in purely mechanist terms. Among them, we concentrate on the approach followed by Reinhardt [13], Carlson [3], and Alexander [1], who demonstrated a series of results in the formal setting of *Epistemic Arithmetic*, which encompasses some typically informal aspects of the Gödelian Arguments about the knowledge that can be acquired by (knowing) machines. These results emphasize several relations among different properties characterizing the expressiveness of machines, including self-awareness of knowledge and factivity of knowledge. As a contribution of this paper, we integrate these results with novel insights, thus providing the formal base for additional elements supporting the Gödelian Arguments [5].

2 Gödel Perspective

With reference to the previous question, Gödel believes that it has two possible answers:

> It does, if by mathematics proper is understood the system of all true mathematical propositions; it does not, however if someone understands by it the system of all demonstrable mathematical propositions. [...] Evidently no well-defined system of correct axioms can comprise all [of] objective mathematics, since the proposition which states the consistency of the system is true, but not demonstrable in the system. However, as to subjective mathematics it is not precluded that there should exist a finite rule producing all its evident axioms. However, if such a rule exists, we with our human understanding could certainly never know it to be such, that is, we could never know with mathematical certainty that all the propositions it produces are correct; or in other terms, we could perceive to be true only one proposition after the other, for any finite number of them. The assertion, however, that they are all true could at most be known with empirical certainty, on the basis of a sufficient number of instances or by other inductive inferences. If it were so, this would mean that the human mind (in the realm of pure mathematics) is equivalent to a finite machine that, however, is unable to understand completely its own functioning. This inability [of man] to understand himself would then wrongly appear to him as its [(the minds)] boundlessness or inexhaustibility [7, pp. 309-310].

Therefore, not only does the previous question pose the problem of the inexhaustibility or incompleteness of mathematics considered as the totality of all

[5] Although there are some very interesting connections between Gödel's Theorems and contemporary research on deep learning, we do not analyze them in this contribute. On this issue you can see [15].

true mathematical propositions; but it also raises the question as to whether mathematics is in principle inexhaustible for the human mind, that is to say, whether the human minds demonstrative abilities are extensionally equivalent to a certain formal system, or to the Turing Machine (TM) connected to it (the TM that enumerates the set of theorems of the corresponding formal system). The question, then, requires due consideration precisely of the relation between what Gödel calls *objective* and *subjective mathematics*.

First, let T be the set of mathematical truths expressible within first-order arithmetic, and call this *objective arithmetic*, or, following Gödel, "objective mathematics", that is "the body of those mathematical propositions which hold in an absolute sense, without any further hypothesis" [7, p. 305]. By Tarski's theorem, T is not definable within the language of arithmetic, hence T is not recursively enumerable. Let us then define K as the set of arithmetical statements that a human being can know and prove absolutely and with mathematical certainty, that is what one can derive [6] and know to be true. Let us call it *subjective arithmetic* or, following Gödel, "subjective mathematics", which "consists of all those theorems whose truth is demonstrable in some well-defined system of axioms all of whose axioms are recognized to be objective truths and whose rules preserve objective truth" [6, p. 135-136]. What is then the relation between K and T? Quoting Feferman, we could synthesize Gödel's answer by saying that if K was equal to T:

> then demonstrations in subjective mathematics [would not be] confined to any one system of axioms and rules, though each piece of mathematics is justified by some such system. If they do not, then there are objective truths that can never be humanly demonstrated, and those constitute absolutely unsolvable problems [6, p. 136-137].

That is, if the equivalence K=T held, the human mind would not be equivalent to any formal system or TM connected to it. In fact, having established characteristics of T, for each formal system there would be a provable statement by the human mind, but not within the formal system. Hence, the mechanistic thesis would certainly be false: T non-recursive enumerability entails, in fact, the non-existence of any effective deductive system whose theorems are only and all truths of arithmetic. If, on the contrary, K did not coincide with T, and thus the human mind were equivalent to a given formal system or to the TM related to it, the existence of arithmetical statements humanly undecidable in an absolute sense would follow. In fact, as underlined by Gödel, the second incompleteness theorem does allow this conclusion: the proposition expressing the consistency of K, say ConK, is true but is not provable within the system itself; the negation of ConK is false and is not provable in K. Having established the equivalence between the human mind and a formal system, ConK is not even provable by the human mind. Finally, since ConK can be put in the form of a Diophantine

[6] As Feferman [6, p. 140] emphasizes, Gödel believes that "the human mind, in demonstrating mathematical truths, only makes use of evidently true axioms and evidently truth preserving rules of inference at each stage".

problem, it is an absolutely undecidable problem. Such a proposition is, thus, an unknowable truth. These arguments lead Gödel to the idea that from the incompleteness results one can at the most derive the following disjunction:

> Either [subjective] mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule, that is to say, the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable diophantine problems of the type specified (where the case that both terms of the disjunction are true is not excluded, so that there are, strictly speaking, three alternatives) [7, p. 310].

So, following Tieszen [16], and considering the translatability between the concept of a well defined formal system and that of a TM, we can say that Gödel's Incompleteness Theorems show that it could not be true that: (i) the human mind is a finite machine (a TM) and there are for it no absolutely undecidable Diophantine problems.

> The incompleteness theorems show that if we think of the human mind as a TM then there is for each TM some absolutely undecidable Diophantine problem. The denial of the conjunction (i) is, in so many words, Gödel's disjunction. In formulating the negation of (i) Gödel says that the human mind infinitely surpasses the powers of any finite machine. One reason for using such language, I suppose, is that there are denumerably many different Turing machines and for each of them there is some absolutely diphantine problem of the type Gödel mentions. So Gödel's disjunction, understood in this manner, is presumably a mathematically established fact. It is not possible to reject both disjuncts. [16, pp. 230-231].

The disjunction leaves open the three following possibilities:

(I) human intelligence infinitely surpasses the powers of the finite machine (TM), and there are no absolutely unsolvable Diophantine problems (see [7, p. 310]).

(II) human intelligence infinitely surpasses the powers of the finite machine (TM) and there are absolutely unsolvable Diophantine problems. That is, although human intelligence is not a finite machine, nevertheless there are absolutely irresolvable Diophantine problems for it.

(III) human intelligence is representable through a finite machine (TM) and there are absolutely irresolvable Diophantine problems for it.

Gödel was convinced that (I) held, but he was also aware that his incompleteness theorems did not make the existence of a mechanic procedure equivalent to human mind impossible. Gödel, however believed that from his theorems it followed that if a similar procedure existed we "with our human understanding could certainly never know it to be such, that is, we could never know with mathematical certainty that all the propositions it produces are correct". This exactly means that "the human mind (in the realm of pure mathematics) is equivalent to a finite machine that, however, is unable to understand completely its own functioning". In 1972 Gödel expressed further on the matter saying [18]:

> On the other hand, on the basis of what has been proved so far, it remains possible that there may exist (and even be empirically discoverable) a theorem-proving machine which in fact is equivalent to mathematical intuition, but cannot be *proved* to be so, nor even be proved to yield only *correct* theorems of finitary number theory.

This formulation is significantly different from that of 1951, as now Gödel appears to recognize that the mind, at least in his doing mathematics, could be a machine and we could not recognize this fact or not be able to prove it.

3 Knowing Machines

After the speculative ideas formulated by anti-mechanists, like the famous argument by Lucas [8, 9], several authors, like Benacerraf [2], Penrose [10–12], Chihara [4], and Shapiro [14] (see [5] for a comprehensive survey), proposed more formal lines of reasoning on the implications of Gödel's Theorems. Here, we consider the results by Reinhardt [13], Carlson [3], and Alexander [1], who analyzed a formal theory, called *Epistemic Arithmetic* (EA), encompassing some typically informal aspects of the Gödelian Arguments about the knowledge that can be acquired by (knowing) machines. EA is the language of Peano Arithmetic enriched with a modal operator K for *knowledge* (or for *intuitive provability*). The formal interpretation of K passes through the definition of the properties at the base of an epistemic notion of *knowability*:

- Logic Consequence: if ϕ and $\phi \to \psi$ are known, then ψ is known.
- Infallibilism: what is known is also true.
- Introspection: if ϕ is known then such a knowledge is known.

The basic axioms of knowledge are:

B1. $K\forall x\phi \to \forall x K\phi$
B2. $K(\phi \to \psi) \to K\phi \to K\psi$
B3. $K\phi \to \phi$
B4. $K\phi \to KK\phi$

where $B2$-$B4$ formalize the intuitions above and are stricly related to, e.g., the modal system $S4$, while $B1$ expresses a first-order condition stating that the assertion "ϕ is known to be valid" implies the knowledge of each element that can be assigned to x in ϕ and the truth of the formula under each such assignment. [7] Assumed that the K-closure of ϕ is the universal closure of ϕ possibly prefixed by K, the axioms of EA are the K-closure of $B1$-$B4$ and of the axioms of Peano Arithmetic. The theory of knowledge defined in such a way extends conservatively the classical interpretation of Peano Arithmetic.

[7] We are assuming that ϕ is a formula with one free variable x.

Under this theory of knowledge, variants of Church-Turing Thesis are investigated to analyze the relationship between properties that are *weakly K-decidable* [8] and the TMs that formalize the decision algorithm for these properties. In the following, we assume that W_e is the recursively enumerable set with Gödel number e.

Theorem 1 (Reinhardt's schema [13]). *$\exists eK\forall x(K\phi \leftrightarrow x \in W_e)$ is not consistent in* EA[9].

Informally, Reinhardt's schema states that a TM exists for which *it is known* that it enumerates all (and only) the elements (for which *it is known*) that make ϕ true. More precisely, as the assignments making ϕ true are a known recursively enumerable set, we then derive the computability, through a known TM, of the (weak K-) decision problem for ϕ. Following Carlson, the intuitive interpretation is: *I am a TM and I know which one.* A weaker version of Reinhardt's schema is conjectured by Reinhardt himself and proved by Carlson, in which the outermost K operator prefixes the statement.

Theorem 2 (Carlson's schema [3]). *$K\exists e\forall x(K\phi \leftrightarrow x \in W_e)$ is consistent in* EA.

Quoting Carlson, *I know that the set of x for which I know $\phi(x)$ is recursively enumerable*, or, by rephrasing an analogous hypothesis studied by Benacerraf independently [2], *I know I am a TM but I do not know which one.* Carlson uses the term *knowing machine* to denote any recursively enumerable proof system that represents a model for the theory of knowledge, and shows that, indeed, EA integrated with his schema is a knowing machine. As a corollary of this result, the schema obtained by removing the outermost K operator is still consistent in EA.

The proofs of the results above rely on the validity of $K(K\phi \to \phi)$, stating that in the formal system the *factivity* of knowledge is known. In between these two limiting results, Alexander has recently proved a dichotomy: a machine can know its own factivity as well as that it has some code (without knowing which, as stated by Carlson's schema), or it can know its own code exactly (proving the consistency of Reinhardt's schema) but cannot know its own factivity (despite actually being factive). Providing that the axioms of EA *mod factivity* consist of the axioms of EA except for the universal closure of $B3$ prefixed by K (that represents knowledge of factivity of knowledge), it is possible to prove that:

Theorem 3 (Alexander [1]). *Reinhardt's schema is consistent in* EA *mod factivity.*

and then to construct the previous dichotomy.

In this setting, we show a result related to a specific case. An interpreter f_u is a function mimicking the behavior of any other function. Formally, $f_u(x, y) = f_x(y)$. For instance, the universal TM is an interpreter. Interpreters represent a

[8] The assignments of x satisfying ϕ are known.

[9] The inconsistency of this schema is proved as a consequence of first Gödel's theorem.

classical tool in computability theory and play a fundamental role for programming languages. Now, let us consider Reinhardt's schema in EA *mod factivity* and $\phi(x) := (f_x(x) = 1)$. Then, from:

$$\exists e K \forall x (K\phi \leftrightarrow x \in W_e)$$

by taking $x = e$ we derive:

$$\exists e K (K\phi(e) \leftrightarrow e \in W_e) \tag{1}$$

and:

$$K(K\phi(e) \rightarrow \phi(e)) \tag{2}$$

which expresses a limited form of knowledge of factivity that is allowed in EA *mod factivity*. More precisely, we have a machine that, for (at least) a specific choice of the function ϕ and of the input x, i.e., the interpreter function and the Gödel number of the machine itself, knows its own code and its own factivity. We have to note that taking $x = e$ roughly speaking means that *if I allow the machine to know its own identity, then of course it will possess this knowledge.* Attributing this capacity to a machine is very natural for us and in our opinion it shows that Alexander's framework is adequate to analyze the machines' knowledge [10]. By virtue of such a choice, the intuition that we stem is that the machine knows its own code and is aware of the factivity of the knowledge resulting by interpreting its own code, while such an awareness, according to the dichotomy above, is lost when interpreting other inputs. As a consequence, by rephrasing Carlson and Benacerraf intuitions, we could say: *If I know which universal TM I am, then I know the factivity of my knowledge.*[11] Hence, to some extent, self-reference increases the expressiveness of knowledge, provided that the machine is an interpreter. In our opinion, this is an interesting enhancement of the tradeoff result provided by Alexander that can represent an additional formal element for the analysis of the Gödelian Arguments.

References

1. S. Alexander. A machine that knows its own code. *Studia Logica*, 102:567–576, 2014.
2. P. Benacerraf. God, the Devil and Gödel. *The Monist, 51*, pages 9–32, 1967.
3. T.J. Carlson. Knowledge, machines, and the consistency of Reinhardt's strong mechanistic thesis. *Annals of Pure and Applied Logic*, 105:51–82, 2000.
4. C.S. Chihara. On alleged refutations of mechanism using Gödel's incompleteness results. *The Journal of Philosophy*, 69:507–526, 1971.
5. V. Fano and P. Graziani. Mechanical Intelligence and Gödelian arguments. In E. Agazzi, editor, *The Legacy of A.M. Turing*, pages 48–71. Franco Angeli, 2013.

[10] Note also that e could have a very high complexity and this fact is compatible with the incapability of humans to understand how the most advanced algorithms produce their results.

[11] Roughly speaking: *If I know which universal TM I am, then I know my factivity.*

6. S. Feferman. Are There Absolutely Unsolvable Problems? Gödel's Dichotomy. *Philosophia Mathematica*, (III) 14:134–152, 2006.
7. K. Gödel. Some basic theorems on the foundations of mathematics and their implications. In K. Gödel, editor, *Collected Works*, volume III, pages 304–335. Oxford University Press, 1995.
8. J.R. Lucas. Minds, Machine and Gödel. *Philosophy*, 36:112–127, 1961.
9. J.R. Lucas. Satan stultified: a rejoinder to Paul Benacerraf. *The Monist*, 52:145–158, 1968.
10. R. Penrose. *The emperor's new mind*. Oxford Univ. Press, Oxford, 1989.
11. R. Penrose. *Shadows of the mind*. Oxford University Press, Oxford, 1994.
12. R. Penrose. Beyond the doubting shadow. *Psyche*, 2-1, 1996.
13. W. Reinhardt. Epistemic theories and the interpretation of Gödel's incompleteness theorems. *Journal of Philosophical Logic*, 15:427–474, 1986.
14. S. Shapiro. Incompleteness, mechanism, and optimism. *The Bulletin of Symbolic Logic*, 4:273–302, 1998.
15. B. R. Steunebrink and J. Schmidhuber. Towards an Actual Gödel Machine Implementation: a Lesson in Self-Reflective Systems. *Theoretical Foundations of Artificial General Intelligence*, 4:173–195, September 2012.
16. R. Tieszen. After Gödel: Mechanism, Reason, and Realism in the Philosophy of Mathematics. *Philosophia Mathematica*, (III) 14:229–254, 2006.
17. M. van Atten. Two Draft Letters from Gödel on Self-knowledge of Reason. *Philosophia Mathematica*, 14:255–261, 2006.
18. H. Wang. *From Mathematics to Philosophy*. Humanities Press, N.Y., 1974.

Extracting Concrete Entities through Spatial Relations

Olga Acosta and César Aguilar

Facultad de Letras, Pontificia Universidad Católica de Chile

{oacostal,caguilara}@uc.cl

Abstract. This paper focuses on the automated extraction of concrete entities from a specialized-domain corpus. Then, in a bootstrapping phase, the candidates are used to extract new candidates. Concrete entities are automatically identified by a set of spatial features. In a spatial scene something is located by virtue of the spatial properties associated with a reference object. The axial properties are represented by place adverbs. Additionally, for identifying referent objects in a sentence we consider syntactical patterns extracted by chunking. In order to reduce noise in results, we take into account a corpus comparison approach and linguist heuristics. Results show high precision in candidates with high weights.

Keywords: Concrete entities, lexical relation, information extraction, term extraction, axial properties, nominalization.

1. Introduction

In recent years, the automatic mining of relevant knowledge in the biomedical domain has become in an interesting research area, particularly in tasks related to the generation of taxonomies and ontologies (Smith and Kumar, 2004). This kind of tasks require the design and implementation of efficient information extraction (IE) methods, capable of identifying and extracting textual patterns that contain such relevant knowledge.

Therefore, in this work we propose a methodology for the automatic extraction of concrete entities implicit in medical documents. Then, in a bootstrapping phase, these candidates are used for extracting a larger set of new candidates.

Linguistically speaking, a main concern is those noun phrases (NP) whose modifiers are relational adjectives and where the noun head is a concrete entity, because relational adjectives introduce semantic features which describe specific properties such as formal, constitutive, telic and agentive qualities (Fábregas, 2007). The identification of this type of NP contributes to delimit the number of possible semantic relations. For testing our method, we work with a corpus of medical texts in Spanish.

We organize our paper as follows: in section 2 we define what a concrete entity is, taking into account the description proposed by Fellbaum (1998) for classifying names in WordNet. Then, in section 3, we show a brief explanation about the representation of space in natural language, according to a cognitive framework. In section 4, we describe the most common deverbal nominalizations in specialized texts. In section 5 we explain the relation noun + relational adjective in order to delineate a set of linguistic heuristics useful for filtering non-relevant adjectives. In section 6 we describe our methodology. In section 7 we offer a description of preliminary results. Finally, in section 8, we give our conclusions.

2. Concrete entities

We understand all that exists in the world as a concrete entity which something can be predicated (in Aristotle's categories: substance). For example, concrete entities can be *artifactual* categories like *vehicles*, *clothing* and *weapons*, or *natural* kinds like *birds*, *fruits* and *vegetables* (Landau and Jackendoff, 1993; Murphy, 2002). This is in line with 8 of the 25 main categories considered in the WordNet hierarchy for nouns denoting tangible things: {*animal*, *fauna*}, {*artifact*}, {*body*}, {*food*}, {*natural object*}, {*person*, *human being*}, {*plant*, *flora*}, {*substance*}. From our point of view these categories can be collapsed in artifactual and natural kinds.

3. Space in language and cognition

Levinson (2004) points out that the spatial thinking is a crucial feature in our lives: we constantly consult our spatial memories in events such as finding our way across town, giving route directions, searching for lost keys, and so on. This importance is mirrored in real discourse where knowledge about formal, agentive, constitutive and telic features, as well as spatial features, are found in specialized domains.

There are three frames of reference lexicalized in language: intrinsic, relative and absolute frame. *Intrinsic frame* involves an object-centred coordinate system, where the coordinates are determined by the "inherent features", sidedness or facets of the objet to be used as the ground (i.e., *he's in front of the house*). *Relative frame* of reference presupposes a viewpoint where a perceiver is located, a figure and ground distinct from the viewpoint. Thus, it offers a triangulation of three points, and utilizes coordinates fixed on viewpoint to assign directions to figure and ground (i.e., *the ball is to the left of the tree*). Finally, *absolute frame* refers to the fixed direction provided by gravity (i.e., *he's north of the house*).

3.1. Work related

Mani *et al*. (2010) focused on the problem of extracting information about places, considering both absolute and relative references. Their goal was on grounding such references to precise positions that can be characterized in terms of geo-coordinates. These authors use a supervised approach to mark up PLACE tags in documents. SpatialML is an annotation scheme derived from this work and which has been applied to annotated corpora in English and Mandarin Chinese. An automatic tagger for SpatialML extents scores 86.9 F-measure, which is a reasonable performance. On the other hand, Clementini et al. (1997) propose a unified framework for the qualitative representation of positional information in a two-dimensional space in order to perform spatial reasoning. The orientation and distance relations for objects modeled as points can determine positional information. The implicit characteristics of an object are its topology and its extension, while, with respect to other objects, topological, orientation, and distance relations have to be considered.

3.2. Axial properties

Evans (2007) explains that a spatial scene is a linguistic unit containing information based on our spatial experience. This space is structured according to four parameters: a figure (or *trajector*), a referent object (that is, a *landmark*), a region and —in certain cases— a secondary reference object. These two reference objects configure a reference frame. We can understand this configuration by considering the following example: *un auto está estacionado detrás de la escuela* (Eng.: "a car is parked behind the school"). In this sentence, *un auto* is the figure and *la escuela* is the referent object. The region is established by the combination of the adverb *detrás*[1] which sketches a spatial relation with the referent object. This relation encodes the location of the figure.

Moreover, Evans (2007) points out the existence of axial properties, that is, a set of spatial features associated to a specific referent object. Considering again the sentence *a car is parked near to the school*, we can identify the location of the car searching for it in the region near to the school. Therefore, this search can be performed because the referent object (the school) has a set of axial divisions: front, back and *side* areas.

3.3. Axial properties and place adverbs

Axial properties are linguistically represented by place adverbs. In this experiment we only consider adverbs functioning in Spanish with preposition *de* (Acosta and Aguilar, 2015):

> *Enfrente*, *delante* (Engl. *In front to/of*); *Detrás*, *atrás* (Engl. *Behind*); *sobre*, *encima* (Engl. *On*); *abajo*, *debajo* (Engl. *under*); *dentro*, *adentro* (Engl. *In/inside*); *fuera*, *afuera* (Engl. *Out/outside*); *arriba* (Engl. *Above/over*).

Additionally, we use some synonymous nouns such as exterior (outside) and interior (in), as well as *side* nouns synonymous with the dimensions left and right.

4. Nominalization

According to Martin (1993: 203-220) and Vivanco (2006), from a linguistic perspective, the discourse neutrality in science and technology is presented by means of im-

[1] In English, *behind* is a preposition. In contrast, in Spanish is an adverb.

personation: missing second person, low presence of first person, abundance of impersonal verbs and passive voice, as well as nominalizations hiding actions made by the subject. These nominalizations are used by scientists to support their arguments, coining new terms by means of nouns and summarizing information previously provided in a text.

In line with the frequent use of nominalization in specialized texts, in the case of Spanish, Cademártori, Parodi and Venegas (2006) show data concerning the use of deverbal nominalizations in three domains: commercial, maritime and industrial. The most used suffixes for constructing nouns are: *-ción*, *-miento*, *-sión*, and *-dor*.

5. Adjectives-Noun modifiers

An adjective is a grammatical category whose function is to modify nouns (Demonte, 1999). There are two kinds of adjectives: descriptive and relational adjectives. The descriptive adjectives refer to constitutive features of the modified noun characterized by means of a single physical property: color, form, character, predisposition, sound, and so on, e.g., *el libro azul* (Eng.: "the blue book"). On the other hand, relational adjectives assign a set of properties, i.e., all the characteristics jointly defining names as *sea*: *puerto marítimo* (Eng.: "maritime port"). In terminology, relational adjectives represent an important element for building specialized terms. For example, *inguinal hernia*, *venereal disease* and others are considered terms in medicine as opposed to NPs with more contextual interpretations like *rare hernia*, *serious disease*, and *critical disorder*.

5.1. Identifying syntactically non-relevant adjectives

If we consider the internal structure of adjectives, we can identify two types: permanent and episodic adjectives (Demonte, 1999). The first kind of adjectives represents stable situations, permanent properties characterizing individuals. These adjectives are located outside of any spatial or temporal restriction (i.e., *psicópata*/"psychopath"). On the other hand, episodic adjectives refer to transient situations or properties implying change and with time-space limitations.

Almost all descriptive adjectives derived of participles belong to this latter class as well all adjectival participles (i.e., *harto*/"jaded"). Spanish is one of the few languages that in its syntax represent this difference in the meaning of adjectives. In many languages this difference is only recognizable through interpretation. In Spanish, individual properties can be predicated with the verb *ser*, and episodic properties with the verb *estar*, which is an essential test to recognize what class an adjective belongs to. In this sense, with the goal of identifying and extracting non-relevant adjectives, we propose extracting adjectives predicated with the verb *estar* (Acosta, Aguilar and Sierra, 2013).

Another linguistic heuristic for identifying descriptive adjectives is that only these kinds of adjectives accept degree adverbs or are part of comparative constructions, e.g., *muy alto*/"very high", *Juan es más alto que Pedro*/"John is taller than Peter".

Finally, only descriptive adjectives can precede a noun because —in Spanish— relational adjectives are always postposed (e.g., *la antigua casa*/“the old house”).

5.2. Types of relational adjectives

According to Bosque (1993) relational adjectives such as *salivary* in the noun phrase *salivary gland* belong to a kind of relational adjectives which do not occupy positions in the argument structure of the predicate, but they denote entities which establish a specific relation with the head noun. Bosque refers to these relational adjectives as *classification relational adjectives*, while the term *thematic relational adjectives* is left for the other group, e.g., the case of *renal infection*, where *infection* is derived from a verb.

6. Methodology

In this paper we propose a methodology for extracting concrete entities from a specialized domain corpus with part-of-speech tags.

6.1. Part-of-Speech Tagging

Part-of-Speech (POS) tagging is the process of assigning a grammatical category to each word in a corpus. The most common taggers used for Spanish are *TreeTagger* (Schmid, 1994) and *FreeLing*[2] (Carreras et al., 2004). In this experiment, we use FreeLing because it is more precise than TreeTagger for tagging texts in Spanish. The following example shows a sentence in Spanish tagged with the FreeLing tagger:

> el/DA tipo/NC más/RG común/AQ de/SP lesión/NC ocurrir/VM cuando/CS algo/PI irritar/VM el/DA superficie/NC externo/AQ del/PDEL ojo/NC

6.2. Chunking

Chunking is the process of identifying and classifying segments of a sentence by grouping the major parts-of-speech that form basic non-recursive phrases.

In this work, we concern the automated extraction of concrete entities. Concrete entities relevant to a domain are terms and the most productive patterns of terms consist of a noun and zero or more adjectives (Vivaldi, 2001). Using FreeLing tags, these patterns can be represented as a regular expression in a single pattern:

<NC><AQ>*

The above regular expression is considered in the first phase of extraction of candidates.

---

[2] FreeLing based on the tags of the EAGLES group.

Concrete entities can be located in spatial scenes as figures or reference objects. In this experiment, only reference objects are extracted with their axial properties that can be linguistically represented as:

<RG|NC><PDEL><DA>?<NC><AQ>*

The regular expressions used to extract non-relevant adjectives according to the linguistic heuristics mentioned in section 5.1 are:

<RG><AQ>
<VAE><AQ>
< D.*|P.*|F.* |S.*><AQ><NOUN>

Where RG, AQ and VAE as tagged with FreeLing, correspond to adverbs, adjectives and the verb *estar*, respectively. Tags *<D.*|P.*|F.*|S.*>* correspond to determinants, pronouns, punctuation signs and prepositions. The expression *<D.*|P.*|F.*|S.*>* is a restriction to reduce noise, since elements wrongly tagged by FreeLing as adjectives are extracted without this restriction.

6.3. Bootstrapping phase

We use the candidates to concrete entities obtained in the first step as seeds for extracting more candidates. On the one hand, we assume that coordinating phrases where a good candidate occurs have a high probability of containing other good candidates for a concrete entity:

<NC><AQ>*<CC><NC><AQ>*

Where *<CC>* tag corresponds to the disjunction (i.e.: *kidney **or** liver*) and conjunction (i.e.: *kidney **and** liver*).

On the other hand, noun phrases with at least an adjective take advantage of the noun head of candidates for a concrete entity for finding more specific candidates (i.e., artery-femoral artery):

<NC><AQ>+

6.4. Reducing noise

We sought to remove non-relevant words from noun phrases before ranking candidates for concrete entities. After the chunking phase, noise was reduced by removing non-relevant open-class words. One of our goals consists of building this stopword list as automatically as possible.

Since concrete entities are terms in the domain, a list of non-relevant words from the domain (i.e., stopword list) can be used to refine the terminology obtained from an automatic process. We considered a list constructed with high frequency words in a reference corpus to have drawbacks because, apart from the selection by occurrence frequency (in the domain corpus, words with high frequency can be terms), human supervision is required in order to determine whether a word is relevant to the domain.

Given the above, we consider that linguistic heuristics operating in a specific language can be taken into account in order to automate the selection of non-relevant words. One of the disadvantages, however, is that this leads to language dependence. For the case of adjectives, in Spanish, characteristic features have been proposed in order to distinguish between descriptive and relational adjectives as mentioned in section 5. On the other hand, with a corpus comparison approach, we obtain both nouns and adjectives where the relative frequency in a reference corpus is greater or equal than in the domain corpus. These words can be used as part of the stopword list. Additionally, we take into account empirical evidence concerning the use of deverbal nominalizations in specialized discourse (Cadermártori, Parodi and Venegas, 2006) for removing phrases where noun heads are indicative of actions, events and states but not concrete entities (in a NP with a noun head of this type, a thematic relational adjective is found). In this sense, suffixes as *–ción, -miento*, and *–sión* were used for filtering out noun phrases. Finally, a short list with the more frequent non-relevant nouns operating as noun heads in phrases: *form, type, kind, cause, effect* and so on, were considered for removing noun phrases.

Adjectives from the reference corpus can be used as a fixed-size list where non-relevant adjectives automatically extracted from the domain can be added. These can be obtained taking into account the three heuristics mentioned in section 5.1. Then, these adjectives can be manually reviewed in order to determine their relevance to any specialized knowledge domain (i.e., adjectives as relevant, important, necessary, appropriate, and so on can be considered for the stopword list). This is a fixed-size list and can be the base-list where non-relevant adjectives automatically extracted from the domain can be added.

6.5. Ranking words

We evaluate termhood of simple words by means of rank difference (Kit and Liu, 2008) between two different corpora as in the formula (1). Given the syntactical pattern used for terms in this study, we take into account only nouns and adjectives in both corpora because they are the kind of words most used for building terms:

(1)

Where f_{dom} and N_{dom} correspond to the absolute occurrence frequency of w_i and the size of the domain corpus, respectively. Similarly, f_{ref} and N_{ref} correspond to absolute occurrence frequency of w_i and the size of the reference corpus.

Kit and Liu (2008) only focus on extracting single-word term candidates, so they only weigh words occurring in both the domain and the general corpus. In our experiment we also consider words that only occur in the domain corpus. We assumed that the reference corpus is large enough to filter out non-relevant words, hence words only occurring in the domain corpus have a higher probability of being relevant and the word's frequency reflects its importance:

(2)

We consider that the larger the reference corpus, the higher the *exhaustivity*[3] of open class words of general usage, as well as a higher probability that specialty terms occur at least one time (the reference corpus was collected from an online newspaper where news about science and technology are published too), so that we would expect a higher precision in ranking.

6.6. Ranking multi-word term candidates

Formally, if a candidate noun phrase (np) has a length of n words, $w_1 w_2 \ldots w_n$, where $n>1$, then the ranking of the candidate *np* is the sum of the frequency of *np* as a whole plus the weights of all the individual words w_i:

(3)

7. Results

This section presents the results of our experiment considering a subset of 1,200,000 tokens of the MedLineplus corpus.

7.1. Sources of textual information

Domain corpus
The source of textual information is constituted by a set of documents of the medical domain, basically human body diseases and related topics (surgeries, treatments, and so on). These documents were collected from MedlinePlus in Spanish.

The size of the corpus is 1.2 million tokens, but we carried out our experiment with a subset of 200,000 words in order to determine manually the number of concrete entities present in the results. As an ongoing work, we are manually determining how many concrete entities are present in the complete corpus. We chose a medical domain due to the availability of textual resources in digital format. Finally, we assume that the choice of domain does not suppose a very strong constraint for generalizing the results to other domains.

Reference corpus
With the goal of ranking words relevant to the domain by means of their relative frequency ratio, a large reference corpus was collected from an online newspaper[4] with new articles from 2014 (the size of corpus is about 5 million tokens). URLs from the

[3] *Exhaustivity* of a document description is the coverage it provides for the main topics of the document. So, if we add new vocabulary terms to a document, the *exhaustivity* of the document description increases (Baeza and Ribeiro, 2011).

[4] www.lajornada.com.mx. Mexican newspaper with information available online.

main heads were automatically extracted using the Python library BeautifulSoup[5]. Then, this set of URLs was introduced in WebBootCat, a search tool of Sketch Engine[6], in order to automatically collect the textual information from each WEB page. The description of the structure of the reference corpus is showed in table 1.

Table 1. Structure of the reference corpus.

Category	Docs	%
Sciences	24	0.4
Politics	1865	29.3
Entertainment	98	1.5
Sports	515	8.1
Society	416	6.5
City	424	6.7
States	449	7.1
Economy	658	10.4
World	662	10.4
Culture	137	2.2
Editorial	316	5.0
Mails	318	5.0
Opinion	319	5.0
Homepage	155	2.4

7.2. Other resources

The programming language used in order to automate all tasks required was Python version 3.4 as well as the *NLTK* module version 3.0 (Bird, Klein and Loper, 2009). Additionally, the POS tagger used in this experiment was *FreeLing* which is included in *Sketch Engine*.

5 www.crummy.com/software/BeautifulSoup/bs4/doc/

6 https://the.sketchengine.co.uk

7.3. Analysis of results

The first phase of extraction of candidates to concrete entity without filters achieves a global precision of 56%. The tables 2 and 3 show precision with different thresholds of candidates starting with the better ranked candidates. With the stopword list built as mentioned in section 6.4, we achieve a global precision of 76%. Global precision with a stopword list reflects an improvement of 20%, but a significant loss of 17% of true candidates. As can be seen from these tables, the ranking of words and noun phrases is useful for sorting results from the most relevant to the least relevant results.

Table 2. Comparison of results.

Candidates	**Precision**	
	Without filter	**With filter**
100	91%	96%
200	87%	87%
300	73%	83%
400	69%	
500	63%	

Bootstrapping phase
The bootstrapping phase taking into account coordinating phrases achieves a set of 1248 candidates, of which 262 are new true candidates. The global precision with this second phase is of 47%, with a precision by thresholds as shown in table 3. The advantage of this phrase structure is that single-word candidates can be extracted.

On the other hand, the bootstrapping phase considering noun phrases achieves a set of 2796 candidates, of which 1534 are good candidates. The global precision of this phase is of 55%, with a precision by thresholds as shown is table 3. One disadvantage of this structure is that only candidates with at least one adjective can be selected.

Table 3 shows a better performance with noun phases. The identification of the concrete entities present in corpus is an ongoing task that will let us evaluate in terms of recall too.

Table 3. Bootstrapping phase.

Candidates	Coordinating phrases	Noun phrases
100	55%	71%
200	59%	71%
300	59%	69%
400	59%	68%
500+	53%	65%

7.4. Discussion

The candidates in a bootstrapping phase give us insight about the kind of semantic relations implicit in noun phrases of the type *<NC><AQ>*. Given the phase of reduction of non-relevant adjectives, we have a great deal of relational adjectives where it is possible to find different relations. For example, *salivary gland* has implicit a telic relation. On the other hand, *testicular gland* has a part-whole or locative relation. Finally*, meibomian gland* may be considered as a specific type of *gland.*

With respect to the extraction of lexical relations, specifically hyponymy-hypernymy relations (Hearst, 1992; Wilks, Slator and Guthrie, 1995; Pantel and Pennacchiotti, 2006), as well as meronymy relations (Berland and Charniak, 1999; Girju, Badulescu and Moldovan, 2006), these works are based on patterns where two terms are located in the context of a sentence: the hand has fingers, the dog is an animal, and so on, but there are few jobs working with noun phrases, which we consider it is very important because we could consider a noun phrase as salivary gland as an hyponym of gland, but it is clear that if we dig a little deeper that the semantic relation implicit is telic.

8. Conclusions

We discussed a methodology for extracting concrete entities in the medical domain. Concrete entities have been studied since Aristotle's works, particularly in his biological and zoological descriptions. According to Aristotle's categories (the first category), many things can be predicated of substances. We assume that substances are concrete entities, with a more extended meaning, i.e.: the eight tangible categories formulated by Fellbaum for WordNet (1998). Thus, we consider that the automated identification and extraction of this kind of information is an important advance in further NLP tasks.

Cognitive abilities as the spatial knowledge and his representation in natural language are important for our extraction methodology. We observe that spatial descriptions are frequent in specialized discourses. Additionally, we propose a further step of bootstrapping in order to find a great number of candidates for concrete entities. Candidates with a concrete entity as a noun head and a relational adjective show semantic relations as part-whole, locative, agentive and telic, which can be interpreted, at first, as hyponymy/hyperonymy relations.

On the other hand, to assign relevance to words is an important step for ranking candidates, according to our exposed results. In this sense, as ongoing work, we are collecting more information about science and technology at the same electronic journal in order to improve the results in the ranking process.

Finally, it is necessary to mention that POST taggers as FreeLing and TreeTagger fail in the task of identifying nouns, adjectives and verbs closely related with the domain. This failure has a negative impact on the results. We believe it is important to face this problem in future extraction tasks.

Acknowledgments

This paper has been supported by the National Commission for Scientific and Technological Research (CONICYT) of Chile, Project Numbers: 3140332 and 11130565.

9. References

1. Acosta, O., Aguilar, C. & Sierra, G. Using Relational Adjectives for Extracting Hyponyms from Medical Texts. In A. Lieto & M. Cruciani (eds.), *Proceedings of the First International Workshop on Artificial Intelligence and Cognition (AIC 2013)*, CEUR Workshop Proceedings, pp. 33-44.Torino, Italy. (2013).
2. Acosta, O. & Aguilar, C. Extraction of Concrete Entities and Part-Whole Relations. In B. Sharp & R. Delmonte (eds.), *Natural Language Processing and Cognitive Science. Proceedings 2014*, pp. 89-100. Berlin, De Gruyter (2015).
3. Baeza, R. & Riveira, B. *Modern Information Retrieval*, 2nd ed. New York, Addison Wesley (2011).
4. Berland, M. & Charniak, E. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 57-64. College Park, Maryland, USA, ACL Publications (1999).
5. Bird, S., Klein, E. & Loper, E. *Natural Language Processing with Python*, Sebastropol, Cal., O'Reilly (2009).
6. Bosque, I. Sobre las diferencias entre los adjetivos relacionales y los calificativos. *Revista Argentina de Lingüística*, No. 9, pp. 10-48 (1993).
7. Carreras, X. Chao, I., Padró, L. & Padró, M. FreeLing: An Open-Source Suite of Language Analyzers. In M.T. Lino *et al.* (eds.) *Proceedings of the 4th International Conference on Language Resources and Evaluation LREC 2004*, pp. 239-242. Lisbon, Portugal, ELRA Publications (2004).
8. Cademártori, Y., Parodi, G. & Venegas, R. El discurso escrito y especializado: caracterización y funciones de las nominalizaciones en los manuales técnicos, *Literatura y Lingüística*, No. 17, pp. 243-265 (2006).
9. Chunyu, K. & Liu, X. Measuring mono-word termhood by rank difference via corpus comparison. *Terminology*, 14(2), 204-229 (2008).

10.Clementini, E., Di Felice, P., & Hernández, D. Qualitative representation of positional information. *Artificial intelligence*, *95*(2), 317-356 (1997).

11.Demonte, V. El adjetivo. Clases y usos. La posición del adjetivo en el sintagma nominal. In I. Bosque & V. Demonte (eds.), *Gramática descriptiva de la lengua española*, Vol. 1, Cap. 3, pp. 129-215. Madrid, Espasa-Calpe (1999).

12.Evans, V. *A Glossary of Cognitive Linguistics*, Edinburgh, UK, Edinburgh University Press (2007).

13.Fábregas, A. The internal syntactic structure of relational adjectives, *Probus*, 19(1), 1-36 (2007).

14.Fellbaum, C. *WordNet: An Electronic Lexical Database*, Cambridge, Mass., MIT Press (1998).

15.Girju, R., Badulescu, A. & Moldovan, D. Automatic discovery of part–whole relations. *Computational Linguistics*, 32(1), 83-135 (2006).

16.Hearst, M. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pp. 539-545, Nantes, France. ACL Publications (1992).

17.Landau, B. & Jackendoff, R. What and where in spatial language and spatial cognition, *Behavioral and brain sciences*, 16(02), 255-265 (1993).

18.Levinson, S. *Space in Language and Cognition: Explorations in Cognitive Diversity*, Cambridge, UK, Cambridge University Press (2004).

19.Mani, I., Doran, C., Harris, D., Hitzeman, J., Quimby, R., Richer, J. & Clancy, S. SpatialML: annotation scheme, resources, and evaluation. *Language Resources and Evaluation*, 44(3), 263-280 (2010).

20.Martin, James R. Technicality and abstraction: Language for the creation of specialized texts. In M.A.K. Halliday & James R. Martin. *Writing science: Literacy and discursive power*, pp. 203-220, London, The Falmer Press (1993).

21.Murphy, G. *The Big Book of Concepts*. Cambridge, Mass., MIT Press (2002).

22.Pustejovsky. J. *The generative lexicon*, Cambridge, Mass., MIT Press (1996).

23.Schmid, H. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Vol. 12, pp. 44-49. Manchester, UK (1994).

24.Smith, B., and Kumar, A. Controlled vocabularies in bioinformatics: a case study in the gene ontology, *Drug Discovery Today: BIOSILICO*, 2(6), 246-252 (2004).

25.Vivanco, V. *El español de la ciencia y la tecnología*, Madrid, Arco Libros (2006).

26.Vivaldi, J. *Extracción de Candidatos a Término mediante combinación de estrategias heterogéneas*. PhD Dissertation. Barcelona, Universidad Politècnica de Catalunya (2001).

27.Wilks, Y., Slator, B. & Guthrie, L. *Electric Words,* Cambridge, Mass., MIT Press (1995).

28.Winston, M., Chaffin, R. & Herrmann, D. A taxonomy of part−whole relations, *Cognitive science* 11(4), 417-444 (1987).

A Framework for Uncertainty-Aware Visual Analytics in Big Data

Amin Karami[1,2]

[1] Computer Architecture Department (DAC), Universitat Politècnica de Catalunya (UPC), Campus Nord, C. Jordi Girona 1-3, 08034 Barcelona, Spain
[2] amin@ac.upc.edu

Abstract. Visual analytics has become an important tool for gaining insight on big data. Numerous statistical tools have been integrated with visualization to help analysts understand big data better and faster. However, data is inherently uncertain, due to sampling error, noise, latency, approximate measurement or unreliable sources. It is very important and vital to quantify and visualize uncertainties for analysts to improve the results of decision making process and gain valuable insights during analytic process on big data. In this paper, we propose a new framework to support uncertainty in the visual analytics process through a fuzzy self-organizing map algorithm running in MapReduce framework for parallel computations on massive amounts of data. This framework uses an interactive data mining module, uncertainty modeling and knowledge representation that supports insertion of the user's experience and knowledge for uncertainty modeling and visualization in the big data.

1 Introduction

The rapid development of data collection technologies in the last decades has led to accumulate the massive amounts of data referred to as Big Data. Today, big data has become an important and hot research topic and a very realistic problem in industry [15]. One of the important and vital aspects of the big data is its veracity, which accounts for the degree of uncertainty (e.g. vagueness, ambiguity, imprecision, and noise) in the content of user- or system-generated data. There are various factors that lead to data uncertainty including approximate measurement, data sampling fault, transmission error or latency, data integration with noise and so on [8][9]. These factors produce a lot of vague and imprecise data which implicitly contains valuable information. The representation of uncertainty is an ongoing unresolved problem and emerging as a problem of great importance in the field of visualization [16]. Hence, various companies and many researchers have been recently attempting to enable and identify new opportunities for markets and design innovative products through the uncertainty visualization in the big data era [1]. The value of uncertainty visualization in the big data is to accurately convey uncertainty to help users and decision makers understand potential risks and hidden knowledge, and to minimize misleading results and interpretations [7]. A challenging and key question is how users can effectively and efficiently understand the uncertain data in the big data sets and interact with them through the user interface. Interaction and user interface challenges are critical aspects of extreme-scale visual analysis to understand and cope with uncertainties. Adapting and applying visual analytics to the big data problems presents new challenges and opens new research questions [18]. Visual analytics is a relatively new field of study that aims at bridging this gap by integrating visualization and analytics in order to turn the information overhead into an opportunity [12]. Contributions in this area integrate information visualization, interaction and computational analysis by data mining techniques in order to transform massive data into knowledge. There have been several researches about visual analytics in the big data such as [18][3][13]. The disadvantages of the existing works are their inability to quantify and visualize uncertainty accurately.

The main contribution of this paper is a novel prototype system embracing uncertainty in the big data through the visual analytics. This system can provide valuable guidance through a close interaction between human operators, preprocessing data, refining model's parameters, building model, visualizing and understanding uncertainty in the data through the visual interface where operators are able to interact and provide desired inputs and configurations. For uncertainty modeling in the big data, we extend our previous work in [8] -a mechanism for mining and visualizing uncertainty in a centralized-batch data processing- through the MapReduce framework. MapReduce [5] is a programming model for executing distributed computations on massive amounts of data in order to model a decentralized-batch data processing. This system leads to an appropriate uncertainty-aware visualization in a massive amounts of data to help both experienced and novice users understand hidden knowledge through minimizing misleading interpretations. In section 2 we present background material related to uncertainty modeling, visual analytics and MapReduce framework. Section 3 presents our designed prototype for uncertainty visualization in the big data. Section 4 discusses proposed interface design suitability from a visual analytics perspective. Finally, section 5 concludes this paper and outlines future work.

2 Background

2.1 Uncertainty modeling

Uncertainty is widely spread in real-world data. A data can be considered uncertain, vague or imprecise where some things are not either entirely true nor entirely false. To model uncertainty, numerous techniques have been proposed, including probabilistic measures, Bayesian networks, belief functions, interval sets and fuzzy sets theory [4]. There has been a lot of research in the application of fuzzy sets theory to model uncertainty [8]. The Fuzzy set (FS) theory introduced by Zadeh [17] is a more flexible approach than classical set theory, where objects belong to sets (clusters) with certain degree of membership ranging [0..1]. In this paper, we use fuzzy sets theory as a mean to measure and quantify uncertainty.

2.2 Visual analytics process model

Visual analytics is defined as analytical reasoning supported by highly interactive visual interfaces that involves information gathering, data pre-processing, knowledge representation, interaction and decision making. A process model of visual analytics by Keim et al. [11] is illustrated in Fig. 1. According to Fig. 1, the first step is pre-processing such as data cleaning and data transformation over input data to be able to use it in the desired format for further investigations. After the pre-processing step, visualization methods and automated analysis methods are applied to the data. Afterward, automated analysis methods using data mining methods are applied to generate models. These models can be evaluated and refined by the user through a modification of initial parameters or selecting other type of analysis algorithms. User interaction with the visualization is needed to reveal information by applying different visualization techniques on the data such as descriptive analysis, graphical representations etc. Based on this interaction, the user can conduct the model building and refinement in the automatic analysis. Furthermore, knowledge can be gained during mentioned different types of user interaction. Finally, the feedback loop stores this knowledge of insightful analyses in the system and enables the analyst to draw faster and better conclusions in the future.

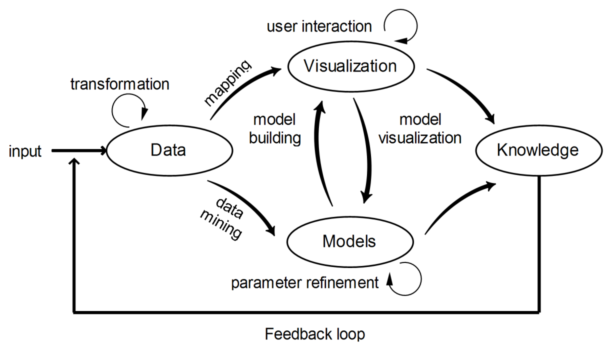


Fig. 1: The visual analytics process model (adapted from [11])

2.3 MapReduce framework for big data processing

MapReduce is a programming model popularized by Google for processing and generating large data sets with a parallel and distributed algorithm using many low-end computing nodes [14]. It is a scalable, fault-tolerant, and ubiquitous data processing tool gaining significant attention from both industry and academia. The main idea of the MapReduce is to hide details of parallel execution and allow users to focus only on data processing strategies [6]. The MapReduce model is composed of two procedures: *Map* and *Reduce*, written by the user. The *Map* function computes a set of intermediate key/value pairs (i.e. a list of $(key, value)$) from the input. The intermediate key/value pairs are then grouped together on the key-equality basis as $(key, list(value))$. The *Reducer* function performs a summary operation on the list of all values based on each unique key. This allows us to handle lists of values that are too large to fit in memory. The reduce function finishes the computation started by the map function, and outputs the final answer.

3 Proposed method: A Framework for Uncertainty-Aware Visual Analytics in Big Data

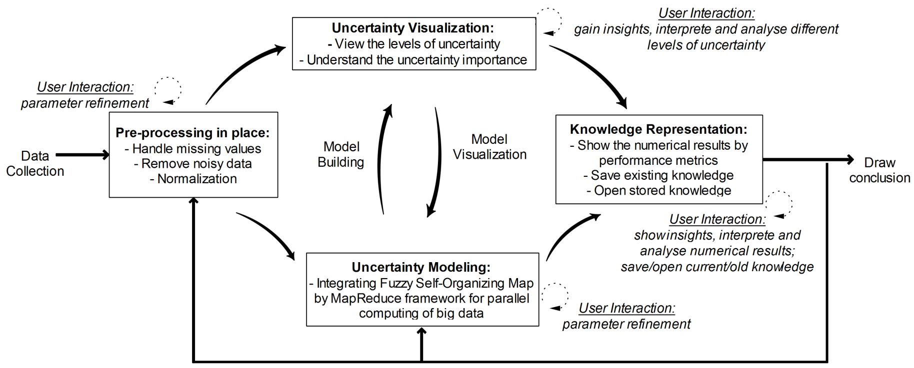


Fig. 2: The proposed model for visual analytics

Our proposed model (see Fig. 2) is derived from the model of visual analytics presented by Keim et al. in Fig. 1. Input data is collected, transformed and pre-processed, both automatically, through the visualization and the user interaction to be ready in the desired format for the analysis. After pre-processing, one of the main challenges is the selection of an appropriate technique for uncertainty modeling. The applied technique is based on our previous work in [8], a fuzzy self-organizing map for uncertainty visualization in uncertain data sets. We have extended our previous work integrating by MapReduce framework to be able to use the big data for uncertainty modeling and visualization (see section 3.1).
We add an interactive module in our prototype design that allows refinement of the applied techniques by the user. This prototype also consists of a graph-

ical representation to support uncertainty visualization as well as a descriptive analysis for knowledge representation to draw conclusion.

3.1 Uncertainty modeling

Our proposed uncertainty modeling is derived from our previous work in [8], called Fuzzy Self-Organizing Map (FSOM). In [8], we proposed a fuzzy self-organizing map algorithm using fuzzy c-mean (FCM) to model uncertainties based on a centralized-batch processing framework. FSOM works in three phases. In the first phase (we called it *fuzzy competition*), FCM technique has been employed to assign a membership degree in clusters' centers in terms of the input data. Then in the second phase (we called it *fuzzy cooperation*), all the clusters' centers cooperate by a Gaussian function with their neighbors in terms of the membership degree. Finally at the third phase (we called it *fuzzy adaption*), all the centers' positions are updated. These three phases are repeated, until the maximum number of iterations is reached or the changes become smaller than a predefined threshold.
First, in this section we present the main design for parallel FSOM based on MapReduce framework for a decentralized-batch processing which is depicted in Fig. 3. Then we explain how the necessary computations can be formalized as map and reduce operations in detail.

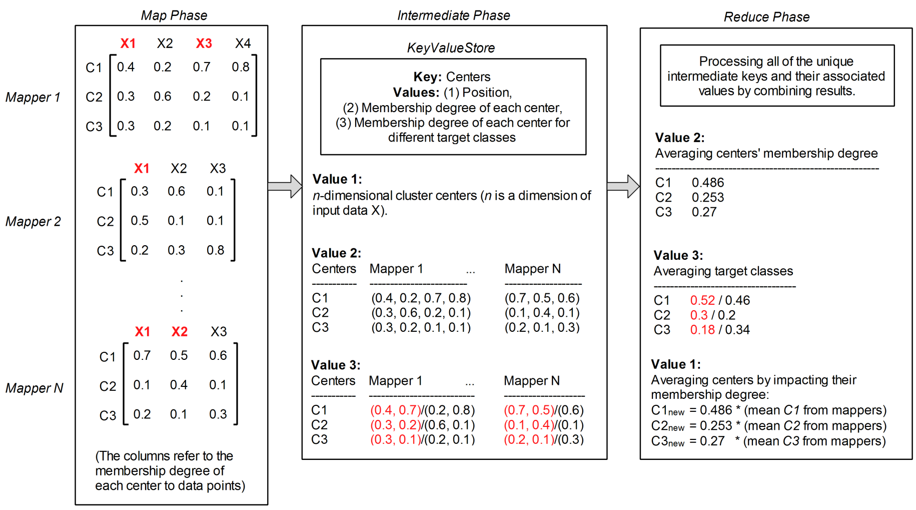


Fig. 3: The schematic of the MapReduce framework. $C1, C2, C3$ refer to cluster centers, $X1, X2, X3, X4$ refer to corresponding uncertain data points in each mapper, and the color of data points refers to target class (red = class 1 and black = class 2).

According to Fig. 3, The map phase applies FSOM algorithm from [8] performing the procedure of defining the membership degree of cluster centers from corresponding uncertain data points while the reducer phase performs the procedure of updating the new centers.
Map Function: The input data set is stored in Hadoop Distributed File System (HDFS) [2]. Data in HDFS is broken down into smaller pieces (called chunks) and distributed throughout the cluster. In this way, the map and reduce functions can be executed on smaller subsets of larger data sets, and this provides the scalability that is needed for the big data processing. MapReduce reads a single chunk of data on the input datastore, then call the map function to work on the chunk. The map function then works on the individual chunk of data and adds one or more key-value pairs to the intermediate *KeyValueStore* object. MapReduce repeats this process for each of the chunks of data, so that the total number of calls to the map function is equal to the number of chunks of data. Each mapper runs FSOM algorithm from [8]. The result of this phase is a *KeyValueStore* object that contains all of the key-value pairs added by the map function. The key is the cluster centers and the corresponding values are the position of centers in each mapper, the membership degree of each center, and the membership degree of each center for different target classes. After the map phase, MapReduce prepares for the reduce phase by grouping all the values in the *KeyValueStore* object by unique key in the intermediate phase.
Reduce Function: The reduce function scrolls through the values from the *KeyValueStore* to perform a summary calculation. We calculate the average of aggregated values to sum up the results (see Fig. 3).
The MapReduce framework is repeated until the clusters' centers do not change any more in the predefined number of iteration (we set 500 iterations) or a maximum purity has been reached. It is highly probable that the formed clusters containing normal data (correct classification) will have a number of abnormal data (incorrect classification) and vice versa. Therefore, we assigned a goodness value in range of [0..1] for each cluster by purity metric. The purity metric determines the frequency of the most common category/class into each cluster:

$$Purity = \frac{1}{n} \sum_{q=1}^{k} \max_{1 \le j \le l} n_q^j \tag{1}$$

Where, n is the total number of samples; l is the number of categories, n_q^j is the number of samples in cluster q that belongs to the original class $j (1 \le j \le l)$. A large purity (close to 1) is desired for a good clustering. If the all data samples in a cluster have the same class, the purity value set to 1 as a pure cluster.

3.2 Case study

To test our framework, we use a case study based on KDD-CUP'99 anomaly detection data set contains a standard set of data, which includes a wide variety of intrusions simulated in a military network environment. Each record in this data

set was labeled as either normal or as exactly one specific kind of attack. Attack labels are classified as DOS (denial-of-service, e.g. syn flood), R2L (unauthorized access from a remote machine, e.g. guessing password), U2R (unauthorized access to local superuser (root) privileges, e.g., various buffer overflow attacks), and probing (surveillance and other probing, e.g., port scanning). These different attacks are considered as a single attack by same labeling in our study. This data set consists of 41 features and 494021 records. In the experiments, 75% of data set is used as training and the rest is considered as testing in order to validate the functionality of the proposed method. To add uncertainty in the considered data set, we add a Gaussian white noise with a zero mean and the standard deviation with the normal distribution $[0, 2 * f]$, where, f is an integer parameter from the set of $\{1, 2, 3\}$ to define different uncertain levels for some features randomly.
This example helps security data analysts to monitor computer network traffic for security purposes. The challenge for an analyst is the discrimination between real attacks and normal traffic, where the nature of the traffic data is uncertain. The proposed framework for uncertainty-aware visual analytics enables insightful analyses in the system and allows the analyst to understand uncertainty for drawing faster and more accurate conclusions.

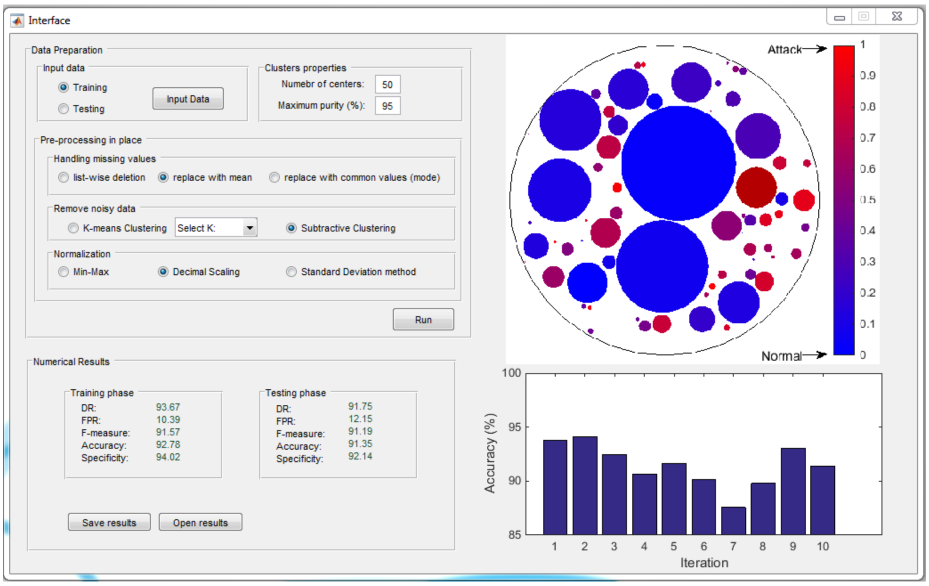


Fig. 4: Prototype design: uncertainty visualization in the big data including configuration section (top left); numerical results section for evaluating model by training and testing data (bottom left); uncertainty visualization plot (top right); the history of recent training (bottom right).

3.3 Performance measurement

To evaluate the results by the proposed algorithm, we apply several criteria including detection rate (DR), false positive rate (FPR), F-measure, accuracy and specificity (true negative rate) which are frequently used measures in the classification problems [10].

4 Prototype system design for visualizing uncertain clusters

The prototype design is depicted in Fig. 4 to provide an useful and effective uncertainty visualization of KDD-CUP'99 traffic data. This prototype was implemented by the MATLAB R2014b. The graphical user interface is designed to allow users for visual analytics through the embedded modules. The graphical interface has been divided into three main modules: data preparation (top left: input data, model properties and pre-processing), numerical results (bottom left: performance metrics for knowledge representation), and graphical representation (right: uncertainty visualization in the top and history of the training in the bottom). The operators can consistently train and test the data, then save the results for further usage or open preexisting results. To visualize the uncertainty, we map the magnitude of the propagated uncertainty to the size (to visualize the volume of the clusters) and the color (to encode the purity of the clusters) of nodes in a 2D plot defined as the projection of the 41 variables from the uncertain input big data. This projection is shown in Fig. 5.

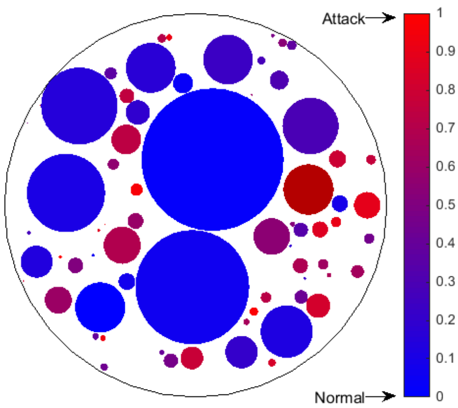


Fig. 5: Uncertainty visualization in the big data

The blue nodes denote normal traffic while the red nodes denote attack traffic. We multiply the third value of the *KeyValueStore* (see Fig. 3) to the corresponding red and blue colors in order to define the impurity of the normal and attack clusters. The more uncertain a cluster is, the more impure is its visual representation. For instance, the purple color denotes a 100% uncertainty in a formed cluster (purity = 0.5), neither completely normal nor attack traffic. This is useful for discovering the sources of uncertainty. This visualizes the effect of uncertainty and steers the user's attention towards the most reliable clusters over uncertain data points so that only the most reliable clusters are highlighted to the user. On the other hand, a large size of a node denotes the more uncertain data involved while a small size of a node denotes the less uncertain data involved which can be interpreted as outliers. As a consequence, these small nodes steer the user's attention visually towards the most unreliable nodes as outliers. This prototype design displays a high-level view of entire uncertain big data together with the numerical results. Preliminary results show that the designed prototype produces satisfactory outcomes. Users can steer and control uncertainty based on their own practices or analytic needs in the data preparing step, find outliers visually as well as distinguish visually reliable and unreliable clusters. User evaluations by zooming into sub-regions of clusters and reveal more details (i.e., details on demand) will be carried out in the future.

5 Conclusion

In this paper, we propose a framework for uncertainty-aware visual analytics in the big data. We integrated a fuzzy self-organizing map algorithm with MapReduce framework in order to execute a parallel computing on big data.
The prototype system includes a set of interactive visual representations that supports the analysis of the uncertain data and user interaction. We believe that this prototype system is useful when the analyst wants to extract a model that explains the behavior of uncertain data, find outliers visually and makes insightful decisions. The future work is needed by more user evaluations: zooming into sub-regions of uncertain clusters and reveal more details.

6 Acknowledgment

This work was partially supported by projects TIN2013-47272-C2-2 and SGR-2014-881.

References

1. Bendler, J., Wagner, S., Brandt, T., Neumann, D.: Taming uncertainty in big data. Business & Information Systems Engineering 6(5), 279–288 (2014)
2. Borthakur, D.: The hadoop distributed file system: Architecture and design. Hadoop Project Website 11(2007), 21 (2007)
3. Cook, K., Grinstein, G., Whiting, M., Cooper, M., Havig, P., Liggett, K., Nebesh, B., Paul, C.L.: Vast challenge 2012: Visual analytics for big data. In: Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on. pp. 251–255 (2012)

4. Correa, C.D., Chan, Y.H., Ma, K.L.: A framework for uncertainty-aware visual analytics. In: IEEE Symposium on Visual Analytics Science and Technology (VAST). pp. 51–58 (2009)
5. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI). pp. 137–150 (2004)
6. Grolinger, K., Hayes, M., Higashino, W.A., L'Heureux, A., Allison, D.S., Capretz, M.: Challenges for mapreduce in big data. In: IEEE World Congress on Services (SERVICES). pp. 182–189 (2014)
7. Jäckle, D., Senaratne, H., Buchmüller, J., Keim, D.A.: Integrated spatial uncertainty visualization using off-screen aggregation (2015)
8. Karami, A., Guerrero-Zapata, M.: Mining and visualizing uncertain data objects and network traffics by fuzzy self-organizing map. In: Proceedings of the AIC workshop on Artificial Intelligence and Cognition. pp. 156–163 (2014)
9. Karami, A., Guerrero-Zapata, M.: An anfis-based cache replacement method for mitigating cache pollution attacks in named data networking. Computer Networks 80, 51–65 (2015)
10. Karami, A., Guerrero-Zapata, M.: A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks. Neurocomputing 149, Part C, 1253–1269 (2015)
11. Keim, D.A., Bak, P., Bertini, E., Oelke, D., Spretke, D., Ziegler, H.: Advanced visual analytics interfaces. In: Proceedings of the International Conference on Advanced Visual Interfaces. pp. 3–10 (2010)
12. Keim, D.A., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H.: Visual analytics: Scope and challenges. Springer Berlin Heidelberg (2008)
13. LaValle, S., Lesser, E., Shockley, R., Hopkins, M.S., Kruschwitz, N.: Big data, analytics and the path from insights to value. MIT sloan management review 21 (2013)
14. Lee, K.H., Lee, Y.J., Choi, H., Chung, Y.D., Moon, B.: Parallel data processing with mapreduce: a survey. In: AcM sIGMoD Record 40. pp. 11–20 (2012)
15. Qian, H.: Pivotalr: A package for machine learning on big data. R Foundation for Statistical Computing 6(1), 57–67 (2014)
16. Riveiro, M.: Evaluation of uncertainty visualization techniques for information fusion. In: 10th International Conference on Information Fusion. pp. 1–8 (2007)
17. Zadeh, A.L.: Fuzzy sets. Information Control 8, 338–353 (1965)
18. Zhang, L., Stoffel, A., Behrisch, M., Mittelstädt, S., Schreck, T., Pompl, R., Weber, S., Last, H., Keim, D.: Visual analytics for the big data era – a comparative review of state-of-the-art commercial systems. In: IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 173–182 (2012)