DAGSTUHL
REPORTS

Volume 11, Issue 7, August 2021

Dagstuhl Reports, Vol. 11, Issue 7 ISSN 2192-5283

ISSN 2192-5283

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at
http://www.dagstuhl.de/dagpub/2192-5283

Publication date
December, 2021

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

Digital Object Identifier: 10.4230/DagRep.11.7.i

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

This basic framework can be extended by suitable contributions that are related to the program of the seminar, e. g. summaries from panel discussions or open problem sessions.

Editorial Office
Michael Wagner *(Managing Editor)*
Michael Didas *(Managing Editor)*
Jutka Gasiorowski *(Editorial Assistance)*
Dagmar Glaser *(Editorial Assistance)*
Thomas Schillo *(Technical Assistance)*

Contact
Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de
http://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 21331

Coalition Formation Games

Edited by

Edith Elkind[1], Judy Goldsmith[2], Anja Rey[3], and Jörg Rothe[4]

1 **University of Oxford, GB,** eelkind@gmail.com
2 **University of Kentucky – Lexington, US,** goldsmit@cs.uky.edu
3 **Universität Köln, DE,** anja.rey@tu-dortmund.de
4 **Heinrich-Heine-Universität Düsseldorf, DE,** rothe@hhu.de

Abstract

There are many situations in which individuals will choose to act as a group, or coalition. Examples include social clubs, political parties, partnership formation, and legislative voting. Coalition formation games are a class of cooperative games where the aim is to partition a set of agents into coalitions, according to some criteria, such as coalitional stability or maximization of social welfare. In our seminar we discussed applications, results, and new directions of research in the field of coalition formation games.

Seminar August 15–20, 2021 – http://www.dagstuhl.de/21331
2012 ACM Subject Classification Theory of computation → Algorithmic game theory
Keywords and phrases Coalition Formation, Cooperative Games
Digital Object Identifier 10.4230/DagRep.11.7.1
Edited in cooperation with Christian Laußmann

1 Executive Summary

Edith Elkind (University of Oxford, GB)
Judy Goldsmith (University of Kentucky – Lexington, US)
Christian Laußmann (Heinrich-Heine-Universität Düsseldorf, DE)
Anja Rey (Universität Köln, DE)
Jörg Rothe (Heinrich-Heine-Universität Düsseldorf, DE)

As mentioned, coalition formation games occur in many real-world settings. We are particularly interested in a subclass of coalition formation games, hedonic games, which were first proposed by Drèze and Greenberg [1] and later formalized by Banerjee et al. [2] and Bogomolnaia and Jackson [3]. Hedonic games are distinguished from general coalition formation games by the requirement that each agent's utility is wholly derived from the members of their own coalition.

This Dagstuhl Seminar brought multiple approaches and viewpoints to the study of coalition formation games, and in particular hedonic games, mainly from the perspective of computer science and economics. Particular topics that were discussed in talks and working groups include:

- succinctly representable preferences over coalitions;
- evolving preferences;
- the existence and verification of stable coalition structures (for various stability concepts);

DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- the computational complexity of finding or verifying stable or optimal partitions, or even determining whether such partitions exist;
- designing (if possible, efficient) algorithms for finding stable or optimal (or nearly so) coalition structures, or for verifying that a coalition structure is (nearly) stable or optimal;
- stability notions restricted to social networks or other networks;
- matching markets and matching under preferences, and their relation to hedonic games;
- dynamics of coalition formation;
- and group activity selection.

The overarching theme of this Dagstuhl Seminar was to bring together different communities working in coalition formation and hedonic games from various perspectives in computer science and economics and to bridge and bundle their research activities.

Much of the great atmosphere of the seminars at Schloss Dagstuhl comes from informal meetings besides the official schedule, with participants doing leisure activities together and enjoying other joint undertakings – this is, by the way, coalition formation in practice. Owing to the hybrid mode and pandemic-related restrictions, it was unfortunately not possible for us to organize group activities with all participants. However, due to the great technical support at Schloss Dagstuhl, the participants – online and on site – were able to take part in talks, discussions and working groups interactively to explore some of the challenging open questions of the field.

The organizers thank all participants for interesting talks and discussions. We also thank Schloss Dagstuhl for the technical preparation and support that made this hybrid seminar possible.

References

1 Dreze, Jacques H and Greenberg, Joseph. *Hedonic Coalitions: Optimality and Stability*. Econometrica: Journal of the Econometric Society (1980): 987-1003.

2 Banerjee, Suryapratim, Hideo Konishi, and Tayfun Sönmez. *Core in a Simple Coalition Formation Game*. Social Choice and Welfare 18.1 (2001): 135-153.

3 Bogomolnaia, Anna, and Matthew O. Jackson. *The Stability of Hedonic Coalition Structures*. Games and Economic Behavior 38.2 (2002): 201-230.

2 Table of Contents

Working groups

3 Overview of Talks

3.1 Bribery and Control in Stable Marriage

Niclas Boehmer (TU Berlin, DE)

Joint work of Niclas Boehmer, Robert Bredereck, Klaus Heeger, Rolf Niedermeier
Main reference Niclas Boehmer, Robert Bredereck, Klaus Heeger, Rolf Niedermeier: "Bribery and Control in Stable Marriage," Journal of Artificial Intelligence Research, 71:993–1048, 2021.
URL https://doi.org/10.1613/jair.1.12755

We initiate the study of external manipulations in Stable Marriage by considering several manipulative actions as well as several manipulation goals. For instance, one goal is to make sure that a given pair of agents is matched in a stable solution, and this may be achieved by the manipulative action of reordering some agents' preference lists. We present a comprehensive study of the computational complexity of all problems arising in this way. We find several polynomial-time solvable cases as well as NP-hard ones. For the NP-hard cases, focusing on the natural parameter "budget" (that is, the number of manipulative actions one is allowed to perform), we also conduct a parameterized complexity analysis and encounter mostly parameterized hardness results.

3.2 Hedonic Games with Deviation Rules as Solution Concepts

Grégory Bonnet (Caen University, FR)

Joint work of Grégory Bonnet, Thibaut Vallée

In hedonic games, solution concepts are considered as global characterization on how co-operation should be. However, we may want to model agents which have different notions of cooperation: egoistic agents, altruistic agents, etc. Thus, we propose a model of hedonic games, called deviation games, where agents locally define their own solution concept based on a set of individual constraints. These rules may be composed to express classical solution concepts, but may also highlight new kinds of solution concepts.

3.3 Group Activity Selection (on Social Networks): Progress or Theoretical Exercise?

Robert Bredereck (HU Berlin, DE)

In the Group Activity Selection Problem, players form coalitions to participate in activities and have preferences over pairs of the form (activity, group size) and the goal is to find a Nash (resp. core, individually, etc.) stable assignment of the players to the activities. In the Group Activity Selection with social networks players can further only engage in the same activity if the members of the group form a connected subset of the underlying communication structure. Athough being motivated and initiated by Dagstuhl seminar participants trying to solve real-world group activity selection, the model received a lot of theoretical attention but never returned into practice. In my talk, calling for a real-world implementation, I review some of the challanges and discuss possible next steps.

3.4 Dynamics Based on Single-Agent Stability in Hedonic Games

Martin Bullinger (TU München, DE)

Joint work of Felix Brandt, Martin Bullinger, Leo Tappe, Anaelle Wilczynski
Main reference Felix Brandt, Martin Bullinger, Anaëlle Wilczynski: "Reaching Individually Stable Coalition Structures in Hedonic Games", in Proc. of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pp. 5211–5218, AAAI Press, 2021.
URL https://ojs.aaai.org/index.php/AAAI/article/view/16658

The formal study of coalition formation in multiagent systems is typically realized using so-called hedonic games, which originate from economic theory. The main focus of this branch of research has been on the existence and the computational complexity of deciding the existence of coalition structures that satisfy various stability criteria. The actual process of forming coalitions based on individual behavior has received considerably less attention. In this talk, we study the convergence of simple dynamics based on single-agent deviations in hedonic games. We consider various strategies for proving convergence of the dynamics based on potential functions. In particular, we showcase methods for dealing with non-monotonic potential functions. On the other hand, it is a challenging task to pinpoint the boundary of tractability of stable states. We show how to construct complicated counterexamples with the aid of linear programs. These counterexamples can usually be used to prove computational intractabilities.

3.5 Testing Stability Properties in Graphical Hedonic Games

Hendrik Fichtenberger (Universität Wien, AT) and Anja Rey (Universität Köln, DE)

Main reference Hendrik Fichtenberger, Anja Rey: "Testing stability properties in graphical hedonic games", Auton. Agents Multi Agent Syst., Vol. 35(2), p. 26, 2021.
URL http://dx.doi.org/10.1007/s10458-021-09505-x

In hedonic games, players form coalitions based on individual preferences over the group of players they could belong to. Several concepts to describe the stability of coalition structures in a game have been proposed and analysed in the literature. However, prior research focuses on algorithms with time complexity that is at least linear in the input size. In the light of very large games that arise from, e.g., social networks and advertising, we initiate the study of sublinear time property testing algorithms for existence and verification problems under several notions of coalition stability in a model of hedonic games represented by graphs with bounded degree. In graph property testing, one shall decide whether a given input has a property (e.g., a game admits a stable coalition structure) or is far from it, i.e., one has to modify at least an ϵ-fraction of the input (e.g., the game's preferences) to make it have the property. In particular, we consider verification of perfection, individual rationality, Nash stability, (contractual) individual stability, and core stability. While there is always a Nash-stable coalition structure (which also implies individually stable coalitions), we show that the existence of a perfect coalition structure is not tautological but can be tested. All our testers have one-sided error and time complexity that is independent of the input size.

3.6 Fair Ride Allocation on a Line

Ayumi Igarashi (National Institute of Informatics – Tokyo, JP)

Joint work of Yuki Amano, Yasushi Kawase, Kazuhisa Makino, Hirotaka Ono

The airport game is a classical and well-known model of fair cost-sharing for a single facility among multiple agents. This paper extends it to the so-called assignment setting, that is, for multiple facilities and agents, each agent chooses a facility to use and shares the cost with the other agents. Such a situation can be often seen in sharing economy, such as sharing fees for office desks among workers, taxis among customers of possibly different destinations on a line, and so on. Our model is regarded as a coalition formation game based on the fair cost-sharing of the airport game; we call our model *a fair ride allocation on a line*. As criteria of solution concepts, we incorporate Nash stability and envy-freeness into our setting. We show that a Nash-stable feasible allocation that minimizes the social cost of agents can be computed efficiently if a feasible allocation exists. For envy-freeness, we provide several structural properties of envy-free allocations. Based on these, we design efficient algorithms for finding an envy-free allocation when at least one of (1) the number of facilities, (2) the capacity of facilities, and (3) the number of agent types, is small. Moreover, we show that a consecutive envy-free allocation can be computed in polynomial time. On the negative front, we show the NP-hardness of determining the existence of an allocation under two relaxed envy-free concepts.

3.7 The Impact of Tolerance in Schelling Games

Panagiotis Kanellopoulos (University of Essex – Colchester, GB)

Joint work of Panagiotis Kanellopoulos, Maria Kyropoulou, Alexandros A. Voudouris
Main reference Panagiotis Kanellopoulos, Maria Kyropoulou, Alexandros A. Voudouris: "Not all Strangers are the Same: The Impact of Tolerance in Schelling Games", CoRR, Vol. abs/2105.02699, 2021.
URL https://arxiv.org/abs/2105.02699

Schelling's famous model of segregation assumes agents of different types, who would like to be located in neighborhoods having at least a certain fraction of agents of the same type. We consider natural generalizations that allow for the possibility of agents being tolerant towards other agents, even if they are not of the same type. In particular, we consider an ordering of the types, and make the realistic assumption that the agents are in principle more tolerant towards agents of types that are closer to their own according to the ordering. Based on this, we study the strategic games induced when the agents aim to maximize their utility, for a variety of tolerance levels. We provide a collection of results about the existence of equilibria, and their quality in terms of social welfare.

3.8 Stable Partitions for Proportional Generalized Claims Problems

Bettina Klaus (University of Lausanne, CH)

Joint work of Bettina Klaus and Oihande Gallo

We consider a set of agents, e.g., a group of researchers, who have claims on an endowment, e.g., a research budget from a national science foundation. The research budget is not large enough to cover all claims. Agents can form coalitions and coalitional funding is proportional to the sum of the claims of its members, except for singleton coalitions which do not receive any funding. We analyze the structure of stable partitions when coalition members use well-behaved rules to allocate coalitional endowments, e.g., the well-known constrained equal awards rule (CEA) or the constrained equal losses rule (CEL).

For continuous, (strictly) resource monotonic, and consistent rules, stable partitions with (mostly) pairwise coalitions emerge. For CEA and CEL we provide algorithms to construct such a stable pairwise partition. While for CEL the resulting stable pairwise partition is assortative and sequentially matches lowest claims pairs, for CEA the resulting stable pairwise partition is obtained sequentially by matching in each step either a highest claims pair or a highest-lowest claims pair.

More generally, we can also assume that the minimal coalition size to have a positive endowment is $\theta \geq 2$. We then show how all results described above are extended to this general case.

3.9 Strict Core and Strategy-Proofness for Hedonic Games with Friend-Oriented Preferences

Bettina Klaus (University of Lausanne, CH) and Seckin Özbilen (University of Lausanne, CH)

Joint work of Bettina Klaus, Flip Klijn, Seckin Özbilen

We consider hedonic coalition formation problems with friend-oriented preferences; that is, each agent has preferences over coalitions she is part of based on a partition of the set of other agents into friends and enemies. We assume that for each of her coalitions, (1) adding an enemy makes her strictly worse off, (2) adding a friend together with a set of enemies makes her strictly better off, and (3) adding a friend makes her strictly better off than losing a set of enemies. We show that the partition associated with the strongly connected components (SCC) of the so-called friend-oriented preference graph is in the strict core. The SCC mechanism, which assigns the SCC partition to each hedonic coalition formation problem with friend-oriented preferences, is group strategy-proof. Furthermore, the SCC mechanism is the only mechanism that satisfies strategy-proofness and strict core stability.

3.10 Coalition Formation Games Span All of Social Choice! Towards a taxonomy.

Jérôme Lang (CNRS – Paris, FR)

I suggested to design a "Sandewallian" taxonomy for coalition formation problems, that turns out to specialize into hedonic games but also resource allocation, various forms of matching, group activity selection, peer selection, and voting. I presented a first step towards this taxonomy.

3.11 Tiered Coalition Formation Games with Extensions

Nathan Arnold (University of Kentucky – Lexington, US) and Judy Goldsmith (University of Kentucky – Lexington, US)

Joint work of Nathan Arnold, Judy Goldsmith, Sarah Snider

In 2017, Cory Siler proposed Tiered Coalition Formation Games, a structure that allows a simple, transitive representation for complicated, intransitive hierarchies of power. This CFG was inspired by a real-world approach for capturing the hierarchy of power in the Pokemon series of video games, and includes a preference framework in which Nash stability and core stability are equivalent. A stable partition is guaranteed to exist for any instance and was found by Siler in polynomial time, but an open problem remained of how to find a partition that is useful in real-world applications of the problem.

Our work proposes a new algorithm, inspired by the game of rock-paper-scissors, and a notion of epsilon-stability for this problem, both of which extend Siler's work and allow us to find more practical partitions for a given instance.

References

1 Cory Siler. *Tiered Coalition Formation Games*. The Thirtieth International FLAIRS Conference, 2017

3.12 Anchored Team Formation Games

Jacob Schlueter (Kyushu University – Fukuoka, JP), Chris Addington (University of Kentucky – Lexington, US), and Judy Goldsmith (University of Kentucky – Lexington, US)

Main reference Jacob Schlueter, Christian Addington, Judy Goldsmith: "Anchored Team Formation Games", in Proc. of the Thirty-Fourth International Florida Artificial Intelligence Research Society Conference, North Miami Beach, Florida, USA, May 17-19, 2021, 2021.
URL http://dx.doi.org/10.32473/flairs.v34i1.128501

We propose Anchored Team Formation Games (ATFGs), a new class of hedonic game inspired by tabletop role playing games. We establish the NP-hardness of determining whether Nash stable coalition structures exist, and provide results for three heuristics for this problem.

We highlight costs and benefits of each heuristic and provide evidence that all three are capable of finding Nash stable coalition structures, when they exist, much more quickly than a deterministic algorithm.

3.13 Team Counter-Selection Games

Matthew Spradling (University of Michigan – Flint, US)

Joint work of Dawson Crane, Zachary Holmes, Taylor Tadziu Kosiara, Michael Nickels, Matthew Spradling
Main reference Dawson Crane, Zachary Holmes, Taylor Tadziu Kosiara, Michael Nickels, and Matthew Spradling: "Team Counter-Selection Games". In IEEE Conference on Games 2021.
URL https://ieee-cog.org/2021/assets/papers/paper_192.pdf

We model team-versus-team contests with limited team size and an open pool of team member candidates. In this setting, candidates with a higher win rate against the open pool may be considered the "meta". Simply selecting the meta candidates leaves the team open to be countered by off-meta candidates which have lower overall win rates but high win rates against the meta in particular. A central authority in this model selects team members in hopes to counter the team composition they believe will be selected by an opponent. We present algorithms that generate a team of candidates based on observed metas and given that both parties have knowledge of pairwise election battle wins of the usable candidate pool. We provide different methodology to generate teams and analyze the teams generated by our algorithms using Pokémon GO team compositions to test them.

3.14 Housing Markets over Social Networks

Taiki Todo (Kyushu University – Fukuoka, JP) and Makoto Yokoo (Kyushu University – Fukuoka, JP)

Joint work of Takehiro Kawasaki, Ryoji Wada, Taiki Todo, Makoto Yokoo
Main reference Takehiro Kawasaki, Ryoji Wada, Taiki Todo, Makoto Yokoo: "Mechanism Design for Housing Markets over Social Networks", in Proc. of the AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021, pp. 692–700, ACM, 2021.
URL https://dl.acm.org/doi/10.5555/3463952.3464036

We investigate the effect of an underlying social network over agents in a well-known multi-agent resource allocation problem; the housing market. We first show that, when a housing market takes place over a social network with more than two agents and these agents have an option to avoid forwarding information about it to their followers, there does not exist an exchange mechanism that simultaneously satisfies strategy-proofness, Pareto efficiency, and individual rationality. It is also impossible to find a strategy-proof exchange mechanism that always chooses an outcome in a weakened core. These results highlight the difficulty of taking into account the agents' incentive of information diffusion in the resource allocation. To overcome these negative results, we consider two different ways of restricting the problem; limiting the domain of preferences and the structure of social networks.

3.15 Coalition Structure Generation Using Concise Characteristic Function Prepresentation

Makoto Yokoo (Kyushu University – Fukuoka, JP)

Main reference Vincent Conitzer, Tuomas Sandholm: "Complexity of constructing solutions in the core based on synergies among coalitions", Artif. Intell., Vol. 170(6-7), pp. 607–619, 2006.
URL http://dx.doi.org/10.1016/j.artint.2006.01.005
Main reference Xiaotie Deng, Christos H. Papadimitriou: "On the Complexity of Cooperative Solution Concepts", Math. Oper. Res., Vol. 19(2), pp. 257–266, 1994.
URL http://dx.doi.org/10.1287/moor.19.2.257
Main reference Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, Makoto Yokoo: "An asynchronous complete method for distributed constraint optimization", in Proc. of the The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings, pp. 161–168, ACM, 2003.
URL http://dx.doi.org/10.1145/860575.860602
Main reference Naoki Ohta, Vincent Conitzer, Ryo Ichimura, Yuko Sakurai, Atsushi Iwasaki, Makoto Yokoo: "Coalition Structure Generation Utilizing Compact Characteristic Function Representations", in Proc. of the Principles and Practice of Constraint Programming – CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20-24, 2009, Proceedings, Lecture Notes in Computer Science, Vol. 5732, pp. 623–638, Springer, 2009.
URL http://dx.doi.org/10.1007/978-3-642-04244-7_49
Main reference Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo, Marius-Calin Silaghi, Katsutoshi Hirayama, Toshihiro Matsui: "Coalition Structure Generation based on Distributed Constraint Optimization", in Proc. of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010, AAAI Press, 2010.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1809

Forming effective coalitions is a major research challenge in AI and multi-agent systems. coalition Structure Generation problem (CSG) involves partitioning a set of agents into coalitions to maximize social surplus. Traditionally, the input of the CSG problem is a black-box function called a characteristic function, which takes a coalition as input and returns the value of the coalition. As a result, applying constraint optimization techniques to this problem has been infeasible. However, characteristic functions that appear in practice often can be represented concisely by a set of rules, rather than treating the function as a black box. Then we can solve the CSG problem more efficiently by directly applying constraint optimization techniques to this compact representation. In this talk, I introduce several representative representations, i.e., graphical representations, synergy coalition group, an distributed constraint optimization problem, and describe how to solve CSG based on these representations.

3.16 Providing Good Model Explanations – a Call to Arms

Yair Zick (University of Massachusetts – Amherst, US)

In this talk, I present some of the ideas at the heart of model explainability, and their deep connections to ideas in cooperative game theory. In the past five years, several cooperative game theoretic solution concepts – and the Shapley value in particular – have been used extensively by the machine-learning community to explain the decisions of black-box models. Papers on the topic regularly appear in flagship ML conferences such as ICML and NeurIPS. However, the cooperative game-theory community has, by and large, remained somewhat uninvolved in this important development. The objective of this talk is to present some of the formal ideas underlying the generation of explanations for black-box machine-learning models, and how they map to game-theoretic solution concepts. We will cover other important criteria such as explanation privacy and fairness, and how they can inform our analysis of classic cooperative game-theoretic domains.

4 Working groups

4.1 Empathy in Dynamic Coalition Formation

Martin Bullinger (TU München, DE)

Joint work of Niclas Boehmer, Florian Brandl, Martin Bullinger, Grégory Bonnet, Edith Elkind, Anna Maria Kerkmann, Bettina Klaus, Seckin Özbilen, Sanjukta Roy

In research on stability in coalition formation, it is commonly assumed that preferences of agents over coalition structures are fixed and given a priori. The main task is then to identify stable states under various notions of stability. A weakness of such models is that they are only capable to capture a static model of coalition formation, where interaction of agents in coalitions plays no further role. In this working group, we study dynamics of coalition formation where single agents perform deviations based on incentives caused by instablities which may evolve over time. In particular, we seek to model aspects of empathy that cause agents to alter their preferences based on the evolution of new coalition structures. These encompass for instance laziness of agents to alter a status quo, or the emergence of friendships.

4.2 Hedonic Games under Evolving Preferences

Paul Harrenstein (University of Oxford, GB)

Joint work of Haris Aziz, Andreas Darmann, Hendrik Fichtenberger, Abheek Ghosh, Judy Goldsmith, Paul Harrenstein, Ayumi Igarashi, Joanna Kaczmarek, Micheal McKay, Anja Rey, Anaëlle Wilczynski, Gerhard Woeginger, Makoto Yokoo

Hedonic games provide a simple and versatile, but static framework to analyse coalition formation from a game-theoretic point of view. Its focus is on the formation of a single coalition structure. Coalition formation, however, is not a one-shot event. Rather, coalitions are formed repeatedly over time. The working group explored the possible directions in which to extend the formal framework of hedonic games to a temporal setting wherein players may evolving preferences over which coalitions to belong to and what would be appropriate dynamic solution and stability concepts. We expect our investigations also to have repercussions for compact representations of preferences, mechanism design, and the computational complexity surrounding this setting.

A first main question is how to model players' preferences over how coalitions change over time. We distinguished three types of temporal preferences:

T1. Sequences of $\vec{R}_i = R_i^0 R_i^1 R_i^2 \dots$ of static preferences over coalition structures. E.g.: *The first couple of years I prefer to be with these colleagues in a research group, then a couple of years with these, and after that with this group, etcetera.*

T2. Functions $\vec{R}_i$ mapping each *history* $\pi_1 \dots \pi_t$ of coalition structures to a static preference relation $\vec{R}_i(\pi_1, \dots, \pi_t)$. E.g., *I want to be in the same research group at least three years in a row, but prefer to move to a group at Harvard after having been four years in the same group.*

T3. Preference relations $\vec{R}_i \subseteq \vec{\Pi} \times \vec{\Pi}$ over coalition sequences. E.g.: *I want to be at a research group in Oxford infinitely often and at Cambridge at least once.*

Each of these types of preferences has its merits, depending on the situation one wishes to model. In our first effort to investigate how static stability concepts can be extended to such that take the dynamic structure into account, we focussed on T3 preferences. Drawing inspiration from the work of Kadam and Kodowski [1] on multi-period matching, we were able to define a dynamic concept of stability for hedonic games with evolving preferences.

References

1 S. V. Kadam and M. H. Kotowski,. *Multiperiod Matching.* Int. Economic Review 59(4): 1927–1947, 1998

4.3 Towards a Coalition Formation Card Game

Jérôme Lang (CNRS – Paris, FR) and Christian Laußmann (Heinrich-Heine-Universität Düsseldorf, DE)

Joint work of Florian Brandl, Robert Bredereck, Piotr Faliszewski, Paul Harrenstein, Shiri Heffetz, Jérôme Lang, Christian Laußmann

We started to design a card game based on additive hedonic games. Players (ideally, between 6 and 15) draw cards that indicate a positive or negative utility for a player, which they will get if they end up in the same coalition as this player. We experienced that the game becomes significantly more interesting if the players draw additional cards from time to time rather than knowing all utilities from the start. We want to further develop the game and finally test it in experiments. Our hope is to get a better understanding on how people act in such games compared to theoretically proposed (or optimal) strategies.

Participants

- Niclas Boehmer
TU Berlin, DE
- Grégory Bonnet
Caen University, FR
- Florian Brandl
Universität Bonn, DE
- Robert Bredereck
HU Berlin, DE
- Martin Bullinger
TU München, DE
- Edith Elkind
University of Oxford, GB
- Piotr Faliszewski
AGH University of Science & Technology – Krakow, PL
- Shiri Heffetz
Ben Gurion University – Beer Sheva, IL
- Martin Hoefer
Goethe-Universität – Frankfurt am Main, DE
- Anna Maria Kerkmann
Heinrich-Heine-Universität Düsseldorf, DE
- Bettina Klaus
University of Lausanne, CH
- Jérôme Lang
CNRS – Paris, FR
- Christian Laußmann
Heinrich-Heine-Universität Düsseldorf, DE
- Seckin Özbilen
University of Lausanne, CH
- Jörg Rothe
Heinrich-Heine-Universität Düsseldorf, DE
- Sanjukta Roy
TU Wien, AT

Remote Participants

- Chris Addington
University of Kentucky – Lexington, US
- Nathan Arnold
University of Kentucky – Lexington, US
- Haris Aziz
UNSW – Sydney, AU
- Vittorio Bilo
University of Salento – Lecce, IT
- Andreas Darmann
Universität Graz, AT
- Gabrielle Demange
Paris School of Economics, FR
- Hendrik Fichtenberger
Universität Wien, AT
- Abhek Ghosh
University of Oxford, GB
- Judy Goldsmith
University of Kentucky – Lexington, US
- Sushmita Gupta
The Institute of Mathematical Sciences – Chennai, IN
- Paul Harrenstein
University of Oxford, GB
- Ayumi Igarashi
National Institute of Informatics – Tokyo, JP
- Joanna Kaczmarek
Heinrich-Heine-Universität Düsseldorf, DE
- Panagiotis Kanellopoulos
University of Essex – Colchester, GB
- Michael McKay
University of Glasgow, GB
- Tomasz P. Michalak
University of Warsaw, PL
- Anja Rey
Universität Köln, DE
- Jacob Schlueter
Kyushu University – Fukuoka, JP

- Matthew Spradling
University of Michigan – Flint, US
- Taiki Todo
Kyushu University – Fukuoka, JP
- Anaëlle Wilczynski
CentraleSupélec – Gif-sur-Yvette, FR
- Gerhard J. Woeginger
RWTH Aachen, DE
- Makoto Yokoo
Kyushu University – Fukuoka, JP
- Yair Zick
University of Massachusetts – Amherst, US

Report from Dagstuhl Seminar 21332

Understanding I/O Behavior in Scientific and Data-Intensive Computing

Edited by

Philip Carns[1], Julian Kunkel[2], Kathryn Mohror[3], and Martin Schulz[4]

1 Argonne National Laboratory, USA
2 Universität Göttingen / GWDG, DE
3 Lawrence Livermore National Laboratory, USA
4 TU München, DE

Abstract

Two key changes are driving an immediate need for deeper understanding of I/O workloads in high-performance computing (HPC): applications are evolving beyond the traditional bulk-synchronous models to include integrated multistep workflows, in situ analysis, artificial intelligence, and data analytics methods; and storage systems designs are evolving beyond a two-tiered file system and archive model to complex hierarchies containing temporary, fast tiers of storage close to compute resources with markedly different performance properties. Both of these changes represent a significant departure from the decades-long status quo and require investigation from storage researchers and practitioners to understand their impacts on overall I/O performance. Without an in-depth understanding of I/O workload behavior, storage system designers, I/O middleware developers, facility operators, and application developers will not know how best to design or utilize the additional tiers for optimal performance of a given I/O workload. The goal of this Dagstuhl Seminar was to bring together experts in I/O performance analysis and storage system architecture to collectively evaluate how our community is capturing and analyzing I/O workloads on HPC systems, identify any gaps in our methodologies, and determine how to develop a better in-depth understanding of their impact on HPC systems. Our discussions were lively and resulted in identifying critical needs for research in the area of understanding I/O behavior. We document those discussions in this report.

Seminar August 15–20, 2021 – `https://www.dagstuhl.de/21332`
2012 ACM Subject Classification General and reference → General literature; Hardware → 3D integrated circuits; Software and its engineering → Software design engineering; Networks → Network performance analysis
Keywords and phrases I/O performance measurement, Understanding user I/O patterns, HPC I/O, I/O characterization
Digital Object Identifier 10.4230/DagRep.11.7.16

Understanding I/O Behavior in Scientific and Data-Intensive Computing, *Dagstuhl Reports*, Vol. 11, Issue 07, pp. 16–75
Editors: Philip Carns and Julian Kunkel and Kathryn Mohror and Martin Schulz
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Philip Carns (Argonne National Laboratory, USA, carns@mcs.anl.gov)
Julian M. Kunkel (Universität Göttingen / GWDG, DE, julian.kunkel@gwdg.de)
Kathryn Mohror (Lawrence Livermore National Laboratory, USA, kathryn@llnl.gov)
Martin Schulz (TU München, DE, schulzm@in.tum.de)

Dagstuhl Seminar 21332, "Understanding I/O behavior in scientific and data-intensive computing," brought together computer scientists from around the world to survey how I/O workloads are measured and analyzed on high-performance computing (HPC) systems, identify gaps in methodologies, and debate how to best apply this technology to advance HPC productivity. The hybrid, week-long event attracted 10 physical and 25 virtual attendees. They included representatives from seven countries spanning a variety of career levels in academia, industry, and government. The diversity of perspectives, combined with an intense week-long seminar format, offered an unprecedented opportunity for researchers to share ideas and spark new collaborative opportunities.

The seminar agenda was structured as a combination of full-group plenary sessions and subgroup breakout sessions. The plenary sessions were used to discuss high-level issues, vote on subtopics to investigate, relay results from breakout sessions, and present "lightning" talks that highlighted key issues in the community. The breakout sessions employed small groups (roughly five people each) to follow up in "deep dive" discussions on specific subtopics. This format enabled attendees from numerous time zones to remain productively engaged throughout the week. We also found it to be successful in facilitating discussion despite the COVID-19 safety considerations that prevented us from assembling at a single venue. The final day of the seminar was devoted to recording seminar findings in a timely manner while subject matter experts were still available for consultation.

Over the course of the seminar, the attendees converged on six high-level topics for deep dive discussions that are covered in this report.

- **Tools: Cross-Cutting Issues** (Section 4.1) explored common challenges in development of tools for understanding HPC I/O.
- **Data Sources and Acquisition** (Section 4.2) addressed how to acquire various forms of raw I/O instrumentation from production systems.
- **Analysis** (Section 4.3) focused on how to interpret I/O instrumentation once acquired.
- **Enacting Actionable Responses** (Section 4.4) investigated how to best utilize the outcomes from I/O analysis.
- **Data Center Support** (Section 4.5) focused on strategies for facility operators to facilitate better understanding of I/O behavior.
- **Community Support** (Section 4.6) explored the unique characteristics of the I/O analysis community and how to foster its growth.

This report presents a separate summary for each deep dive topic, including a survey of the state of the art, gaps, challenges, and recommendations. The report concludes in Section 5 with a summary of cross-cutting themes and recommendations produced by the seminar as a whole. We found that understanding I/O behavior in scientific and data-intensive computing is increasingly important in an era of evolving workloads and increasingly complex HPC systems and that several cross-cutting challenges must be addressed in order to maximize its potential.

2 Table of Contents

3 Brief Summaries of All Breakout Reports

Dagstuhl Seminar *Understanding I/O Behavior in Scientific and Data-Intensive Computing* explored a variety of subtopics in small breakout sessions over the first two days in order to identify key issues for broader discussion. This section briefly summarizes the findings of each of these early breakout sessions.

3.1 Tuesday Reports

The Tuesday breakout groups were organized around topics. Each group brainstormed ideas, raised questions, attempted to answer existing questions, and identified one most important question for the specific topic. Then, after 30 minutes, each group moved to the next topic. A document on each topic from each group was kept, providing the collective knowledge of all the attendees. The last group working on each topic summarized it and presented it in the plenary meeting.

3.1.1 I/O Workflow Analysis

The most important question identified was how to set up a proper abstraction level for workflow analysis. This covered the definition of workflows, the means to describe workflows and their characteristics, the analysis of workflows, and the optimization of workflows. The groups found that different workflow systems may define workflows differently. Typically, characteristics cover jobs, job steps, or tasks, organized hierarchically or as a directed acyclic graph with data or task dependencies, which may span high-performance computing, edge computing, and the cloud, and at multiple sites. The participants felt that a unanimous definition of workflow was not needed for I/O analysis and optimization.

A key issue raised was that a community standard for workflow specification does not yet exist. A portable and abstract representation would be useful for the community. Users should specify their workflows using such a description, and systems should be able to infer (learn) such descriptions using monitoring systems.

In the analysis stage, the perspective is generally system-centric and application-centric. A holistic view covering individual applications with their workloads but also the emerging workflows across sites was identified as important for tools. Having workflow knowledge, systems could utilize node local storage and burst buffers and improve job scheduling. However, the community is not yet able to exploit workflow specifications because of the lack of descriptions and monitoring systems that cover workflows.

3.1.2 Tools for I/O Analysis

The discussion started by analyzing the state-of-the-art tools for I/O performance monitoring and identifying the gaps, limitations, and unanswered questions for the I/O performance monitoring and analysis tools. There was consensus that many different I/O monitoring tools are available (at application, file system, and storage system levels). Obtaining an integrated view of the I/O performance is challenging, however, because the results are from diverse performance-monitoring technologies and often target different users (e.g., system administrators vs. applications developers). Moreover, the user typically has no access to monitoring or not enough knowledge to interpret the profiling results.

With regard to next-generation tools, the consensus was that I/O tools are needed for monitoring and analyzing applications using I/O on beyond-POSIX file systems and for different consistency models of I/O operations. In addition, tools are needed for supporting new storage heterogeneous platforms and emerging HPC applications, not necessarily using MPI. A big challenge ahead for I/O monitoring tools is the large amount of data collected. The tools will likely select only certain parts of the codes to be profiled or traced, disregarding data collection.

An open and challenging question is how tools can help understand the application and system behavior causing contention and performance degradation (during the workshop, this was called "I/O Weather" of the shared I/O infrastructure).

A second open question is how to help users and application developers understand application I/O performance in a context that they can utilize to improve performance. One possible option identified was the possibility for users having a report about their job's I/O performance with hints of varying difficulty levels for improving performance.

Further, standardization efforts for I/O tools were seen as important. Such actions might include common data format, core functionality, and open standards for usability and acceptance by the user community.

3.1.3 Changing Workloads and Their Requirements

Workloads can be defined as the I/O access patterns that hit the system partially or as a whole. This definition can be defined abstractly, and workloads can be also collected as traces. Nevertheless, characterizing workloads is difficult even for a specific application workflow, since the spatial and temporal granularity of the workflow can significantly change over its lifetime. Investigating the aggregate over HPC systems becomes even more complex, since many scientific applications typically run concurrently on them in an uncoordinated way.

Workload analysis (and I/O system optimizations) could be supported by information from applications and users about the intended use of I/O. Nevertheless, currently no formalism is available on exchanging this information; and many applications are also not documented well enough to derive it. In many cases, the information is restricted to whether an application is performing checkpoints or not.

Nevertheless, the HPC and data center communities expect that workloads are changing. One reason is the introduction of new storage interfaces, such as the S3 object storage interface. The impact of such emerging interfaces has not been thoroughly investigated yet. Also unclear is how hybrid systems consisting of POSIX-based parallel file systems and object stores will interact and whether existing applications will produce significant new access patters when using new interfaces. Another reason to expect new workloads is the widespread adoption of deep learning, which might lead to an increase in random accesses.

An open question is whether workload characteristics are really actually changing and on what systems and what the corresponding evidence is. If workloads are changing, then it is not obvious whether such changes are equally true for Tier-1, Tier-2, and Tier-3 systems. One therefore must collect and store a broad set of workload traces for different HPC sites to compare short- and long-term trends on HPC I/O usage.

3.1.4 Data Center Support

After the brainstorming, numerous questions were raised involving data center support to understand I/O behavior in scientific and data-intensive computing. We grouped the questions into five themes: (1) data center requirements to fulfill its role; (2) data center functionality for I/O optimization; (3) user applications compared with those of system administrators; (4) community and standardization for data centers; and (5) open challenges such as POSIX. This breakout helped inspire further discussion about the data center and community for the I/O.

3.1.5 Storage System Design

Important questions about new storage systems are how their design can help capture application I/O behavior and how it can help predict behavior. Both features are needed in order to support application performance portability. In general, these issues should be addressed in a co-design scheme where we work with parallel file system developers to specify application I/O behavior requirements, share application I/O patterns and workload descriptions, and discuss predictive models for the next generation of storage systems. While we have been able to specify the information we need from a new storage system (instrumentation data, system view and description) and what we can provide from I/O behavior analysis (high-level access patterns, mapping of application I/O activity to file systems and components of the storage system), a number of questions have been left open. For example, how much information about file system internals is needed by the user, and can an artificial intelligence model learn I/O behavior sufficiently well?

3.2 Wednesday Reports

The Wednesday workgroups were organized to cover the topics from Day 1 plus hot topics suggested by attendees after the discussion of Day 1 topics. Then a voting took place, and people were grouped according to their interest. Topics were covered in two longer sessions with varying groups and attendees asked to brainstorm issues, identify challenges, and collect solutions. The resulting reports were presented to the whole group for discussion, which then led to the following six core topics forming this report.

3.2.1 I/O Workflow Analysis

The analysis of I/O in workflows is a complex task. The workflows are often developed organically as the domain's needs grows. This development has led to a proliferation of workflow engines such as Pegasus, Swift, Fireworks, Kepler, Sandia Analysis Workbench, and Dask and domain-specific workflow engines (Galaxy, Taverna, etc.). However, our profiling tools are often limited to the scope of a specific application and lack the temporal correlation between multiple applications participating in a workflow. To achieve a cohesive analysis of workflow, we need to extend the profiling capabilities beyond an application, capture temporal data dependencies between applications, and present a standardized format for representing the I/O behavior. Additionally required are analysis tools that can operate on the standardized I/O logs and extract key I/O behaviors. For analysis, one could potentially leverage existing routines presented in workflow engines (typically used for load balancing, task scheduling, etc.) and extend them with Python-based wrappers to expose the data frame directly to the user.

3.2.2 Tools for I/O Analysis

Many tools have been developed to acquire I/O instrumentation data on HPC systems, with each instrumenting applications, libraries, file systems, and other storage subsystem components. These tools typically come with corresponding analysis components geared toward providing understanding of HPC I/O behavior, with presentation of this data tailored to application users, facilities staff, or I/O researchers. However, existing I/O analysis tools suffer from a number of shortcomings that impede their ability to provide meaningful insights into I/O behavior. One of the biggest shortcomings is that existing analysis tools are not well suited to aggregating data from multiple instrumentation sources into a holistic systemwide view, which is critical to understanding I/O behavior. Communication-related problems with tools also exist: these tools do not usually have mechanisms for communicating feedback (i.e., to users or libraries) and are often intended for certain types of I/O experts (researchers, system admins, etc.), making them incomprehensible to many users. Developers of I/O analysis tools should think more explicitly about interoperability with other tools in order to enable a more holistic I/O analysis tool ecosystem. Additionally, tool developers should rethink communication strategies to ensure they are providing users with meaningful feedback and are communicating with users in terms that are understandable and relevant to the analysis task at hand.

3.2.3 Standardization of Workflow Specification and Characterization

An I/O workflow usually starts from something simple (e.g., a sequential producer-consumer model) and then grows organically. Many workflow representations and engines were developed to meet the requirements of advanced features, such as fan-in/fan-out control and in situ analysis. However, the lack of a standardized workflow representation binds the analysis tools to one specific workflow engine. The lack of such representation also makes it difficult for users to switch between different workflow engines.

Designing an appropriate intermediate representation for an I/O workflow is not easy. In the same way that users compose a workflow, the standard should also start from the simplest case and add more features gradually. Existing workflow languages (from non-I/O fields, e.g., CWL) may serve as a good starting point. This representation should not be too generic and thus not put too much burden on the users. The ability to support hints such as I/O patterns between two interacting components is useful because it allows the implementation to perform various optimizations accordingly. Such hints can be platform dependent and thus should be made optional to allow portability. Both portability and reproducibility help encourage the standard adoption so we should keep them in mind when designing the workflow representation.

3.2.4 How to Better Engage Users in I/O Performance Tuning (“gamifying” I/O tuning)

Engaging users to tune I/O is a challenging task for several reasons, including low understanding of I/O performance, lack of analysis tools that are easy to use, and less importance given to I/O. Three main classes of users were identified in this breakout session: application developers/users, system administrators, and I/O library developers. Typical issues reported by these stakeholders are poor metadata or I/O performance in their application or library, variation of performance for the same I/O pattern, and increased workload. To improve participation of these users via “gamifying” I/O tuning, the workgroups discussed various incentive strategies. These potential strategies include granting rewards for fixing I/O problems, rewarding usage of efficient high-level libraries, comparing a user’s I/O performance with that of other users with the same I/O pattern, and showing a history of performance with previous runs. Having policies that limit usage of resources when a problem is not fixed (e.g., writing file per process by large-scale applications that burdens the metadata servers) was also discussed as a potential strategy. Improvements needed for facilities and the I/O community to increase user engagement in I/O tuning include easily understandable metrics to demonstrate benefit to the applications and I/O libraries, targeting of applications and workflows that provide the most benefit for the effort, continuous instrumentation, effective communication strategies, and I/O-tuning cookbooks.

3.2.5 Evaluation of Application Semantics and Matching Them to the Appropriate File System

The first question was whether tools could extract I/O semantics of applications matched for a specific file system. Some tools and automated approaches, including some benchmarking, are available. They should be used more, at least for the top 10 of the I/O-intensive applications in every center. Doing so would allow some pragmatic benefit.

Such tools will not be completely sufficient, however, because they observe only what an application is actually doing. They cannot capture what an application *should* do in terms of I/O, for example, when unnecessary information is written or when too frequent checkpoints are made. Thus, a structured dialog with either application users or developers cannot just be replaced. This can cover the observed I/O behavior as well as the intended one.

The groups also discussed the danger of assuming that the parallel file system would provide all the strict POSIX semantics. Such a provision could lead to slowdown for the application or the entire parallel file system. It might also lead to silent data corruption, which is even more critical. Therefore, the current situation remains unacceptable.

3.2.6 Sustaining the Tool Ecosystem and What Tools to Focus On

A variety of research tools exist for understanding I/O. While building a prototype for a new tool is easy, however, it is extremely hard to sustain the development and maintain the software such that it is useful for the community in the long run.

The groups identified the following methodology to discuss the topic: (1) investigate success stories (and failed attempts) for sharing of tools in order to identify common schemes; (2) discuss challenges; (3) identify and discuss approaches to mitigate issues; and (4) propose next steps.

The following tools and approaches were discussed: VI-HPS, IO500, IOR+MDTest, TAU, Darshan, MPI, SLURM, and Allinea tools – DDT/Map, Ellexus Breeze / Mistral.

Successful approaches have a combination of funding, luck, commitment to use software due to networking or initial collaboration projects, unfunded time commitment, research (and publication) opportunities, and community support. Here DOE/DOD centers provide better than do EU centers; presumably working for the “same boss” makes it easier to work on a shared roadmap.

Challenges that inhibit the usage of a tool include a lack of roadmaps, lack of documentation, lack of communication, and closed-source software (e.g., available only in an internal GitLab).

The proposed actions were as follows:

1. Document the history of various tools to learn from it (could be a publication).
2. Organize regular meetings covering operational and development aspects of tools.
3. Raise awareness of publication venues for software products – and publish there.
4. Try to get commitments from data centers for tool sustainment.
5. Join VI-HPS for analysis tools.
6. Establish joint steering for VI4IO, and add information and schedules to webpage.
7. Investigate whether software developers (instead of researchers) could be engaged in tool development.

3.2.7 I/O Performance Tuning – Actionable Strategies

I/O performance tuning can be seen from the viewpoint of different stakeholders: users (who have the goal of tuning their application and who can affect knobs only for their own job), administrators (who care about overall efficiency and stability and who can affect systemwide parameters), and developers (who design libraries and runtime systems for efficient I/O usage). Because of time limits we focused on the first role, the user.

The groups classified possible actions by their ease of execution or deployment, using the metaphor of picking fruit from a tree:

1. Picking fruit from the ground
   The first line of defense is datacenter-driven training for users, given by experts who often have simple rules of thumb that can help in many scenarios.
2. Picking low-hanging fruit
   Since the number of experts is often limited, some of their support can be automated. This can include automatic and machine-readable system configurations, summaries of current system state, and possibly interactive visualization.
3. Picking fruit from the middle of the tree
   The resulting data could be interpreted for users, providing basic feedback on how well their application used I/O, for example, with a red/yellow/green light indication. Such a report could be given in the job output file, sent via email or made available via a web form. Ideally, it could be completed with simple suggestions for improvement based on expert rules.
4. Picking high-hanging fruit
   The data could then be further used to automate basic guidance, for example, by providing mapping information or visualizing the application components and workflow on system data.
5. Picking fruit out of the sky
   As the final step, tuning could be fully automated, including system and state detection (from Step 2), output evaluation (from Step 3), workflow integration and basic guidance (from Step 4), and ultimately automatic mapping and tuning.

3.2.8 I/O Performance Analysis

Differentiating this topic from I/O performance tuning, variability, and tools is difficult. One approach is to look at I/O performance analysis from both the application and system levels and the available monitoring and analysis data depending on the access permission (e.g., end users, admins, experts, support). Currently, continuous monitoring and profiling provide I/O characterization, traces, server-side statistics and logs, scheduler logs, and job metadata. Various, and often datacenter-dependent, visualization tools such as dashboards and scripts help with the interpretation of the performance data, but they can be difficult to understand and respond to accordingly, especially for end users.

During the breakout session the following open research questions were identified. At the application level: Can I/O tools provide I/O efficiency metrics (e.g., for end users as a score)? Is there a way to describe boundaries of good/appropriate/poor performance? Can we detect the root cause of individual application performance degradation and slowdowns? At the system level: Can monitoring tools be preventive (detect an I/O system overload before system performance degradation)? Can jobs with problematic and performance-degrading I/O patterns be identified?

One idea of this breakout session was the creation of an application I/O utilization "score" that enables users and administrators to understand utilization characteristics at a glance in the context of system capability. This score could be based on a unification of the output of monitoring tools, which typically provide information to system administrators, and performance-tracing tools, which provide information for the users.

3.2.9 Scalable and actionable I/O log/trace analysis

Logs and traces of I/O operations are integral to analyzing and improving I/O behavior. One obstacle when processing logs and traces is the amount of data generated by multiple nodes, each with multiple processes doing I/O operations. Since this data collection is expensive in terms of both additional overhead during the program runtime and long-term storage, preprocessing or in situ data reduction and compression should be considered. The analysis of these traces should supply the user with a description of the I/O behavior. This can include efficiency metrics (such as percentage of available bandwidth used), usage over time, and some general classifications (e.g., possibly I/O bound). A further result of the analysis could be recommendations for the user: steps to take to improve the I/O pattern of the program. An evaluation of currently available tools showed that while many offer some form of this analysis, they usually need some form of expert knowledge. Ideally there should be a recommendation that is suitable for nonexpert users. In order to devise these recommendations. artificial intelligence methods may play a role, but that has to be evaluated on its own.

3.2.10 Open-Source Community-Supported Complete I/O Software Research and Development

Open source, community driven research software is important for supporting I/O understanding. Vendor-provided software often suffers due to lack of needed features and inability to verify methodologies for obtaining results due to being closed source. That said, there are many challenges in developing open source scientific software. Many times researchers are only funded to develop prototypes, and the tools that are made available are not robust or easily portable to new systems. It is very challenging for researchers to obtain funding to pay for development and maintenance, and the projects are maintained (if they are maintained at all) by small research groups instead of by the larger community.

We advocate for increased funding for maintenance and support of open-source tools and interfaces. The funding can support documentation, open data formats, customer support, porting of code to new platforms, and community input for development of new features. In particular for I/O understanding, we recommend:

- The establishment of user groups or virtual institutes, e.g., for I/O monitoring software;
- Better communication across the community about the goals of the software product, the state of the software, and how community members can contribute;
- Software availability on public platforms like GitHub;
- Reward or motivation for community members to volunteer their efforts to software products;
- Reproducibility testing for software, e.g., publishing test cases and expected results;
- Testing with standard benchmarks.

3.2.11 POSIX Replacement with Low-Level Specifications

A common notion in the HPC I/O community is that we should invent a more appropriate alternative to the POSIX I/O API and the POSIX semantics. The working groups concurred.

What is needed is a separate API that is simplistic, offering only a few well-defined typical I/O tasks. A novel API would clearly communicate this notion to developers. A transparent approach intercepting existing POSIX calls is bound to lead to confusion and disappointment.

Some of the simplifications that will lead to eventual advantages are the following:

- No directory notion but only a flat per-job data set
- No individual permissions checks but uniform permissions for the data set
- Forbiddance to read something written by the same job (because this should be communication)
- Introduction of a few mutual exclusive flavors for typical I/O activities per file or data object.

The reward for such a new API will be delayed, that is, until HPC applications can use it on top of production HPC file systems. The implementation should therefore start with a demonstration layer on top of existing parallel file systems, SQLite, S3, or PMEM. Then, the optimization potential for file system implementers can be demonstrated. Only after they adopt it will HPC applications be able to profit from the new approach.

3.2.12 I/O Performance Variability

The variability of I/O performance is the main reason for different execution times of similar user jobs. Many possible factors can cause this variability—not only that storage is a shared medium—and figuring out the real cause can be difficult. For example, the cause can be hardware issues (RAID rebuild after disk failure, failed redundant components such as servers or RAID controllers, reduced speed on network links), changed software versions (application, libraries, file systems), or bad configurations or bad I/O implementations. Improving the I/O, for example by using optimized striping options or by avoiding conflicting I/O (do not write to the same area from many clients, create files in separate directories), lessens this variability in many cases. Another option is to use dedicated resources, namely, a private parallel file system that uses node-local storage (e.g., BeeOND, GekkoFS).

3.2.13 Artificial Intelligence for Storage and I/O Systems

Work has been done successfully in artificial intelligence (AI) for storage, notably by vendors and researchers for predicting drive failures and characterizing/categorizing HPC applications in terms of I/O read and write characteristics. It is currently unclear, however, where AI is better suited than simpler statistical techniques for performing such characterizations. The groups recommended analyzing use cases and available data to determine applicability and possible performance gains if AI is utilized in this area for providing both understanding and automated feedback to applications and system components.

4 Deep Dive Topics

After two days of breakout sessions, which were described above, the group identified a series of topics to be investigated in more depth by individual groups. The following section describe the results of these group discussions (with a complete list of group members).

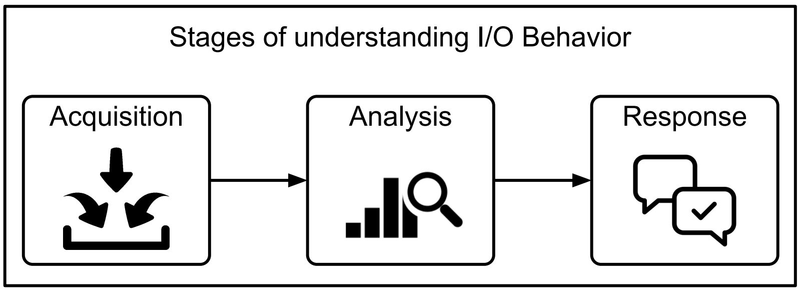


Figure 1 The process of understanding HPC I/O behavior can be decomposed into three core steps as defined by Ahlgren et al. [1]. The *acquisition* step includes components that produce instrumentation data and instrumentation data acquisition methods. The *analysis* step refers to visualization and other methods of deriving interpretations from data. The *response* step encompasses the enacting of policy or tuning changes.

4.1 Deep Dive Topic: Tools: Cross-Cutting Issues

Fahim Chowdhury (Florida State University, USA, fchowdhu@cs.fsu.edu),
Hariharan Devarajan (Lawrence Livermore National Laboratory, USA, hariharandev1@llnl.gov),
Ann Gentile (Sandia National Laboratories, USA, gentile@sandia.gov),
Jay Lofstead (Sandia National Laboratories, USA, gflofst@sandia.gov),
Kathryn Mohror (Lawrence Livermore National Laboratory, USA, mohror1@llnl.gov),
Devesh Tiwari (Northeastern University, USA, d.tiwari@northeastern.edu),
Chen Wang (University of Illinois at Urbana-Champaign, USA, chenw5@illinois.edu)

The heterogeneity and hierarchical nature of storage resources in HPC systems dictate the need for a holistic understanding of an application's I/O behavior in large-scale HPC systems. A compute-centric capturing of application behavior is not sufficient to capture the I/O behavior in these applications. First, compute resources are often isolated to a single node, with the scheduler providing exclusive access to these resources. In contrast, storage resources are often shared (e.g., shared burst buffers, I/O forwarders, and even global parallel file systems). This isolation increases the complexity of capturing accurate I/O behavior due to interference and colocation variability. Second, most middleware libraries and users have access to the compute cluster where the application runs but no access to storage resources. For instance, monitoring a Datawarp burst buffer at a system level or accessing Lustre logs to collect observed I/O behavior is extremely challenging. Third, compute-centric monitoring and tracing look at capturing individual elements of interest. However, the application's I/O behavior is often aggregated across the whole system (e.g., complex multijob workflows, producer-consumer paradigms). These factors dictate the need for a holistic set of tools spanning across software and hardware stack layers and providing a complete picture of the I/O behavior in modern HPC systems.

To achieve this holistic view of the I/O behavior across the whole HPC system, one can decompose this problem into three core steps as defined by Ahlgren et al. [1]. The *acquisition* step represents acquiring information from a collection of inputs such as instruments, monitors, profilers, and tracing libraries. This step also handles the actual data represented as instrumentation data, activity log data, application traces, and so on. The *analysis* step refers to extracting meaningful information from all the data collected by using visualization plots, interpretations, I/O timeline, and so on. This step converts the unprocessed data collected from various sources into analyzed I/O behavior for different jobs in the HPC system. The *response* step provides actionable items to improve the efficiency of storage systems to accelerate or support complex applications by enacting policies and procedures for tuning the I/O subsystem.

4.1.1 State of the Art

In the past decade scientists have built several tools for understanding the I/O behavior of applications in the three stages mentioned above.

4.1.1.1 Tools for *Acquisition*

The tools in the *acquisition* step aim to capture information about the application and system depending on the target layer of the software stack. These tools can be categorized as profiling, tracing, and monitoring tools. **Profiling** tools, including Darshan [13], Vampir [2], and TAU [14], record I/O operations for different interfaces such as STDIO, POSIX, and MPI-IO, or even higher-level I/O libraries such as HDF5 [4], pNetCDF [3], and ADIOS [4]. These tools report aggregate activities of an individual job running on the HPC system. This information can often be collected by using a low-overhead probing mechanism within the application. Additionally, in order to reduce variance within the information collected, they are run multiple times at similar times of the day to account for temporal variance.

Tracing tools collect operation-level information of I/O within the applications. This information is often collected during the software's runtime with detailed granularity and is nonaggregated over the application or system software. These tools are often built for a specific layer of the software stack, such as an application, higher-level I/O library, or storage system logs. Examples of these tools include Recorder [7], Darshan (with DXT) [5], Score-P [6], and Vampir [2].

Monitoring tools are passive tools that hook into existing applications to extract system-level information. These tools are built from a system-centric point of view and are often deployed at the system software layer. These tools use a passive probing mechanism to efficiently monitor the I/O activity, hardware health, and even the overall system. Examples of these tools, such as LDMS [10], DCDB [11], PIKA [8], TACCStats [9], and Beacon [12], are widely seen in many clusters, supercomputers, and data centers. These tools collect systemwide resource utilization and performance counter information from well-known sources (e.g., `/proc`), which can include I/O and storage-related information, and make it available for runtime display and postprocessing diagnostic analysis. There is widespread interest in using information from monitoring systems to understand the effects of the system on application performance. Hence, much work has enabled low overhead to support subsecond data collection, although not at the fidelity of typical application profiling tools. The tools aim to explain the I/O behavior of a single application/job.

4.1.1.2 Tools for *Analysis*

The tools in *analysis* aim to consume the activity/log data produced by the acquisition tools to extract meaningful information about the application/system. Currently, these are built as companion tools for existing acquisition tools. For instance, Darshan has its own postprocessing tools called PyDarshan [13] and VaniDL [14]. Recorder is accompanied by recorder-viz [15], which extracts information from recorder traces and visualizes them for the user. Additionally, we have a collection of multirun analysis tools such as IOMiner [19], TOKIO [18], Gauge [24], and ARCHER-LASSI [20]. Moreover, some generic visualization and analysis toolkits further enhance analysis capabilities of the existing analysis tools such as Elastisearch [21] with Grafana [22] for Mistral [23], pandas [25] DataFrames for Darshan with generic plotting libraries, and Splunk [26] as a general log and regex parser to look for patterns. All these tools aim to extract useful information from existing acquisition tools. Hence, they also target a single application/job as their primary use case.

4.1.1.3 Tools for *Response*

The tools to provide *response* to the various I/O stakeholders are, at best, limited. We can categorize these tools as human-centric or automated. For human-centric tools, some recent work suggests providing hints through visualization plots and bottleneck analysis to assists developers and system admins to improve the I/O performance. Examples of these tools are monitoring tools providing a visual dashboard to look at activity data with simple drill up and drill down visualizations to manually catch I/O bottlenecks or using Vidya [16], a compiler-based approach to extract code structures and highlight code improvements for the user.

Some automated tools, for example Apollo [17], can provide feedback to middleware libraries to improve I/O decisions such as data placement, scheduling, and data prefetching. These tools are in the early stage of development and require a lot of polishing to be utilized on production machines.

The biggest theme in the state-of-the-art tools is the lack of support for a holistic view of I/O. Here, holistic means looking at several layers of the software stack and multiple applications running in a workflow. A plethora of tools have been developed for different scopes of information. However, these tools are incompatible with each other in all three stages needed to understand the I/O behavior accurately.

4.1.2 Gaps

In this subsection we look at the individual steps involved in understanding I/O behavior and identify the gaps in modern workflows.

The tools for data acquisition lack support for capturing I/O behavior on complex systems and workflows. We identify three major gaps in acquiring complete information from modern systems.

4.1.2.1 Low-Fidelity Acquisition

First is *low-fidelity acquisition*. The diversification of tools has led to lower fidelity of collected data, affecting the collection of required features to understand I/O behavior. This further results in non-reproducible results logs. For instance, data acquisition through current state-of-the-art tools is not reproducible on the same system. Furthermore, the acquired logs are tightly coupled with the system architectures and cannot be portably transferred to other machines with similar architectures. This gap is even wider when we consider multiple applications across geodistributed clusters working together.

4.1.2.2 Missing Hierarchical Scope

The second gap is *non-hierarchical scope*. Current state-of-the-art tools do not consider I/O at multiple levels of scope, such as process, job, workflow, and system. This gap often leads to restrictive analysis capabilities, which hinder the ability to observe I/O bottlenecks at different levels and their propagation within the system. This is even direr with the rise of several high-level I/O libraries because I/O operations by the application do not match the I/O seen by the underlying storage system.

4.1.2.3 Missing Compatibility of Logs

The third gap is *noncompatible logs.* Every tool has its own format for collecting and storing I/O information. These representations are often motivated based on the level of information collection and scope of the data acquisition. However, these representations are not compatible with each other. Hence, users analyzing multiple logs must build custom merging tools for their use cases, leading to restrictive information about their I/O behavior.

4.1.2.4 Gaps in Data Analysis

The tools for I/O analysis are closely tied with the acquisition tools. This leads to the following four gaps in data analysis.

First, *incompatible formats* cannot be associated with analysis because of the different scope and target of collected data. Hence, stitching these logs together is a critical gap that needs to be filled in the analysis space.

Second, the *complexity of analysis* is limited to simple statistical graphs. The analysis needs to be evolved to support complex I/O analysis such as performance bottleneck identification, I/O roofline, and error and fault detection.

Third, *standardization of analysis* is nonexistent in the current tools. The standardization needs to provide common API, functionality, and visualizations that analysts as a community can extend.

Fourth, tools for the *response* have several gaps that need to be addressed for a cohesive understanding of I/O behavior. These gaps can be divided into three categories.

First, *response standardization APIs* can enable users and system software libraries such as schedulers, buffering software, and prefetchers to improve the I/O performance of the overall system by providing feedback to these complex systems and libraries.

Second, *I/O quality of service* is lacking in modern storage systems. The response tools can predict I/O expectations based on temporal and spatial information and drive several system optimization algorithms or even I/O policies.

Third, *autotuning capabilities* are not present in the feedback tools. These capabilities can enhance feedback accuracy as these tools learn more about the system and application behaviors.

4.1.2.5 Gaps in understanding I/O behavior

Some gaps in understanding I/O behavior span across all tools and services. These can be categorized as systems, geolocations, and stakeholders. In terms of systems, we see the growth of heterogeneous resources in modern HPC systems. The tools to understand I/O behavior need to evolve past monolithic parallel file system design to heterogeneous storage cluster architectures. The tools have to capture the heterogeneity with a low-overhead, modular design for adapting to new technologies and with a flexible metric system for different hardware. In terms of geodistributed systems, the tools need to address the reproducibility and portability of I/O behavior across these systems. We need to build models that are transferable across machines and are platform-independent.

Moreover, the whole process of understanding I/O behavior needs to cater to the different stakeholders involved: I/O users, I/O practitioners, and I/O researchers. Users require the understanding that their application is getting “good” I/O performance. This is currently not defined and often ignored by the users. I/O practitioners require systemwide information and statistics over multiple applications. These should be presented in the form of visualizations similar to those provided by compute-centric monitoring tools. I/O researchers require detailed I/O patterns (e.g., overlapping I/O, write-after-write patterns, and file/block size information) or even raw data accesses.

Addressing these gaps across the different stages of tools, multiple application workflows, heterogeneous storage solutions, and multiple stakeholders is essential for meeting the growing I/O challenges in the HPC community.

4.1.3 Challenges

Understanding I/O behavior for multitenant, heterogeneous, and geodistributed systems running complex multistage workflows depends on closing the gaps in tools we described in preceding sections. However, we identified several challenges that the community faces in this path.

4.1.3.1 Multistep Tools

The multistep nature of the tool from acquisition to response makes it hard for researchers to build a cohesive toolkit that can tame the hardware, software, and application workflow complexity all at once. Therefore, we believe approaching this problem in a modular way (i.e., one step at a time) could help us take our first step toward a holistic set of tools for understanding I/O behavior. We discuss the key challenges we envision in each of these three steps as follows.

Data Sources and Acquisition for I/O faces three challenges: collection granularity, quality vs. performance trade-off, and interoperability. First, we have different tools for acquiring a different level of information for each software stack and heterogeneous hardware. However, the nature of collected data varies so diversely that it is challenging to combine different data collected by different acquisition tools.

Second, most acquisition tools are based on the trade-off of performance vs. quality of data collected. Since the required nature of data varies for each stakeholder, it is a persistent challenge to identify the bare minimum set of metrics that acquisition tools should always collect. This decision needs standardization and should not vary based on an HPC site.

Third, the diversification of the tools for data acquisition has resulted in a lot of manual effort to make a group of tools interoperable. Currently, this is done based on the research requirement without any standardization. Researchers build their own parser over existing tools to interpret data for their research. These challenges hinder our development toward a holistic set of tools for data acquisitions for I/O.

Analysis of I/O logs is critical for understanding the environment and the nature of the application. However, currently analysis is tightly coupled with the acquisition step. Hence, building comprehensive analysis tools for understanding I/O is challenging for the following three reasons.

First, *tight coupling with data acquisition tools* makes every analysis engine out there nontransferable because of the unique format for each log. Additionally, analysis tools often are bounded the resolution of data provided by the acquisition tools. We need to move away from non-interoperable tools to a standardized log format for capturing I/O behavior so that we can build analysis tools that can be used and developed by the community (e.g., the sklearn package in Python for machine learning works on any data set that conforms to a particular format).

Second, *a lack of standard analysis definitions* within the tools results in every analysis tool recreating simple statistical functions such as aggregate I/O, observed data access patterns, and small vs large I/O. As we move to more complex analysis, most of the current analysis tools fall short of providing the required flexibility for data analysts to experiment on. Also, the nature of the standardized analysis is unknown. Some potential venues are application I/O performance, I/O efficiency, I/O bottleneck analysis, parallel vs. serial I/O, detection of producer-consumer patterns, or even I/O performance prediction metrics.

Third, *analysis performance* is often neglected and performed serially. Consuming logs for large-scale application from several different software layers demands smart analysis frameworks such as Dask [27], Spark [28], and Hive [29]. Currently, the tools process these logs serially with no distributed or out-of-core analysis. As the resolution of data increases along with multicomponent logs, smarter analysis engines will be needed that can scale efficiently on modern HPC systems.

To provide meaningful responses to the storage system or users, we need to be able to convert our analysis into "actionable" items. However, actionable items are not well defined in the literature. Some efforts have been made by auto-optimizers such as Vidya and autotuning tools to define them at the application or system level, respectively. However, the actionable items lack standardization. Additionally, the delivery mechanism for response (e.g., online APIs or passive event log) is ambiguous for tools to implement meaningful actions for their middleware libraries and system systems. Another challenge with actionable items is that they require multiple data points (significantly temporally separated) with analytics to identify potential issues/feedback. Single data points are not sufficient because of performance variability. Furthermore, actionable items can have a side-effect on other applications and processes in the system. For instance, if we improve the I/O for one application, do we adversely affect others sharing the resources?

4.1.3.2 Heterogeneous Storage Environment

Heterogeneous storage environments in HPC systems lead to challenges at three levels: application, hardware/software, and system level. The scope of I/O behavior broadens as we go from the application to the system level.

At the application level, acquisition tools attached to the job have low impact with periodic logging. However, this reduces the resolution of the data collected. Full-scale logging has been shown to significantly impact the application's runtime (10–15% in some cases). This leads to a trade-off between the performance and quality of collected data by the tools.

Additionally, as we see a diversification of applications and storage resources, these tools are often utilized to perform best-fit or matching analysis for an application. These new requirements lead to the building of "yet another profiling tool" within the community.

Moreover, modern applications seldom run in isolation. They are generally part of a bigger workflow where several components exchange data among themselves. The application-centric approach in current tools limits their ability to detect and collect I/O behavior across applications, multiple heterogeneous resources (CPU and GPUs), and even different software stacks. The middleware libraries, which accelerate data management within the application, currently profile and manually manage this information within themselves.

At the hardware and software level, we see a growth in several hardware solutions with multiple levels of abstractions through middleware libraries. This complex and deep hardware and software stack (e.g., high-level libraries to low-level I/O interfaces to software abstraction through parallel file systems to the local filesystem) makes collecting I/O data extremely challenging.

Furthermore, several of these resources are allocated dynamically by the scheduler, forcing the tools to adapt a more dynamic approach to resource discovery and collecting information.

Additionally, the heterogeneity of the environment (e.g., faster node-local devices) potentially shifts the trade-off between overhead and data collected within tools.

Furthermore, the performance measurements and optimizations within tools are not portable across different systems because of machine architecture differences such as interconnect, node core and memory count, storage device type and distribution, bandwidth to storage system from compute nodes, and storage system device ages.

External tools such as profilers and tracers are often more costly than storage-level monitoring solutions at the system level. At the storage level, however, these monitoring tools lose the application behavior. Additionally, the logs at the storage level grow rapidly and hence cannot be left running.

Furthermore, since storage is often a shared resource, the potential I/O problems are much broader than the storage device or software. These problems could stem from the interconnect, a memory bus, or other system components. This diversity prompts us to monitor more of the system from the I/O perspective to understand the actual source of issues, since storage device contention, hardware issues (slowly failing components), and latencies are less reliably the source of the problem than they were in the days of hard disk drives.

Assessing the system state and tuning across multiple system components without any isolation is a challenging task.

4.1.3.3 Geodistributed Systems

Tools for understanding I/O behavior in a geodistributed/multisystem setup are challenging from two aspects: the scope of I/O behavior and the nature of application workflows.

In terms of *scope*, the primary challenge is to gain global insights from the system. This is because a performance bottleneck for a particular workflow could be attributed to individual components all the way to the topology of the system. This situation dictates the need for a comprehensive view of the overall system.

Moreover, the limited scope of optimization (e.g., a job or application) can adversely impact other applications in the system. The reason is that I/O tuning is often on a shared resource that can adversely affect other applications if prioritized for one application.

Additionally, multitenant nodes (e.g., fat nodes, CPU + GPU applications, or in situ computations) require tools to span multiple applications because of the shared nature of the resources. Application workflows span multiple systems that could be potentially geographically distributed. The tools for analyzing such I/O behaviors need to be cross-platform and highly portable. Analyzing the I/O over architecturally different HPC sites is extremely challenging, even if application behavior is collected.

Furthermore, the data dependencies from workflow make temporal relations of data access important. However, the capturing resolution of tools along with clock skewness makes this analysis extremely hard.

Furthermore, the current tools do not support or provide enough information for workflow-specific analysis. We need to move toward more portable, holistic, and robust tools that address these challenges.

4.1.4 Next Steps and Recommendations

In this subsection we present four groups of high-priority next steps that can help drive the tool community forward. First, the primary goal of the tools is to provide insight. The community should put priority **focus on analysis** as opposed to getting stuck in optimization of design, particularly in data collection techniques.

Table 1 Tools Chapter Summary. Gaps/Challenges: green/yellow/red for increasing difficulty/large problems. Gaps and Challenges exist for all categories. For Next Steps: green/yellow/red for least to most important.

		Gaps	Challenges	Next Steps
Multistage	Acquisition	🟨	🟩	🟩
	Analysis	🟥	🟨	🟥
	Response	🟥	🟥	🟥
Heterogeneous Environment		🟨	🟩	🟨
Geodistributed/Multisystem		🟥	🟥	🟥
IO Stakeholders	Admin	🟨	🟨	🟨
	Users	🟥	🟥	🟥
	Researcher	🟩	🟩	🟩

Second, **agreement on the relevant I/O performance metrics** is required in order to ensure that the tools are capturing the necessary information to accurately characterize I/O behavior.

Third, success of tools relies on the adoption of tools by stakeholders. **Development of a compelling set of use cases** is essential to demonstrate potential benefit to users and system administrators. A particular challenge in the application of tools is that a single user may derive limited benefit in performance tuning of I/O within that user's application. Given the interdependence of applications due to shared resources, bigger gains may be obtained by using tools to understand applications' I/0 behavior and use that information to enable the system (through enacting fixes at the hardware, middleware, I/O libraries, system software, and subsystem levels) to automatically manage the overall system efficiency. Aggregate user performance and workload throughput may be the level at which substantial value from tuning may be realized.

Furthermore, a **balance of tools** is needed to provide the production hardening required for use and deployment while still supporting exploratory development needed for research to thrive.

References

1 V. Ahlgren et al. *Generic Monitoring System Requirements*. 2018 IEEE International Conference on Cluster Computing (CLUSTER), 2018, pp. 532–542, doi: 10.1109/CLUSTER.2018.00069.

2 Holger Brunst and Matthias Weber. Custom hot spot analysis of HPC software with the Vampir performance tool suite. In Alexey Cheptsov, Steffen Brinkmann, José Gracia, Michael M. Resch, and Wolfgang E. Nagel, editors, *Tools for High Performance Computing 2012*, pages 95–114, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

3 Jianwei Li, Wei keng Liao, Alok Choudhary, Robert Ross, Rajeev Thakur, William Gropp, Rob Latham, Andrew Siegel, Brad Gallagher, and Michael Zingale. Parallel netCDF: A high-performance scientific I/O interface. *SC Conference*, 0:39, 2003.

4 Jay Lofstead and Robert Ross. Insights for exascale I/O APIs from building a petascale I/O API. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 87:1–87:12, New York, NY, USA, 2013. ACM.

5 Cong Xu, Shane Snyder, Omkar Kulkarni, Vishwanath Venkatesan, Philip Carns, Suren Byna, Robert Sisneros, and Kalyana Chadalavada. "dxt: Darshan extended tracing". In *CUG*, 2017.

6 Jan Frenzel, Kim Feldhoff, Rene Jaekel, and Ralph Mueller-Pfefferkorn. Tracing of multithreaded Java applications in Score-P using bytecode instrumentation. In *ARCS Workshop 2018; 31th International Conference on Architecture of Computing Systems*, pages 1–8. VDE, 2018.

7 Chen Wang, Jinghan Sun, Marc Snir, Kathryn Mohror, and Elsa Gonsiorowski. Recorder 2.0: Efficient parallel I/O tracing and analysis. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1–8. IEEE, 2020.

8 Robert Dietrich, Frank Winkler, Andreas Knüpfer, and Wolfgang Nagel. Pika: Center-wide and job-aware cluster monitoring. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 424–432. IEEE, 2020.

9 Todd Evans, William L Barth, James C Browne, Robert L DeLeon, Thomas R Furlani, Steven M Gallo, Matthew D Jones, and Abani K Patra. Comprehensive resource use monitoring for HPC systems with TACC stats. In *2014 First International Workshop on HPC User Support Tools*, pages 13–21. IEEE, 2014.

10 Steven Feldman, Deli Zhang, Damian Dechev, and James Brandt. Extending LDMS to enable performance monitoring in multi-core applications. In *2015 IEEE International Conference on Cluster Computing*, pages 717–720. IEEE, 2015.

11 Alessio Netti, Micha Müller, Axel Auweter, Carla Guillen, Michael Ott, Daniele Tafani, and Martin Schulz. From facility to application sensor data: modular, continuous and holistic monitoring with DCDB. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–27, 2019.

12 Bin Yang, Xu Ji, Xiaosong Ma, Xiyang Wang, Tianyu Zhang, Xiupeng Zhu, Nosayba El-Sayed, Haidong Lan, Yibo Yang, Jidong Zhai, et al. End-to-end I/O monitoring on a leading supercomputer. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pages 379–394, 2019.

13 Argonne National Laboratory. PyDarshan: HPC I/O characterization tool. `https://www.mcs.anl.gov/research/projects/darshan/docs/pydarshan/index.html`.

14 Hariharan Devarajan, Huihuo Zheng, Xian-He Sun, and Venkatram Vishwanath. Understanding I/O behavior of scientific deep learning applications in HPC systems. 2020.

15 UIUC. Recorder-Viz. `https://github.com/wangvsa/recorder-viz`.

16 Hariharan Devarajan, Anthony Kougkas, Prajwal Challa, and Xian-He Sun. Vidya: Performing code-block I/O characterization for data access optimization. In *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, pp. 255–264. IEEE, 2018.

17 Neeraj Rajesh, Hariharan Devarajan, Jaime Cernuda Garcia, Keith Bateman, Luke Logan, Jie Ye, Anthony Kougkas, and Xian-He Sun. Apollo: An ML-assisted real-time storage resource observer. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*, pp. 147-159. 2020.

18 Glenn K Lockwood, Nicholas J Wright, Shane Snyder, Philip Carns, George Brown, and Kevin Harms. Tokio on ClusterStor: connecting standard tools to enable holistic I/O performance analysis. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2018.

19 Teng Wang, Shane Snyder, Glenn Lockwood, Philip Carns, Nicholas Wright, and Suren Byna. IOMiner: Large-scale analytics framework for gaining knowledge from I/O logs. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 466-476. IEEE, 2018.

20 Karthee Sivalingam, Harvey Richardson, Adrian Tate, and Martin Lafferty. Lassi: metric based I/O analytics for HPC. In *2019 Spring Simulation Conference (SpringSim)*, pages 1–12. IEEE, 2019.

21 Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W Godfrey. Mining modern repositories with Elasticsearch. In *Proceedings of the 11th working conference on mining software repositories*, pages 328–331, 2014.

22 Daniel Thalmann. An interactive data visualization system. *Software: Practice and Experience*, 14(3):277–290, 1984.

23 Julian Kunkel, Eugen Betke, Matt Bryson, Philip Carns, Rosemary Francis, Wolfgang Frings, Roland Laifer, and Sandra Mendez. Tools for analyzing parallel I/O. In *High Performance Computing: ISC High Performance 2018 International Workshops, Frankfurt/-Main, Germany, June 28, 2018, Revised Selected Papers*, volume 11203, page 49. Springer, 2019.

24 Del Rosario, Eliakin, Mikaela Currier, Mihailo Isakov, Sandeep Madireddy, Prasanna Balaprakash, Philip Carns, Robert B. Ross, Kevin Harms, Shane Snyder, and Michel A. Kinsy. Gauge: An interactive data-driven visualization tool for HPC application I/O performance analysis. In *2020 IEEE/ACM Fifth International Parallel Data Systems Workshop (PDSW)*, pp. 15–21. IEEE, 2020.

25 Wes McKinney et al. pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

26 Jon Stearley, Sophia Corwell, and Ken Lord. Bridging the gaps: Joining information sources with Splunk. In *SLAML*, 2010.

27 Matthew Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th Python in Science conference*, vol. 130, p. 136. Austin, TX: SciPy, 2015.

28 Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud 10*, no. 10-10 (2010): 95.

29 Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: a warehousing solution over a Map-Reduce framework. *Proceedings of the VLDB Endowment 2*, no. 2 (2009): 1626-1629.

4.2 Deep Dive Topic: Data Sources and Acquisition

Stefano Markidis (KTH Royal Institute of Technology – Stockholm, SE)
Sivalingam Karthee (Huawei Technologies – Reading, GB)
Sandra Mendez (Barcelona Supercomputing Center, ES)
Roland Laifer (KIT – Karlsruher Institut für Technologie, DE)
Osamu Tatebe (University of Tsukuba, JP)
Michèle Weiland (EPCC, The University of Edinburgh, GB)

This section explores the extraction of the raw information that guides our understanding of I/O behavior in scientific and data-intensive computing. Potential data sources include both hardware (e.g., disks, networks, and compute nodes) and software (e.g., file systems, libraries, and applications). The scope of data acquisition can vary from application-level profiles all the way to full-scale data center telemetry. The following sections summarize the state of the art in data sources and acquisition, identify gaps and challenges, and propose recommendations for how to address them.

4.2.1 State of the Art

4.2.1.1 Benchmarks

Several benchmark programs have been developed to evaluate storage systems with various metrics. Benchmark programs measure the performance under several I/O access patterns. An excellent list of parallel I/O benchmarks, applications, and traces is available at `https://www.mcs.anl.gov/~thakur/pio-benchmarks.html` and for benchmarks at `https://www.vi4io.org/tools/benchmarks/`.

Benchmarks can be divided into two categories: microbenchmarks and application benchmarks. Microbenchmarks measure performance using a simple access pattern. Typical microbenchmark tools are IOR and MDtest, available at `https://github.com/hpc/ior`. IOR is a parallel I/O benchmark that can be used to measure the bandwidth of parallel storage systems using various interfaces and access patterns. MDtest tests the peak metadata rates of storage systems under different directory structures. The IO500[1] combines these benchmarks into meaningful patterns.

Application benchmarks measure the I/O performance of real applications. Application benchmarks include DLIO for scientific deep learning workloads, WRFIO for the Weather Research and Forecasting model, and MACSio to mimic I/O workloads for a wide variety of real applications.

4.2.1.2 Workload Generators

A workload generator creates a complex benchmark that exhibits the temporal and spatial performance characteristics of a combination of applications representing the total workload of a system. Workload generators analyze the jobs that are running on a system, for instance by taking a snapshot of that system over a fixed period of time, and creating a "condensed" version of that workload. The aim is for the generated workloads to exhibit performance characteristics similar to real workloads; this is particularly important at the system specification and acceptance testing stages. For I/O performance, the main objective of testing a system using a generated workload as opposed to a single benchmark application is to understand the impact of the workload on shared resources, namely, the network, the storage subsystem, and the (parallel) file system. Unlike benchmarks, workload generators create situations where multiple different applications that form part of the workload compete for resources and thus create contention. Workloads can span a broad range of application areas (traditional HPC as well as emerging application areas such as AI), performance characteristics and I/O patterns.

One example of an existing workload generator is the Kronos [1] tool, developed by the European Centre for Medium Range Weather Forecasts as part of the NEXTGenIO project.

4.2.1.3 Monitoring Data Access and Retrieval

Monitoring tools provide information about the HPC systems state such as the jobs running, state of the different components, and workload. From the point of view of I/O, all this information is collected along the I/O path. As shown in Figure 2, different components of HPC systems are sources of the data, depending on the scope of the analysis, granularity, and components selected. Because of the complexity and different information that can be obtained in each element, different monitoring tools are run for collecting information.

[1] `https://io500.org`

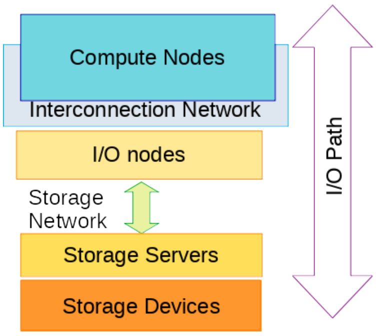


Figure 2 I/O hardware along the I/O path. Each component is a source for data acquisition.

HPC centers have different monitoring tools and in several cases have their own implementation tools that provide data logs at different levels. Data can be obtained from compute nodes, storage network, I/O nodes, storage nodes, and storage devices. For example, at the compute node level, we can monitor information related to memory, frequency, CPU usage, hardware counters, and so on. Although there is an effort to provide I/O data at this level, it depends on the file system and tools provided to capture the information for each job, file, application, or user.

4.2.1.4 Storage System Health Checks

Checking the health of the storage system is an important data source for understanding the I/O behavior. For example, if a Lustre file system is down, the I/O of all applications will hang and then will normally continue after the file system is available again. Or if a part of the system is degraded, for example, a RAID rebuild after disk failure, failed redundant components such as servers or RAID controllers, or reduced speed on network links, this might have a huge impact on the I/O performance. Such issues are likely with huge storage systems. Sometimes the issues are healed automatically; for instance, if a storage server hangs and does not respond to heartbeat messages, it might be automatically killed and restarted.

4.2.1.5 High-Level Libraries and APIs for I/O

Large-scale HPC applications use high-level libraries and APIs to optimized read and write operations from/to the parallel file systems. Such high-level HPC libraries allow for coordinating these operations across several processes, reading and writing operations to shared files, and provide means to optimize the efficiency of the operations, for example, by collecting several requests from multiple processes and merging them.

MPI-I/O is among the most used approaches for parallel I/O. MPI-I/O was first featured in MPI-2 and released in 1997 [2]. MPI is a convenient setting for parallel I/O since write and read operations can be conceived as send and receive operations. Moreover, MPI provides mechanisms for collective operations exploiting communicators and noncontiguous data access with MPI derived datatypes. MPI-I/O implementations, such as ROMIO, provide two-phase and data-sieving optimization techniques for parallel I/O on HPC systems [3].

Among other important HPC libraries for parallel I/O are HDF5 and NetCDF. HDF5 (Hierarchical Data Format) is an open-source library that supports parallel I/O [4]. It provides a machine-independent data storage format and user-defined datatypes and metadata.

NetCDF (Network Common Data Form) is a software library that provides machine-independent data formats to support operations on array-oriented scientific data. NetCDF is used largely by the weather and climate simulation community.

4.2.1.6 Metrics

The most common I/O metrics are as follows:

- Bandwidth (read & write)
- I/O operation per second (IOPs)
- Metadata operations (stat, open, close, create, remove, etc.)
- Latency, which can be useful as an indication whether the system is overloaded or something is broken

Other metrics can be more specific, for example, measuring time for file system operations such as creating a small file or checking the queue depth of storage devices. At the device level, some storage arrays are able to report the worst time for read and write operations on their devices.

Depending on the point of the view of the I/O analysis, metrics report different values. For example, tools such as Darshan capture information related to all the files open by the application but cannot obtain data at the system level. If the analyst needs information at the system level (e.g., at the storage node level), other data must be collected in order to obtain the appropriate metric. Having a metric in each component could be advantageous because we could identify the slowest I/O component and the possible source of an I/O bottleneck, but it is not trivial to measure the same metrics in each component.

Furthermore, metrics depend on the I/O patterns, and it is not trivial to define a value to have an indicator of poor or appropriate I/O performance. For example, if one needs to have an indicator of I/O performance for an specific I/O system, one should use benchmarks configured for specific patterns. For this situation, the IO500 score is a good indication of performance for a range of access patterns.

4.2.2 Gaps

A policy is needed for making the I/O telemetry and log collection to be on at all times except for extreme conditions. Networking telemetry, for example, is on, but I/O profiling is either opt-in or opt-out in most cases.

4.2.2.1 Benchmarks and Workload Generators

Numerous benchmarks are regularly used on HPC systems (from HPL and HPCG to STREAM and IOR) to derive the performance of the whole system in production. However, the practice of using workloads (in particular, generated workloads that represent a specific system load) to assess the performance of HPC systems is uncommon. For I/O performance in particular, this is a problem because single benchmarks cannot replicate the type of shared resource contention that workloads will inevitably encounter. This is a clear gap in the evaluation of I/O subsystems.

4.2.2.2 Monitoring Data Access and Retrieval

Data from the whole system is needed in order to have a holistic view of the I/O behavior. A monitoring tool or a set of monitoring tools could provide a holistic view of the I/O behavior at different levels or granularity. For example, if a job opened a file at runtime, one could track how the resources were being used by such a job for that file. This will require a common log data format for the monitored data that considers the granularity and level of the data.

4.2.2.3 Storage System Health Checks

System administrators are usually aware of an unhealthy or degraded storage system since they get alerts and have access to corresponding monitoring systems. Users, however, are frequently unaware of such issues. This is a gap since it might help users understand the I/O behavior or the reason for I/O performance variability.

4.2.2.4 High-Level Libraries and APIs for I/O

New and emerging storage systems, such as heterogeneous storage architectures and object stores, require the extension of established parallel I/O libraries and the creation of new approaches to enable the usage of these systems [6].

In particular, high-Level libraries need to consider that multiple storage might be available to an application: for instance, there might be a storage system on a compute node or shared by compute nodes and global storage shared among all the compute nodes. Ongoing research is extending and designing new libraries for these platforms. However, no established approach for these efforts exists. In addition, new storage technologies such as object stores are gaining more space on HPC systems. There exist emerging high-level APIs for programming object stores, such as Intel's DAOS [7] and Seagate Motr [5]. However, it is not clear how MPI I/O and other traditional I/O parallel approaches need to change to support object stores.

4.2.2.5 Metrics

Metrics differ depending of the I/O component monitored, because an I/O operation changes as it is processed along the data path. Therefore metrics cannot be obtained with the same tool in all the I/O components or with the same time interval.

4.2.3 Challenges

4.2.3.1 Benchmarks

Numerous benchmarks and I/O kernels exist that represent different I/O patterns and are used to evaluate the different components of the I/O software stack. Therefore, the main challenge is to provide a guide for selecting an appropriate benchmark or set of benchmarks to evaluate the performance at different levels and for the different I/O analyst roles (user, developer, or administrator).

Also needed are benchmarks for applications such as deep learning or I/O workflow in order to analyze whether we should use existing benchmarks or we need to implement new ones. For example, for deep learning applications several data formats and frameworks exist that present different I/O behavior; therefore a benchmark may be needed that allows selecting the data format and framework to best represent the I/O behavior of the deep learning application.

4.2.3.2 Workload Generators

Benchmarking whole system behavior with workloads is challenging for multiple reasons.

1. Generating the workloads is difficult because, in order for them to be representative, the whole system workload that will inform the generated workload has to be analyzed carefully. This process involves monitoring, recording, and interpreting the temporal and spatial performance characteristics of multiple snapshots of a system under load.
2. Using a generated workload as a mandatory benchmark as part of a procurement would deliver the highest impact, because the characteristics of the benchmark would directly influence the design of the system. Doing so is difficult in practice, however, because vendors often do not have access to large systems in house and thus have to project performance from a small to a large system. Such projection is already difficult with single benchmarks.

4.2.3.3 Monitoring Data Access and Retrieval

The main challenges in monitoring data are as follows.

- Monitor data that allows correlating the I/O performance with computation, communication, or memory performance issues.
- Monitor data that allows tracking I/O performance issues along the data path to identify I/O bottlenecks sources.
- Evaluate how to monitor the influence of the memory usage on the I/O behavior.
- Evaluate what data needs to be monitored in heterogeneous compute nodes such as CPU+GPU, because these nodes have their own I/O techniques to reduce the impact of transfer data between the host and GPU devices.
- Define and decide what and how to monitor data for I/O workflows.

4.2.3.4 Storage System Health Checks

System providers do not report storage-system-related issues for multiple reasons.

- They do not want to report every issue since this looks like the system is frequently in bad shape.
- They do not want to bother users with reduced performance that could also be caused by other jobs competing on the same resources.
- This information is usually available automatically only on internal systems. Site-specific solutions might be required to make this information available for users.

4.2.3.5 High-Level Libraries and APIs for I/O

Developers of libraries for parallel I/O face two main challenges. The first is the rise of heterogeneous systems with local (to compute nodes) and global storage systems. The second challenge is the uptake of new storage technologies, such as object stores, with different data consistency. The main challenge for library developers is understanding how to express parallel I/O in existing or new frameworks to take advantage of these new emerging systems without precluding the possibility of optimization.

4.2.3.6 Metrics

In order to allow I/O analysis by users, administrators, and developers, the metrics to be collected need to be clearly defined. Depending on the role of the I/O analyst, the metrics differ. The information therefore should be shown at different granularity levels such as job, application, file, and file system.

Furthermore, current I/O workflows present anther challenge. Are classical metrics sufficient for measuring I/O performance, or must we define composed metrics or new metrics?

4.2.4 Next Steps and Recommendations

4.2.4.1 Benchmarks

In IO500 a big effort was made to provide a set of benchmarks (IOR and MDtest) to measure the performance of the I/O system and define an appropriate setting of the benchmark parameters. This set is used mainly by system administrators. A similar idea could be done to guide users and developers in the selection of benchmarks.

Benchmarks for deep learning applications exist but are not focused on I/O. They can provide a start to evaluate whether they can be extended for I/O or whether new benchmarks must be developed.

4.2.4.2 Workload Generators

All system benchmarking activities should, to a degree, include generated workloads. In order to achieve this, generating representative and realistic workloads must be simplified. This process relies on better monitoring data that is explicitly tied to the workloads running on the system. At any one point, one should be able to link system behavior to workload and ideally to specific jobs in the workload. Using this information, one then can generate workloads that can expose specific and measurable system behavior.

4.2.4.3 Monitoring Data Access and Retrieval

As recommendations, we can start by considering the following:

- Identifying the information required by users, developers, and administrators to understand I/O behavior and identify I/O performance issues. The information that still is not being monitored could be added to current monitored data.
- Selecting tools that provide compatible data log format to facilitate the analysis and visualization of I/O behavior along the I/O path.
- Providing or selecting monitoring tools that can show the information considering the role of the I/O analyst (user, developer, or administrator).

4.2.4.4 Storage System Health Checks

As a first step system providers need to decide if and which storage-related issues they want to expose to their users and if they want to provide the additional effort to set up such a system. For example, they could create a web page that reports the issue with timestamps when they existed and possible impacts.

Even if system providers do not want to expose such issues, they could run regular performance checks to get system health and performance data over time and make this data available to their users. Users could then compare their application performance variation with the I/O system performance data.

4.2.4.5 High-Level Libraries and APIs for I/O

A recommendation for parallel I/O developers is to investigate the extension of established libraries, such as MPI I/O and HDF5, to use emerging heterogeneous systems and object stores efficiently. Having support in established libraries would allow porting HPC applications to new I/O solutions without using other APIs.

4.2.4.6 Metrics

Some work that defines useful metrics such as application I/O “risk” (LASSi) or I/O efficiency at application level (POP2) can be considered by HPC centers and users in the I/O analysis process.

I/O efficiency usually is measured as bandwidth relative to an empirical maximum. However, users sometimes require the I/O efficiency related with the whole application running, as is done in the POP2 project.

By using an I/O risk metric one can identify cases such as OpenFOAM that could show a “high risk” because it is (mainly) metadata intensive.

It could also useful to define a metric that assesses performance relative to system capability and capacity at the time of the job running. Also useful would be a metric for I/O workflow, for example identifying how often files are used.

References

1 https://github.com/ecmwf/kronos

2 Thakur, Rajeev, Ewing Lusk, and William Gropp. Users guide for ROMIO: A high-performance, portable MPI-IO implementation. No. ANL/MCS-TM-234. Argonne National Lab., IL (United States), 1997.

3 Thakur, Rajeev, William Gropp, and Ewing Lusk. “Data sieving and collective I/O in ROMIO.” In Proceedings. Frontiers’ 99. Seventh Symposium on the Frontiers of Massively Parallel Computation. IEEE, 1999.

4 Folk, Mike, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. “An overview of the HDF5 technology suite and its applications.” In Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases, pp. 36-47. 2011.

5 Narasimhamurthy, Sai, et al. “The SAGE project: a storage centric approach for exascale computing.” In Proceedings of the 15th ACM International Conference on Computing Frontiers. 2018.

6 Wei-der Chien, Steven, Stefano Markidis, Rami Karim, Erwin Laure, and Sai Narasimhamurthy. “Exploring scientific application performance using large scale object storage.” In International Conference on High Performance Computing, pp. 117–130. Springer, Cham, 2018.

7 Lofstead, Jay, Ivo Jimenez, Carlos Maltzahn, Quincey Koziol, John Bent, and Eric Barton. “DAOS and friends: a proposal for an exascale storage system.” In SC’16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 585-596. IEEE, 2016.

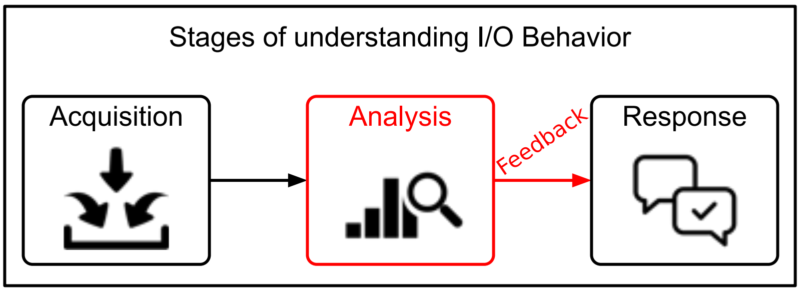


Figure 3 Relationship between data acquisition, analysis, and response as defined in Section 4.1. The focus of this section is highlighted in red. The link that connects analysis and response is referred to as *feedback* in this report.

4.3 Deep Dive Topic: Analysis

Shane Snyder (Argonne National Laboratory (ANL), USA, ssnyder@mcs.anl.gov)
Sarp Oral (Oak Ridge National Laboratory (ORNL), USA, oralhs@ornl.gov)
Suren Byna (Lawrence Berkeley National Laboratory (LBNL), USA, sbyna@lbl.gov)
Philip Carns (Argonne National Laboratory (ANL), USA, carns@mcs.anl.gov)

This section focuses on analysis of I/O behavior and the feedback resulting from that analysis. We define *analysis of I/O behavior* as interpreting and deriving meaning from I/O instrumentation data. We define *feedback* as the product of the analysis process. Feedback can be thought of as the link between analysis and the subsequent response to that analysis, such as I/O tuning or policy decisions (see Section 4.4).

Analysis and feedback may be consumed by either human stakeholders or machine algorithms. If it is to be consumed by human stakeholders, then the analysis process will likely emphasize understandable language and visualization techniques. If it is to be consumed by machine algorithms, then it will likely emphasize the use of machine-parsable, consistent formats that are suitable for ingestion by policy engines and data-mining tools.

4.3.1 State of the Art

Extensive literature exists related to the analysis of the I/O behavior of data-intensive applications, most of which can be broken down into following categories.

4.3.1.1 Existing Machine Learning Analysis Approaches

ML approaches to analysis of I/O behavior have gained a lot of traction over the years. These approaches are popular because they are effective at navigating large volumes of disparate I/O instrumentation data to provide insights into the behavior of complex storage systems. While considerable work has been done in the use of ML for understanding I/O behavior, much of it has been research oriented and therefore does not have many practical use cases at this time.

Xie et al. [1] used ML to help model and predict I/O performance and I/O variability of HPC file systems, specifically focusing on large-scale parallel checkpointing workloads that are common in HPC. Gauge [2] is an interactive I/O analysis tool that uses ML clustering techniques to hierarchically organize large collections of Darshan I/O logs for a facility to identify groups of applications with similar I/O behaviors. Isakov et al. [3] leveraged the

Gauge I/O analysis tool to help identify I/O performance bottlenecks in production and to inform potential application- and system-level improvements. Madireddy et al. [4, 5] utilized ML to develop I/O performance variability models using application I/O characterization data and file system monitoring data. In subsequent studies [6], Madireddy used I/O changepoint detection to identify noticeable changes in the performance of production file systems and used transfer learning to update I/O performance prediction models to account for these changes. Agarwal et al. [7] used active learning techniques to develop models to help optimize MPI-IO library and Lustre file system usage in applications. Wyatt [8] proposed AI4IO, a suite of AI-based tools that can help enable I/O-aware resource scheduling by predicting and mitigating I/O resource contention.

4.3.1.2 Existing Workflow Analysis Systems

Workflows have become a critical abstraction for scientific computing, providing application scientists with frameworks for effectively representing the compute and data dependencies in their campaigns. Little work has been done, however,in analyzing the I/O behavior of these workflows. Thus there is a lack of understanding of the data movement patterns typical of workflows and likely points to untapped optimizations for workflow management systems, job schedulers, I/O libraries, and so on. To address that lack, some preliminary exploratory research has been done into the I/O behavior of workflows and strategies for better understanding them.

Luttgau et al. [9] presented an approach to allow for better understanding of I/O workflow behavior by augmenting workflow systems and I/O analysis tools to be aware of each other. This work outlines challenges in connecting I/O analysis tools with workflow systems and suggests guidance for developers of each to cope with these challenges. Patel et al. [10] analyzed large amounts of Darshan logs from the Cori supercomputer to help identify how files are reused over time and across applications. While not tied to specific workflow systems, this work illustrates how implicit workflows can be identified by mining I/O instrumentation data for data dependencies. Lockwood et al. [11] analyzed numerous I/O monitoring data sources across the entire NERSC data center to gain an understanding of characteristics of data movement in typical scientific workflows and implications for facilities.

4.3.1.3 Visualization and Web Dashboards

Data visualization tools can be critical to application scientists, system admins, and I/O researchers to gain a deeper understanding of I/O behavior on HPC systems. Tools such as Grafana [12] have became increasingly popular for analyzing telemetry data in conventional data centers, but they can also be readily leveraged in HPC contexts. Existing HPC I/O analysis tools can be used to help better understand different types of captured instrumentation data as well. For example, the Darshan [13] I/O characterization tool comes with a tool for generating summary PDF reports that capture highlights about the job's I/O behavior from a corresponding Darshan log. Similarly, tools such as Tau [14] provide analysis capabilities for identifying I/O performance bottlenecks using captured application I/O traces.

A number of more comprehensive I/O visualization and analysis systems have been developed specifically for HPC platforms, such as Altair Mistral and Breeze systems [15], PIKA [16], Beacon [17], Gauge [2], and UMAMI [18]. Altair Mistral and Breeze are I/O profiling and analysis tools that can help characterize and tune application I/O dependencies, file access patterns, and so on. PIKA is a continuous monitoring system designed for use on production HPC systems that contains numerous I/O performance metrics that can be used to characterize I/O behavior of jobs and to identify potential optimization opportunities. Beacon is an end-to-end continuous I/O monitoring system designed for the Sunway TaihuLight supercomputer that comes with corresponding I/O analysis and visualization tools that can be useful to both users and administrators. The Gauge dashboard can be used by system admins and I/O researchers to cluster applications based on similarity of I/O behavior to determine characteristics of different classes of applications that typically run on a given HPC system. UMAMI can provide historical I/O performance context for applications across a range of application- and system-level metrics to help gain insights into how these metrics correlate with general I/O performance over time. While the Altair tools, PIKA, and Beacon have already been successfully demonstrated in production, UMAMI and Gauge are much more research-quality tools at this point.

Prior research studies focused on visualizing I/O behavior could also provide inspiration for the design of I/O analysis tools. For example, Sigovan et al. [19] present some visualization methodologies for enabling better understanding of parallel I/O traces that could be adopted by the tools presented above or by new I/O analysis tools.

4.3.1.4 Technologies Used Outside of HPC

Beyond HPC, a number of I/O analysis tools are available that may not be easily deployed on HPC systems but that could influence the design of HPC-specific I/O analysis tools. For instance, Facebook has developed the HALO [20] system for monitoring and analyzing a number of different hardware-centric metrics on its storage networks. The VMWare I/O analyzer [21] provides a similar tool for measuring various I/O performance metrics and diagnosing issues in virtualized environments. Additionally, Kubernetes Datadog [22] can be used to collect and analyze similar I/O performance metrics in the context of Kubernetes clusters.

4.3.1.5 Technologies Used Outside of I/O

Beyond I/O-related technologies, numerous performance analysis toolkits are available that could be used to motivate new capabilities for HPC I/O analysis tools. One example is Intel VTune [23], a CPU profiling and analysis tool that can optimize application and system performance for different computing environments. Another example is TensorBoard [24], a tool for measuring various performance metrics from TensorFlow applications and for providing detailed visual analysis of the captured metrics.

4.3.1.6 Usage of I/O Signatures

Various research projects have also explored the use of compact representations of I/O workload behavior called *I/O signatures*. Researchers like I/O signatures because they can be used to reproduce or replay I/O workloads without relying on a costly full trace of the application. Byna et al. [25] outlined a set of signature classifications relevant to HPC and used these signatures to help guide application I/O prefetch strategies. Feng et al.[32] presented a tool for capturing I/O signatures, called IOSig+, based on the initial classification of I/O signatures [25]. Behzad et al. [26] similarly used I/O signatures to classify applications at runtime and to help select various I/O tuning parameters based on the observed application I/O signature. Dorier et al. [27] presented Omnisc'IO, an approach that builds grammar-based models of application I/O workloads and uses these models to help predict application I/O behavior.

4.3.1.7 General Studies on Analyzing I/O Behavior

In addition to the research areas covered above, many general studies geared toward analyzing HPC I/O behavior have made important contributions in this space. One notable study is the Charisma project [28], which provided early results for how to best characterize file access patterns of parallel scientific applications. This research enabled many insights that guide not only the design of HPC file systems but also the design of HPC I/O characterization tools such as Darshan. Carns et al. [29] performed an evaluation similar to that of the Charisma project, this time using Darshan to analyze thousands of jobs to identify potential tuning opportunities and systemwide I/O trends at the Argonne Leadership Computing Facility. Luu et al. [30] performed a multiyear, multiplatform I/O study using Darshan logs to help understand the behavior of different types of application I/O workloads over time and across different platforms.

4.3.2 Gaps

I/O pattern/performance analysis is a multivariate, multidimension, complex and dynamic problem. Although state-of-the-art research and practice have made strides in trying to manage this problem, significant shortcomings remain in approach, technology, and capabilities.

Gaps in I/O pattern/performance analysis and feedback can be categorized in four broad areas:

1. Scope
2. Infrastructure
3. Common terminology and communication
4. Limited feedback

4.3.2.1 Scope

Our current I/O pattern/performance analysis and feedback techniques can be viewed horizontally across the I/O software stack or vertically across a file or a storage system. Our current capabilities are, by and large, vertical and focused on a single application and do not take into account the system view. Furthermore, our vertical capabilities mostly target the I/O client side and do not cover the server side of the problem as well. Also, these techniques are mostly file oriented and miss the holistic data view of a given application.

The vertical integration of telemetry data and logs from various layers across the stack, including the client and server sides, the network, and the scheduler, is essential for getting a complete view of how an application's I/O pattern and performance progresses and functions. While this is a crucial capability, the current state of the art is far from it.

The horizontal integration of the I/O pattern/performance telemetry data and logs from all applications running on a given system can provide a systemwide view of how the I/O subsystem is functioning at a given time or how applications are taking advantage of it. Current state of the art has some capabilities in providing a semi-complete picture here, but often a gap remains in correlating the data with other systemwide data streams, such as the network or the scheduler.

With these limited and narrow-scoped glimpses into the complex problem, the analysis or the feedback to consumers (e.g., users, administrators) commonly lacks reference points or context.

Also, a big gap in our arsenal is the lack of capabilities in analyzing workflows. The workflow I/O is becoming more prevalent with emerging edge computing and the coupling of scientific experiments and instruments with HPC data centers and resources.

4.3.2.2 Infrastructure

The I/O pattern/performance analysis and feedback are an acute problem, and most HPC data centers are trying to create solutions to it in an ad hoc manner. While these tackle different aspects of the problem, they are data centered and system specific and not easily portable. This creates a duplication of effort across multiple data centers, wasting scarce and precious resources and being far from effective for answering the community's collective needs.

Also, the limited analysis and feedback capabilities available are tailor-built for solving specific problems and are not versatile enough to be used by both machine and human consumers. With the proliferation of AI techniques, having the capability of currently feeding information to both is becoming more pressing.

4.3.2.3 Common Terminology and Communication

An apparent lack of common language and terminology exists among the three largest I/O stakeholders: scientific users, I/O practitioners (system architects and administrators), and I/O researchers (tool builders).

Often a user raises an issue pointing to a potential problem with the I/O subsystem. The I/O subsystem is usually the canary in the coal mine and can exhibit the first symptoms of a problem anywhere in the system, not necessarily emanating from the I/O subsystem itself. Therefore, trying to understand the full context of the user problem is crucial for conducting a proper root cause analysis, and using a common terminology across the board between the scientific users and the I/O practitioners is essential.

On the flip side, when I/O practitioners identify and point out to the user a pattern or a performance problem with a certain application, the lack of common language and terminology creates a barrier between the user and the I/O practitioner, hindering an expedited and effective solution to the problem.

Also, a lack of common language and a terminology between I/O practitioners and the researchers can create barriers for taking valuable research artifacts and deploying them on production systems or for researchers addressing pressing production I/O problems.

4.3.2.4 Limited Feedback

The feedback provided with existing technology and capabilities falls short of moving the needle in terms of better I/O subsystem usage and utilization at HPC data centers.

A limiting factor here can be the lack of clear demonstration of a cost/benefit analysis of an I/O improvement suggested to a scientific application user. The I/O routines are usually expected to consume less than 10% of the application's total runtime. Some applications are already following good I/O practices, and a negligible improvement in the I/O pattern or performance might not be good use of limited developer resources from the perspective of the scientific application team. In certain cases, however, the benefit of doing the right thing will outweigh the cost of changing the application I/O routines and dramatically increase the application efficiency or scalability. As a community, we need to careful select where we focus our efforts and what kind of feedback we provide (e.g., detail and granularity).

Also, providing comprehensive feedback on the system-level I/O activity, whether as a whole or only for a subset or a kind of application, is challenging. As an example, consider the case where an I/O practitioner needs to build an I/O workload generator mimicking the behavior of a class of small-scale applications generating heavy read requests with random offsets. The challenge here would be identifying what would be a good representative scale (for the size of files or read requests) to build a good analogue for the actual applications.

4.3.3 Challenges

I/O behavior analysis is fundamentally a complex, dynamic, and multivariate problem with no easy or visible solutions. Different facilities have diverse applications, data sources, and use cases. A stark contrast exists between I/O behavior analysis challenges and various success stories from environments with more homogeneous constraints.

Challenges in I/O behavior analysis can be categorized into four broad themes.

1. Technology
2. Complexity
3. Interoperability
4. Resources

4.3.3.1 Technological Impediments to Resolving Gaps

Providing more context and a "bigger picture" of I/O behavior requires very high volume of data from different sources, namely, I/O logs from various subsystems including network. Collecting such high-volume and fine-grained data is challenging for multiple reasons, including the overhead in collecting the data as well as in maintaining the data with indexing. HPC facilities have shown little interest in building storage systems just for handling telemetry and for analyzing feedback that could help understand I/O behavior of applications. Another technical challenge is that AI and other models that can be used to analyze or to predict I/O behavior and performance of applications are not robust enough.

4.3.3.2 Complexity

Various interdependencies among I/O software and hardware layers and heterogeneity of these solutions make understanding I/O behavior extremely complex. In addition, because storage is shared among a large number of users in a system, understanding of performance is often hindered by variability. Since variability is unpredictable or not well understood, trust in the I/O performance prediction and analysis feedback is typically low. (More discussion of variability is available in §4.1.) Moreover, some facilities have a diverse set of application programming models that exacerbate the challenge of finding common analysis.

4.3.3.3 Interoperability of Analysis Tools

A significant incompatibility exists among telemetry collected at data sources (§4.2) and I/O analysis tools (§4.1) that is leading to poor interoperability of analysis tools. Since no standards exist for what telemetry to collect, different facilities generate logs at different frequencies, coverage, diverse sets of details, and so on. Disparity in the use of schedulers also exists. For instance, an analysis tool requiring details of all the jobs running concurrently when an application's I/O was occurring may not have access to similar details in the output of different schedulers. As a result, an analysis tool that requires data in a specific format becomes specific only to the data available to it. This challenge warrants the need for native telemetry formats across multiple data source components across multiple facilities. Another challenge is with sharing telemetry data openly for various analyses. For instance, facility security enclaves sometimes inhibit data sharing. In this case, analysis tools have to be developed in collaboration with the facilities, and requirements from these facilities may become too customized, thus challenging the goals of analysis tool interoperability.

4.3.3.4 Resources

In some cases, although analysis tools and tuning options are well established, low engagement from users in tuning I/O (§4.6) discourages facilities from justifying the effort needed to push the frontier of I/O behavior analysis. For example, using collective buffering in MPI-IO is a well-known strategy when there are a large number of small I/O requests from each MPI rank. However, identifying the poor performance caused by this inefficient use of I/O software either has to be initiated by users or has to be identified automatically by a systemwide monitoring tool. Triggering a tuning effort and contacting facility support for tuning may need first to train users. Facilities may have to provide resources for these training events as well as expertise for solving storage and I/O issues. All these efforts require resources from facilities, and facilities often struggle to justify such efforts with low demand for tuning I/O.

4.3.4 Next Steps and Recommendations

In our analysis of I/O behavior we identified the following recommendations and next steps for the community based on our survey of gaps and challenges .

4.3.4.1 Technical

Developers of analysis tools and methodologies should plan for interoperability up front. Although extensive work exists in this space, projects that start off as independent efforts seldom arrive at an integrated state organically. Required instead is a conscious design effort to be inclusive of modular and interoperable tools. Community organizations or funding agencies could help spur effort in this direction, but the responsibility ultimately lies with researchers and developers. For example, a developer of a new methodology or tool for analysing I/O behavior in workflows could begin by considering how it would fit into a broader ecosystem of platform-level or job-level analysis tools. If the tools are cognizant of each other up front, then design considerations can be taken to allow them to flow together as part of a unified tool chain.

Generalizabillity of I/O models should be considered a first-class problem. An I/O model could be developed by using analytical methods, artificial intelligence, or discrete event simulation. Regardless of the technique employed, little evidence indicates that they have been trusted for use in production environments in practice, despite the promise of potential to guide performance enhancement. The underlying cause is a lack of trust by potential consumers of those models and, in many cases, a lack of effort by model developers to try to win that trust. This disconnect could be addressed with emphasis on how to reason about the generalizability and correctness of a model. Initial validation and reproducibility efforts are valuable, but insufficient. To act on this recommendation, the community must go a step further and pursue continuous validation and refinement methods as well.

4.3.4.2 Social

I/O analysis researchers and practitioners should gather and document lessons and requirements from stakeholders in the field. The purpose of this recommendation is to ensure that effort is expended on the right problems and that feedback from analysis tools is formatted appropriately for consumers. These lessons and requirements could take several forms. The most valuable would be broad surveys shared with the community to help inform overall activities. More narrowly scoped research efforts could also benefit from more limited focus group discussion, however. Without this key step, there is a danger that an analysis tool could distill unwanted information or could distill information into a form that inadvertently causes an educational bottleneck by requiring too much effort on the part of consumers to use.

Community standards should be developed for common analysis tasks. Analysis tools could be made more portable and generalizable by consuming input data in standardized formats, but that topic is more appropriately covered in Section 4.2. For analysis methods themselves, the standards that are lacking are less concrete. What is needed is a better community agreement on terminology and taxonomies of I/O motifs to help structure feedback and frame discussions. The current state of the practice is that different tools employ their own jargon or overload existing jargon in ways peculiar to their analysis task. This sort of standardization may be best handled by organizations such as VI4IO [31] that span research teams, organizations, and countries. Standardization bodies that address broad issues such as these must take care that standardization efforts do not unintentionally inhibit research innovation, however.

References

1 B. Xie et al. *Applying machine learning to understand write performance of large-scale parallel filesystems.* In 2019 IEEE/ACM Fourth International Parallel Data Systems Workshop (PDSW), pp. 30–39. 2019.

2 E. Del Rosario et al. *Gauge: An interactive data-driven visualization tool for HPC application I/O performance analysis.* In 2020 IEEE/ACM Fifth International Parallel Data Systems Workshop (PDSW), pp. 15-21. 2020.

3 M. Isakov et al. *HPC I/O throughput bottleneck analysis with explainable local models.* In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–13. 2020.

4 S. Madireddy et al. *Analysis and correlation of application I/O performance and system-wide I/O activity.* In 2017 International Conference on Networking, Architecture, and Storage (NAS), pp. 1–10. 2017.

5 S. Madireddy et al. *Machine learning based parallel I/O predictive modeling: A case study on Lustre file systems.* In International Conference on High Performance Computing, pp. 184-204. 2018.

6 S. Madireddy et al. *Adaptive learning for concept drift in application performance modeling.* In Proceedings of the 48th International Conference on Parallel Processing, pp. 1–11. 2019.

7 M. Agarwal et al. *Active learning-based automatic tuning and prediction of parallel I/O performance.* In 2019 IEEE/ACM Fourth International Parallel Data Systems Workshop (PDSW), pp. 20-29. 2019.

8 M. Wyatt. *AI4IO: A suite of AI-based tools for IO-aware HPC resource management.* University of Delaware. 2020.

9 J. Luttgau et al. *Toward understanding I/O behavior in HPC workflows.* In 2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage and Data Intensive Scalable Computing Systems (PDSW-DISCS), pp. 64–75. 2018.

10 T. Patel et al. *Uncovering access, reuse, and sharing characteristics of I/O-intensive files on large-scale production HPC systems.* In 18th USENIX Conference on File and Storage Technologies (FAST 20), pp. 91–101. 2020.

11 G. Lockwood et al. *Understanding data motion in the modern HPC data center.* In 2019 IEEE/ACM Fourth International Parallel Data Systems Workshop (PDSW), pp. 74–83. 2019.

12 *Grafana: The open observability platform.* https://grafana.com/.

13 *Darshan I/O characterization tool.* https://www.mcs.anl.gov/research/projects/darshan/.

14 *TAU: Tuning and analysis utilities.* http://tau.uoregon.edu/.

15 *Altair Mistral: Live system telemetry and I/O monitoring.* https://www.altair.com/mistral/.

16 R. Dietrich et al. *PIKA: Center-wide and job-aware cluster monitoring.* In 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 424–432. 2020.

17 B. Yang et al. *End-to-end I/O monitoring on a leading supercomputer.* In 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19), pp. 379-394. 2019.

18 G. Lockwood et al. *UMAMI: A recipe for generating meaningful metrics through holistic I/O performance analysis.* In Proceedings of the 2nd Joint International Workshop on Parallel Data Storage and Data Intensive Scalable Computing Systems, pp. 55-60. 2017.

19 C. Sigovan et al. *A visual network analysis method for large-scale parallel I/O systems.* In 2013 IEEE 27th International Symposium on Parallel and Distributed Processing, pp. 308–319. 2013.

20 *Facebook HALO (Hardware Analytics and Lifecycle Optimization).* https://engineering.fb.com/2017/03/21/data-center-engineering/hardware-analytics-and-lifecycle-optimization-halo-at-facebook/.

21 *VMware I/O analyzer.* https://flings.vmware.com/i-o-analyzer.

22 *Monitoring Kubernetes with Datadog.* https://www.datadoghq.com/blog/monitoring-kubernetes-with-datadog/.

23 *Intel VTune Profiler.* https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/vtune-profiler.html.

24 *TensorBoard: TensorFlow's visualization toolkit.* https://www.tensorflow.org/tensorboard.

25 S. Byna et al. *Parallel I/O prefetching using MPI file caching and I/O signatures.* In SC'08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, pp. 1–12. 2008.

26 B. Behzad et al. *Pattern-driven parallel I/O tuning.* In Proceedings of the 10th Parallel Data Storage Workshop, pp. 43-48. 2015.

27 M. Dorier et al. *Omnisc'IO: A grammar-based approach to spatial and temporal I/O patterns prediction.* In SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 623-634. 2014.

28 N. Nieuwejaar et al. *File-access characteristics of parallel scientific workloads.* IEEE Transactions on Parallel and Distributed Systems 7, no. 10 (1996): 1075-1089. 1996.

29 P. Carns et al. *Understanding and improving computational science storage access through continuous characterization.* ACM Transactions on Storage (TOS) 7, no. 3 (2011): 1-26. 2011.

30 H. Luu et al. *A multiplatform study of I/O behavior on petascale supercomputers.* In Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing, pp. 33–44. 2015.

31 The Virtual Institute for I/O. https://www.vi4io.org/.

32 Feng, Bo, Xi Yang, Kun Feng, Yanlong Yin, and Xian-He Sun. *IOSIG+: on the Role of I/O Tracing and Analysis for Hadoop Systems.* In IEEE International Conference on Cluster Computing (CLUSTER), IEEE, pp. 62–65, 2015.

4.4 Deep Dive Topic: Enacting Actionable Responses

Andreas Knüpfer (Technische Universität Dresden, DE, andreas.knuepfer@tu-dresden.de)
Julian M. Kunkel (Universität Göttingen / GWDG, DE, julian.kunkel@gwdg.de)
Erwin Laure (Max Planck Computing and Data Facility, DE, erwin.laure@mpcdf.mpg.de)
Radita Liem (RWTH Aachen University, DE, liem@itc.rwth-aachen.de)
Frank Mueller (North Carolina State University, USA, mueller@cs.ncsu.edu)

This section describes how the output from analysis tools, current and imagined, could be utilized to provide more efficient execution of applications and workflows through optimization of I/O across systems and data centers. It identifies current state-of-the-art practices and proposes actionable approaches based on emerging I/O monitoring and analysis tools and I/O-related technologies.

4.4.1 State of the Art

4.4.1.1 Applications

Currently, users are expected to perform their own I/O optimization on their applications. This approach works to a degree with expert users but takes a sizable part of developer time. Inexperienced users often find it nearly impossible to perform any I/O optimization, since this would require deep knowledge about the underlying file systems. Particularly at smaller HPC sites with a sometimes less knowledgeable (in regard to I/O) community, actions cannot always be taken on a per-application level. Nevertheless, high-level libraries such as netCDF and HDF5 are a good starting point for any level of sophistication, since they are ubiquitously used in many HPC applications.

4.4.1.2 Workflow

On most systems, the state of the art for data management and workflows requires that users move their data around manually. Many scientific domains have their preferred workflow managers, but most of these either are not built for HPC-systems or have limited options for data locality. Data is taken from one or more specified files and written back to the specified output file. More sophisticated options such as staging into local storage or using a burst buffer need to be defined by the user and are often not built into the manager itself. While more I/O-optimized workflow managers may exist, domain scientists naturally tend to use the preferred options in their domain, which are often less HPC friendly. Even in batch schedulers, the options to define an I/O-aware workflow or even use data staging are limited.

4.4.1.3 System

Procurement and acceptance of storage systems as well as fine-tuning of file system parameters are often based on artificial benchmarks or single applications that may not be representative of real system usage.

Users have only limited insight into the status of the file systems and are usually not aware of current bottlenecks or performance degradation. Consequently, they do not understand the root cause of poor application performance and cannot identify suboptimal usage patterns in their jobs. Furthermore, they have no chance to schedule future highly demanding I/O operations regarding current system load.

4.4.1.4 Data Center

Batch schedulers for HPC data centers typically do not explicitly manage I/O resources. In contrast to compute requirements, users are not required to define I/O bandwidth or latency requirements when submitting a job. Therefore, data-intensive phases of several applications can occur at the same time, leading to significant congestion of the storage system, which decreases both the overall performance of the storage system and the I/O bandwidth seen by the individual applications.

A number of approaches do exist that could make storage an actively manageable resource in HPC data centers. On the one hand, file systems such as Lustre provide quality of service management on the level of metadata operations and I/O operations [7]. On the other hand, extensions to batch environments, such as NORNS, facilitate asynchronous staging of job input data between different storage tiers [5]. Also, many applications are, in principle, prepared to move data-intensive phases to optimize overall storage usage. Checkpointing environments such as SCR provide feedback mechanisms that allow the application to delay the execution of a checkpoint in the event that the system is currently oversubscribed [6]. Nevertheless, these approaches currently require that either the user or the administrator understand the storage requirements of applications, while tools to automatically derive these requirements and develop appropriate actions based on them (e.g., automatic staging of data between different storage tiers) are still in their infancy [8].

Most HPC data centers also do not have a consistent set of rules and guidelines on how to deal with I/O or I/O-based problems. Instead, they often use self-developed solutions to issue alerts when systems are overloaded. Furthermore, users have difficulty identifying I/O problems (beyond experiencing long application runtimes), since most data centers do not provide reports on the I/O usage of individual applications or the congestion conditions of the file system as a whole.

4.4.2 Gaps

4.4.2.1 Applications

Clearly lacking is easily digestible feedback from monitoring and analysis tools that can guide users in optimizing the I/O characteristics of their applications. Besides better tool support, more and better training and spreading knowledge about readily available I/O libraries would be beneficial.

4.4.2.2 Workflow

The diversity in the current workflow management systems makes it difficult to analyze and therefore hard to act on. Any attempts to define rules for a workflow manager based on a particular I/O analysis work only for that specific case. The lack of standardization limits the definition of any actionable steps to a narrow scope.

4.4.2.3 System

Synthetic benchmarks are missing that are able to emulate actual I/O workloads in real-world production use. Moreover, user-accessible and machine-readable monitoring and file system status information is lacking.

4.4.2.4 Data Center

Currently no consistent set of rules or guides exist on how to deal with I/O or I/O-based problems at a data center level. Typically what is used are home-brewed solutions that provide alerts if systems become overloaded.

4.4.3 Challenges

4.4.3.1 Applications

Understanding an application's I/O behavior and potentially detrimental data access patterns is not trivial and requires significant experience. Automated tools and access to monitoring data to support user insight in this area are lacking.

4.4.3.2 Workflow

Workflows are not well defined, and hence ensuring efficient I/O at an aggregate or even subcomponent level in a consistent and programmatic manner is currently difficult or even impossible depending on the complexity of the workflows being addressed.

4.4.3.3 System

Creating benchmarks that are both representative of production workloads and reproducible is difficult. They would have to be based on historical monitoring data and job traces.

4.4.3.4 Data Center

The management of I/O resources requires that users and administrators be well informed about the use of the underlying storage systems. An important gap is that users (and most administrators) generally do not have access to I/O usage statistics. Providing access to this type of data through text reports and I/O visualization and browsing tools would take significant work but could provide great benefit in terms of more efficient utilization of file systems.

New approaches to manage storage resources are already well developed within the storage research domain and are also implemented in widespread storage systems and parallel file systems. Nevertheless, their active application in data centers is typically delayed by several years. One example is the availability of on-demand file systems, which can build parallel file systems on top of very fast node-internal storage resources [1, 9]. These ad hoc file systems can be used to isolate I/O accesses across applications and from joint parallel file systems [10].

4.4.4 Next Steps and Recommendations

4.4.4.1 Applications

Focus should be put on high-level libraries such as netCDF and file formats such as HDF5. Feedback from analytics tools, such as Darshan, Tau, and Score-P [2, 4, 3], should be utilized to automatically optimize these in the context of particular applications. Such actions could be taken either on a per-application level or on a file system level: Optimize the library specifically for each application or to work best on average for the underlying file system.

4.4.4.2 Workflow

The current practice of users manually orchestrating their own data movement is inefficient because they cannot know the state of the system and how to optimize for current or future conditions. Data orchestration tools exist that users may be unaware of and that could be utilized to provide more efficient data management. By promoting these tools for use by users, data management across the system could be performed more efficiently, resulting in overall efficiency gains. In order to enable automated I/O optimization on the level of workflow management systems that can apply to many systems at once, guidelines on I/O integration should be designed. These guidelines could act as a loose standard for orchestration approaches, which could be used across the majority of workflow engines to further enhance overall data movement across applications on a system.

4.4.4.3 System

As mentioned, file systems are currently being tendered and procured on the basis of synthetic benchmarks. It would make more sense if these systems were benchmarked against I/O patterns from real applications, as is done for the computing part. Therefore, we suggest utilizing access patterns learned from real applications to tune/define PFS parameters and future designs. We note that the current parallel filesystem implementations may affect those access pattern characteristics and that this possibility needs to be taken into account.

When using the file system of an HPC system, avoiding congestion is preferable. Here available solutions should be used, for example, using quality of service for file systems [7] or extending schedulers to support data-driven workload [5]. With such an API for feedback, applications can use it to find the optimal time for I/O heavy operations such as checkpointing. Users should take into account, however, that such feedback can cause problems if too many I/O heavy jobs follow these signals.

Two steps should be taken here. First, real I/O workloads should be collected in order to plan and design new file systems on the basis of these. Second, some means of controlling I/O traffic should be created.

4.4.4.4 Data Center

The next steps to be performed on a data-center level can be derived from both the state of the art and the identified gaps.

- In a first step, users and administrators must have access to I/O usage statistics, in the form of a visual and intuitive “I/O Weather” report or as an extended text report. The report results should become comparable between data centers by driving the standardization of data/metrics format. Also, users have to be actively informed about bad I/O scores, if such a score is available.
- File system feature and performance variability study outcomes should be used to offer users hints as to what features might provide best performance for their workloads.

- Data centers have to introduce incentives to optimize I/O usage, so that users are encouraged to optimize their I/O usage. These incentives can be based on a gamification approach, where users, for example, can get recognized for their improvements or can get credit, for example, in the form of additional CPU-hours.
- An efficient usage of I/O resources can be achieved only if solutions such as quality of service, I/O management through the batch system, or ad hoc file systems are made available in data centers.
- Load should be (automatically) shifted according to "I/O Weather" between storage systems.

References

1 André Brinkmann, Kathryn Mohror, Weikuan Yu, Philip H. Carns, Toni Cortes, Scott Klasky, Alberto Miranda, Franz-Josef Pfreundt, Robert B. Ross, Marc-Andre Vef. *Ad Hoc File Systems for High-Performance Computing.* J. Comput. Sci. Technol. 35(1): 4–26 (2020)

2 Philip H. Carns, Kevin Harms, William E. Allcock, Charles Bacon, Samuel Lang, Robert Latham, Robert B. Ross. *Understanding and Improving Computational Science Storage Access through Continuous Characterization.* ACM Trans. Storage 7(3): 8:1–8:26 (2011)

3 Andreas Knüpfer, Christian Rössel, Dieter an Mey, Scott Biersdorff, Kai Diethelm, Dominic Eschweiler, Markus Geimer, Michael Gerndt, Daniel Lorenz, Allen D. Malony, Wolfgang E. Nagel, Yury Oleynik, Peter Philippen, Pavel Saviankou, Dirk Schmidl, Sameer Shende, Ronny Tschüter, Michael Wagner, Bert Wesarg, Felix Wolf. *Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir.* In Parallel Tools Workshop 2011: 79–91

4 Sameer S. Shende, Allen Malony *The TAU Parallel Performance System.* The International Journal of High Performance Computing Applications 20(2) (2006): 287–311.

5 Alberto Miranda, Adrian Jackson, Tommaso Tocci, Iakovos Panourgias, Ramon Nou. *NORNS: Extending Slurm to Support Data-Driven Workflows through Asynchronous Data Staging.*. In IEEE International Conference on Cluster Computing (CLUSTER), Albuquerque, NM, USA, September 23–26, 2019

6 Kathryn Mohror, Adam Moody, Bronis R. de Supinski. *Asynchronous Checkpoint Migration with MRNet in the Scalable Checkpoint / Restart Library*. In IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN), Boston, MA, USA, June 25–28, 2012

7 Yingjin Qian, Xi Li, Shuichi Ihara, Lingfang Zeng, Jürgen Kaiser, Tim Süß, André Brinkmann. *A Configurable rule Based Classful Token Bucket Filter Network Request Scheduler for the Lustre File System.* In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Denver, CO, USA, November 12–17, 2017

8 Yingjin Qian, Xi Li, Shuichi Ihara, Andreas Dilger, Carlos Thomaz, Shilong Wang, Wen Cheng, Chunyan Li, Lingfang Zeng, Fang Wang, Dan Feng, Tim Süß, André Brinkmann. *LPCC: Hierarchical Persistent Client Caching for Lustre.* In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Denver, CO, USA, November 17–19, 2019

9 Sebastian Oeste, Marc-André Vef, Mehmet Soysal, Wolfgang E. Nagel, André Brinkmann, Achim Streit. *ADA-FS – Advanced Data Placement via Ad hoc File Systems at Extreme Scales.* Software for Exascale Computing 2020: 29–59

10 Mehmet Soysal, Marco Berghoff, Thorsten Zirwes, Marc-André Vef, Sebastian Oeste, André Brinkmann, Wolfgang E. Nagel, Achim Streit, *Using On-Demand File Systems in HPC Environments.* In 2019 International Conference on High Performance Computing & Simulation (HPCS), 2019, pp. 390–398.

4.5 Deep Dive Topic: Data Center Support

Andreas Knüpfer (Technische Universität Dresden, DE, andreas.knuepfer@tu-dresden.de)
Julian M. Kunkel (Universität Göttingen / GWDG, DE, julian.kunkel@gwdg.de)
Erwin Laure (Max Planck Computing and Data Facility, DE, erwin.laure@mpcdf.mpg.de)
Radita Liem (RWTH Aachen University, DE, liem@itc.rwth-aachen.de)
Frank Mueller (North Carolina State University, USA, mueller@cs.ncsu.edu)

4.5.1 State of the Art

The responsibilities of a data center with regard to I/O aspects are focused on a set of broad areas. For each area we identify the current state of data center support by color, where green denotes adequate measures/tools/strategies are already researched and mostly in place on the data center side; yellow means support is improvable: that is, on most data centers some aspect is missing that should be improved; orange means insufficient support; and red means largely absent support.

The broad areas of responsibilities related to I/O for data centers are as follows:

1. Procurement
   - Storage systems
   - Network
   - Standardized benchmarking
   - Monitoring software
2. Operating the system
   - Maintenance
   - I/O monitoring
   - Accounting Job-specific monitoring)
   - Hardware/software tuning
   - Policy enforcement
   - User-specific dedicated resource allocation
   - Data security and safety (isolating users/groups, access permissions, backups)
3. Supporting the users
   - Giving feedback from monitoring – Showing the I/O weather
   - Providing joint code optimizations (brainware missing)
   - Tuning hardware/software parameters for applications
   - Understanding what users are doing and really need
4. Training
   - Topics: Policies, hardware/software stack, selected high-level APIs, Workflow systems, best practices, typical mistakes, tools
   - Documentation
   - Tutorials
   - Creation of reasonable user expectations
   - Development of I/O components
   - User acceptance of training
5. Research (leads to training)

- Understanding hardware/software behavior
- Tuning, optimizing codes
- Measurement and monitoring including tools → sustaining development

6. Policy making
   - Application for resources
   - QoS
   - Resource assignment (e.g., quota) => static vs. Dynamic
7. Long-term strategic planning of all the above
   - I/O demands, e.g., capacity planning
   - Technology selection / evaluation (hardware/software) / vendor discussions
8. Community support
   - Site users
   - Contributions to standardization (arguably not necessarily a responsibility)

While we only briefly touch on some issues colored green and yellow (slightly improvable), we particularly elaborate on the orange (insufficient) and red (largely absent) focus areas to identify gaps and inherent challenges before making recommendations for them.

4.5.2 Procurement

The procurement of **storage systems** including necessary **network infrastructures** is a core responsibility of data centers. They usually have good relationships with vendors, know the state-of-the-art technologies, and are proficient in specifying the requirements for the hardware components. However, the specification of **benchmarks** for procurements has room for improvement. While the IO500 is a huge benefit for establishing performance expectations, application-specific patterns still are lacking. This situation applies to both standardized I/O benchmarks that mimic typical applications (in particular w.r.t. I/O) and mapping a data center's application mix to a representative set of (standardized) benchmarks.

Monitoring software is often considered as part of procurements. Yet it is mostly basic monitoring in a proprietary manner. It usually does not meet the required level of detail, may be difficult to integrate into an analysis stack, or may miss certain relevant subcomponents of the cluster. When moving from one system to the next, it is often not compatible, specific measurement values are not 1:1 comparable, and the expertise built with the past system is not applicable to the next one. More standardization in well-defined measurements and in software components is desirable.

4.5.3 Operating the Systems

In HPC operations many well-established I/O aspects exist, such as **maintenance**, **enforcements of policies** (esp. quotas, data lifecycle, etc.), **accounting** including I/O resources, and **data security and safety** measures, as well as means for **dedicated resource allocations** for user-specific purposes.

4.5.3.1 I/O Monitoring

Even though most data centers have implemented a global monitoring system, I/O monitoring is often insufficient to help optimize the system as a whole as well as particular workloads.

A particular problem is the amount of data generated by monitoring tools. We identified the need for scalable tools to store and analyze monitoring logs. The analysis results need to be better understandable by both admins and users (e.g., by providing some kind of *I/O Weather* status/forecast). Such analysis should identify actionable items (what to do when), which should be fed back to the users and administrators. While the monitoring of the system and its workflows involves obtaining data from different jobs (user) and I/O servers (system), a holistic view is missing for both a workflow-centric manner (across jobs) and a center-wide manner (I/O resource status). Tool support is lacking for identifying misconfigurations, indicating performance regressions, and predicting failures.

To mitigate these problems, we suggest standardizing the format of monitoring data (including possibilities for compression), establishing a common terminology for I/O monitoring, creating a holistic view of different monitoring sources and dimensions (users and systems), and investigating the potential benefit of AI/ML methods to further the understanding of monitoring data for administrators, developers, and users.

4.5.3.2 Job-Specific Monitoring

Current accounting practices often lack detailed information about the dynamics of data access (frequency, bandwidth, behavior patterns).

In order to address this problem, I/O accesses should always be monitored also on a per-job level, or the global monitoring results should be mapped to jobs, workflows, users, or projects. Excerpts or coarse-grained *footprints* should be incorporated into accounting (albeit the delimination between the terms *accounting* and *job monitoring* remains fuzzy) in terms of key performance indicators such as metadata operation rates, transfer rates, and IOPS. Moreover, tuning potentials with estimated benefits should be indicated.

Furthermore, accounting should “feel” similar across different data centers to avoid user confusion. Reporting metrics thus should be carefully selected, standardized, and unified in terms of terminology while considering their impact on steering I/O performance.

4.5.3.3 Hardware and Software Tuning

Hardware and software tuning is focused on both the system side and the application side. One challenge here is that the system can affect an application’s performance and vice versa. In general, the effect of tuning typically cannot be predicted: tuning attempts therefore need to be implemented before their benefit will be known.

Another problem is a lack of clear knowledge about all available tuning options that exist at applications, runtime, and system and hardware levels. Some tuning knobs may require administration rights or even reboots to change BIOS/hardware settings.

Approaches to mitigate these problems include performance modeling and analysis of historic information of the system. Of further importance is the identification of *all* relevant tuning options. This includes static tuning at the center/application levels with optional dynamic tuning at the user/runtime level. When considering tuning options, simulation-based prediction tools can help but often fall short of capturing a sufficiently detailed hardware model.

4.5.4 Supporting the Users

Joint code optimization on a system/infrastructure level and on the application level remains a challenge, mostly because of a lack of experts.

Unilateral **tuning of hardware/software parameters** remains a general challenge, because on a system level they always need to accommodate the average workload. Research into more dynamic on-demand adaptation of system/infrastructure parameters would be desirable.

Improved **feedback from monitoring** would help better aim at the optimization goal. By improved we mean a holistic consideration of all resources, with more detail and enriched with hints that indicate potential issues (first guesses what the issue is and how this can be improved). On the one hand, this would improve job-specific optimization for a single application or all jobs of a particular kind. On the other hand, a notion of the current *I/O Weather* would be helpful, namely, general information about the state or the stress of the shared I/O subsystem. This in turn would be a key criterion for reasonable expectations of how high the I/O performance of a given application could become. It could also be used to dynamically activate or deactivate certain I/O optimizations in an application.

4.5.4.1 Understanding What Users Are Doing vs. What They Really Need

Some users have limited understanding of their application's actual I/O behavior and their needs. They act only if their problem turns out to be a grave performance problem to them or others. These users accept suboptimal performance and ignore the potential for improvement. Also, it is difficult to assess which applications have good vs. bad performance, or even what performance to expect.

Some experimental tools uncovering problematic application behavior are available and should be used more by data center experts. However, they can "only" uncover the problematic effects to the shared I/O system; and, since we will likely never have a full formal specification of I/O behavior, they will remain intrinsically limited. Thus, structured discussions with users and developers is needed, since domain-specific knowledge is required to fully understand I/O needs. Both approaches, the use of tools and structured dialogues with users and developers, require dedicated HPC experts for consultation and support and active help in application development and usage.

This issue is even more acute for data workflows (multiple jobs or steps with multiple applications as part of a processing chain) or data lifecycle decisions that may even span multiple sites.

4.5.5 Training

Training on I/O, particularly advanced training, is currently not sufficiently offered in the various training programs. Limited training exists for proper data lifecycle and workflow handling. Another problem is user acceptance of training, which is also lacking to some extent.

We suggest establishing coordinated training programs with a consensus on what should be taught. We also suggest creating a simple I/O performance model (I/O roofline model) to train best- and worst-case practices by listing pitfalls in application, domains, and I/O patterns. Furthermore, training material should be shared. Training should be offered for multiple levels of expertise (fine-grained). Also, better documentation on the topics and common knowledge (best practices, patterns) should be available and communicated.

We could solve the problems by devoting more resources to develop training. One could submit a paper to a workshop about the state of training practice or could join forces with an existing training program such as VI-HPS [1] and the HPC Certification Forum (HPC CF)[2].

[2] https://www.hpc-certification.org/

4.5.6 Research

A number of I/O research directions exist as part of HPC research. Some are more focused on I/O; some cover I/O as one aspect among others. The research has been roughly classified into **understanding hardware/software behavior**, **tuning and code optimization**, and **measurement and monitoring, including tools**.

4.5.6.1 Sustaining Development of Monitoring Systems and Tools

Many valuable tools and software components have been created by the research community. Yet follow-up activities for the projects, tools, and libraries are problematic in terms of sustained funding. Certainly, this should not become the default case for all research projects. Yet few I/O tools and libraries exist, and many fewer with an I/O focus subject to some level of production-level code quality and sustained development. Darshan and SIONlib are two of the notable exceptions.

Reasons for this situation are more or less all connected to sustained funding. While commercialization may be an option for some, in general open-source community-driven development and support models might be more suitable for the existing community. Follow-up research funding will certainly help as long as additional research questions remain. However, research funding by government or academic funding sources is looking at different goals and cannot be a long-term answer here.

Instead, the academic, government, and industry players in the HPC community should address this situation. Data centers should play a role and try to secure funding for some tools, software components, or projects that are in everybody's interest to be sustained. They should argue to their funding agencies that this is a fair return of effort to selected projects compared with the direct benefit the center receives from the much larger amount of open-source and community software used. This should go beyond the current opportunities. For example, in the United States, Small Business Innovation Research and Small Business Technology Transfer programs [2] exist within NSF and DOE, and also other NSF programs especially from the CISE Office of Advanced Cyberinfrastructure and similar programs in other countries.

4.5.7 Policy Making

Policy making covers the **application for resources**, **quality of service**, and resource assignment. Existing policies focus mainly on restricting user capacity and number of files (quota mechanisms) and some purging policies of scratch file systems. Isolating users and applications that negatively influence the storage performance of the shared storage can be improved on. Defining appropriate policies and having mechanisms to do so would be an evolutionary step from the current situation. File systems and networking need to be improved to better support this option. Similarly, when users apply for resources, the data centers should be more rigid in requesting information about the I/O usage and workflows. Some data centers request more details, but others limit themselves to the storage capacity needed. The resource assignment of projects is mostly statically assigned; in other words, with a project proposal one receives an allocation for the project campaign. In practice, however, the requirements change over the lifetime of a project. Therefore, a more **dynamic resource assignment** would be useful, but this is not yet available or implemented by the centers.

4.5.8 Long-Term Strategic Planning

Long-term strategic planning is sought by all data centers, and some strategies that have been been developed have been successfully applied. Improvements are still needed, however, particularly in better planning of the I/O storage landscape, selecting and evaluating storage systems, and engaging with vendors. All these points require a better understanding of storage systems and the implications specific design decisions would have.

4.5.9 Community Support

Community support could be improved for data center users, for example, transforming the vertical communication between users and the center to a more vibrant one where users communicate directly with one another (horizontally) and share best practices. This goal requires platforms for discussion and information exchange. Arguably, the contribution to standardization bodies and technical solutions are not a key responsibility of a data center per se; nevertheless, providing service to users and operating the systems effectively clearly support the data center mission.

Additionally, synchronization of training and documentation is desirable, for example, establishing common terminology and training materials. Particularly useful would be developing training materials that can be used across data centers with little adaptation for site specifics. Another form of knowledge sharing is the data repository, which collects and manages datasets for analysis and sharing. An excellent example is the Darshan log repository provided by Argonne National Laboratory [3].

The community aspects are discussed in more detail in the next chapter.

References

1 https://www.vi-hps.org/
2 https://www.sbir.gov/
3 https://www.mcs.anl.gov/research/projects/darshan/download/

4.6 Deep Dive Topic: Community Support

Wolfgang Frings (Jülich Supercomputing Centre, Germany, w.frings@fz-juelich.de),
Yi Ju (Max Planck Computing and Data Facility, Germany, yi.ju@mpcdf.mpg.de),
Sarah Neuwirth (Goethe-University Frankfurt, Germany, s.neuwirth@em.uni-frankfurt.de),
Sebastian Oeste (TU Dresden, Germany, sebastian.oeste@tu-dresden.de),
Martin Schulz (TU Munich, Germany, martin.w.j.schulz@tum.de)

While many community aspects identified in this seminar match the general challenges faced throughout the entire HPC community (or even beyond), the I/O community faces several specific issues in addition. The cause is rooted in the special characteristics found in the I/O community, which are listed in the following subsection. The rest of the section then focuses on consequences from these special characteristics and how to merge them with other, general community problems.

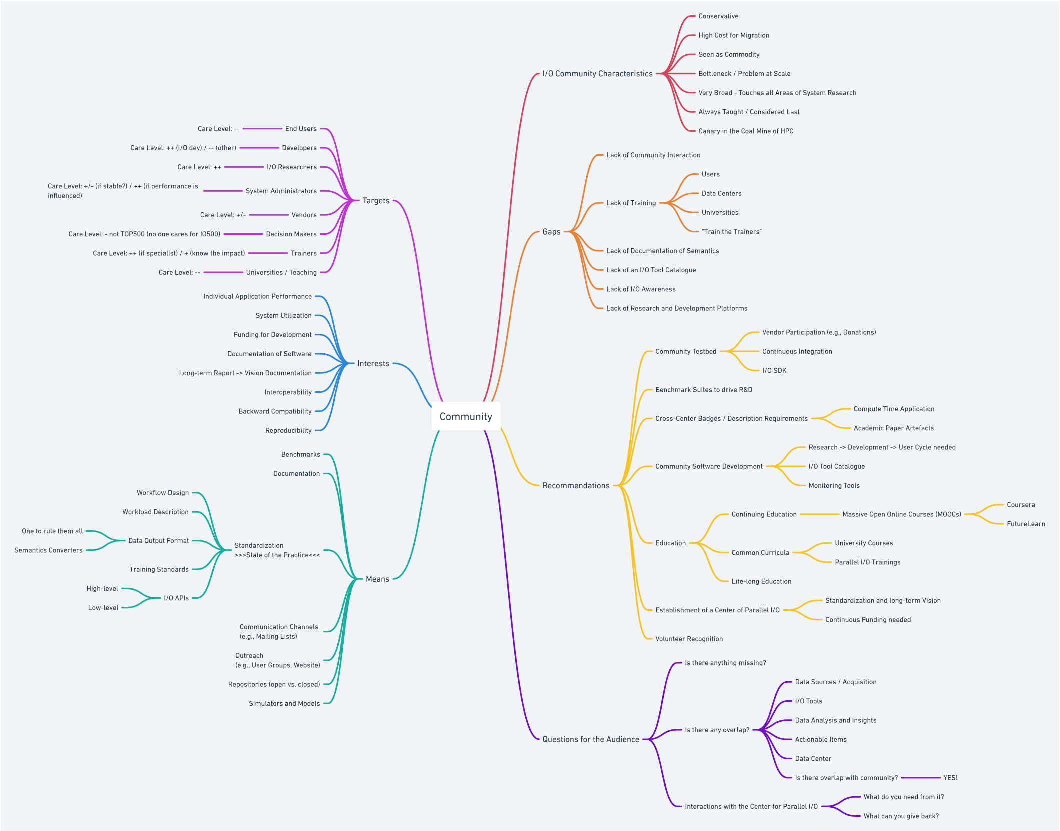

Community
I/O Community Characteristics
Conservative
High Cost for Migration
Seen as Commodity
Bottleneck / Problem at Scale
Very Broad - Touches all Areas of System Research
Always Taught / Considered Last
Canary in the Coal Mine of HPC
Gaps
Lack of Community Interaction
Lack of Training
Users
Data Centers
Universities
"Train the Trainers"
Lack of Documentation of Semantics
Lack of an I/O Tool Catalogue
Lack of I/O Awareness
Lack of Research and Development Platforms
Recommendations
Community Testbed
Vendor Participation (e.g., Donations)
Continuous Integration
I/O SDK
Benchmark Suites to drive R&D
Cross-Center Badges / Description Requirements
Compute Time Application
Academic Paper Artefacts
Community Software Development
Research -> Development -> User Cycle needed
I/O Tool Catalogue
Monitoring Tools
Education
Continuing Education
Massive Open Online Courses (MOOCs)
Coursera
FutureLearn
Common Curricula
University Courses
Parallel I/O Trainings
Life-long Education
Establishment of a Center of Parallel I/O
Standardization and long-term Vision
Continuous Funding needed
Volunteer Recognition
Questions for the Audience
Is there anything missing?
Is there any overlap?
Data Sources / Acquisition
I/O Tools
Data Analysis and Insights
Actionable Items
Data Center
Is there overlap with community?
YES!
Interactions with the Center for Parallel I/O
What do you need from it?
What can you give back?
Targets
End Users
Care Level: --
Developers
Care Level: ++ (I/O dev) / -- (other)
I/O Researchers
Care Level: ++
System Administrators
Care Level: +/- (if stable?) / ++ (if performance is influenced)
Vendors
Care Level: +/-
Decision Makers
Care Level: - not TOP500 (no one cares for IO500)
Trainers
Care Level: ++ (if specialist) / + (know the impact)
Universities / Teaching
Care Level: --
Interests
Individual Application Performance
System Utilization
Funding for Development
Documentation of Software
Long-term Report -> Vision Documentation
Interoperability
Backward Compatibility
Reproducibility
Means
Benchmarks
Documentation
Standardization
>>>State of the Practice<<<
Workflow Design
Workload Description
Data Output Format
One to rule them all
Semantics Converters
Training Standards
I/O APIs
High-level
Low-level
Communication Channels
(e.g., Mailing Lists)
Outreach
(e.g., User Groups, Website)
Repositories (open vs. closed)
Simulators and Models

Figure 4 Overview of the community-related topics.

4.6.1 Special Characteristics in the I/O Community

I/O systems have to be conservative because they affect data that (a) has to be stored long term, (b) is relied on to be available at any time, and (c) is hard to migrate to new systems. At the same time, I/O systems are seen as a commodity that is a given. This situation leads to little incentive to make major changes even if they would result in better performance, because of the cost and effort required to make any changes, validate them, and guarantee stable and reliable operation.

Additionally, I/O work spans the entire system (storage, networks, OS, security, APIs, etc.) and is therefore highly interdisciplinary. Consequently, the entrance barrier for I/O work is high, because of both the needed knowledge and the needed testbed systems (which often have to be large scale to show impact). Further, I/O cannot be seen as isolated—or should not be seem as isolated—despite the fact that many research projects in the area focus on individual elements (e.g., one file system) and hence have a hard time showing impact or usability.

4.6.2 Overview

Figure 4 shows a summary of the discussions, split into several subcategories. These will be detailed in the remainder of this section.

4.6.3 Targets

I/O community topics affect a large number of target groups, which are listed below. Each of these groups has a different level of interest in I/O, or label "Care Level," which ranges from "++" (cares a lot) to "–" (doesn't care).

- End users: the application users running an application on an HPC system, often deploying existing codes or just minimally configuring/extending it **Care Level for I/O: –**
- Developers: developers who create both applications (used by end users) and libraries and runtime systems. The latter also contains software systems that implement I/O and/or use I/O in significant amounts. **Care Level for I/O: ++ (for I/O developer) and – (for any other developer)**
- I/O researchers: researchers in computer science (or related fields) that directly develop novel solutions to improve the I/O of HPC systems (e.g., the participants in this seminar). **Care Level for I/O: ++**
- System administrators: administrators of HPC systems responsible for safe, secure, and reliable operation of HPC systems. **Care Level for I/O: +/- (for general admins) and ++ (for storage admins)**
- Vendors: HPC system vendors selling HPC solutions to HPC centers, which often include I/O subsystems (or subcontractors specializing in I/O systems) **Care Level for I/O: ++ (for specialized I/O vendors) and +/- (for others)**
- Decision makers: the leads for public and private procurements who write RoIs and RFPs and evaluate them. **Care Level for I/O: – (main care is Top500)**
- Trainers: people running training or tutorial sessions at HPC centers and/or conferences with the target of current and future HPC users. **Care Level for I/O: ++ (for I/O trainers) and + (for general HPC trainers)**
- Teachers: those who teach I/O as part of bachelor's or master's curriculum at the university level. **Care Level for I/O: –**

4.6.4 Interests

- Performance of individual application
  from end-user perspective: improve I/O performance and reduce execution time; from support: identify application with I/O performance bottlenecks.
- System utilization
  increase the system utilization by reducing I/O waiting times; monitor overall I/O activity.
- Funding for development
  provide continuous funding of software development for I/O monitoring and analysis, essential for keeping software alive; and push R&D.
- Documentation of software
  provide good documentation of tools for support members and software developers; especially for end users, provide best practice guides that include also the usage workflow of the tools in different environments.
- Long-term report and vision document
  provide regular updated vision documents to developers, decision makers, and others, including hints about trends of development. For example, for tool developers it would be helpful to get early information about storage system development to react directly and to adapt their development strategies. Such vision documents can help end users as they adapt their I/O strategy to future trends. Long-term reports also can be seen as input for standardization efforts.

- Interoperability
  provide multiple tools for I/O monitoring exists with overlapping functionality. However, user and support people use different tools for their tasks. Interoperability of these tools would help these people easily change tools when functions are missing in one tool (see also data formats).
- Backwards compatibility
  monitor the development of the different I/O tools, and consider the backward compability of these tools. An individual application may not follow development cycles at the same speed and may require tools that can handle old data. Backward compatibility is also required for I/O monitoring tools that use old data sources (see also interoperability).
- Reproducibility
  ensure that single-application I/O measurements are reproducible (e.g., for papers). This requires that changes in tools and environment be well documented and that, for major changes, measurements be recalibrated.

4.6.5 Means

- Benchmarks
  A suite of I/O benchmarks provides a defined set of I/O metrics that can be used on different storage system and different software layers. IO500, for example, defines benchmark configurations for IOR and mdtest, collects the results in a web-accessible repository, and provides an overview of I/O capability of different storage systems.
- Documentation
  Documentation should include different perspectives of the community. It can provide an overview of the community and possible connections that the participants can expect from the community. Moreover, systematic documentation can cover technical aspects, for instance, benchmarks, repositories, simulations, and standardization.
- Standardization
  In the I/O field, some standard or common definition is still missing. The community should agree on the primary concepts in the I/O field.
  - Workflows design
  - Workload description
  - Data output format (one rule for them all, semantic converter)
    Already a group of different performance tools can be used for I/O performance analysis. The same parameter can be named in different ways in individual tools. The community can organize discussions with researchers and tool developers to agree on reasonable data output formats or identify one generic semantic converter allowing global understanding of the results from individual tools.
  - Training standards
    With proper training, end users can improve the I/O performance of their application. In some rare cases, however, the training is organized in an unattractive or even inefficient way, resulting in the training being held in low regard and fewer people seeking it. The training material and techniques should be standardized at a high level in order to guarantee the quality of the training.
  - I/O APIs (high Level, low level)
    The community should define the standards for both high-level and low-level I/O APIs. There is heated discussion about whether a higher-level or lower-level API should be considered in the optimization of the POSIX performance. With optimization of POSIX using low-level APIs, the POSIX APIs are kept, and therefore the applications built

on POSIX need no update, but standard low-level APIs are still necessary. When the optimization of POSIX is built on high-level APIs, an enormous number of applications need to be updated according to the new standard, which should be defined by the community.

- Communication channels (e.g., mailing Lists)
  Users should be able to contact researchers sharing the same interest. For instance, I/O performance analysis tools should be available if related research needs this tool but technical problems exist.
- Outreach (e.g., user groups, website)
  In order to reach specific community targets, such as user groups, a website with well-organized structure and sufficient information should be provided and maintained.
- Repositories (open vs. closed)
  The I/O community should host a repository to allow the members to find useful material. Open repositories will be the practical method to provide the information and source code. But in reality, some innovative tools or ideas are developed by Ph.D. students, who need to use them for personal publication. In this situation, protected or closed repositories should be the choice for them to receive feedback from the community while having their data secured. Another beneficial functionality can the "issue" system, whereby researchers or users can report issues to the developers.
- Simulators and models
  One of the essential aims of the I/O community should be to encourage innovation in the I/O research field, such as I/O performance analyzers. In the early stage of the development of the tools, the simulators based on reliable I/O models should be the foundation of the testbed.

4.6.6 State-of-the-Art Resources

4.6.6.1 Benchmarks and Tool Catalog

- IO500, https://www.vi4io.org/io500
- I/O tools and benchmarks: https://www.vi4io.org/tools/start
- https://www.vi-hps.org/tools/tools.html

4.6.6.2 Training events

- XSEDE user training: https://www.xsede.org/for-users/training
- PRACE training portal https://training.prace-ri.eu/
- GCS: https://www.gauss-centre.eu/trainingsworkshops/
- NHR training events
- VI-HPS: https://www.vi-hps.org/training/index.html

4.6.7 Gaps

- Lack of community interaction
  Little interaction currently occurs among I/O researchers, tool developers, and file system administrators outside of research projects or within a single center. Users might get confused by getting different recommendations, for example, for tools that they should use to understand I/O behavior. A platform is needed for exchanging knowledge and developing common strategies to guide users. A recurring exchange between I/O researchers, tool developers, and file system administrators is necessary in order to develop and maintain such strategies and get the most viable outcome. Furthermore, users should have a common place to asking questions regarding their I/O problems.

- Lack of training
  With a steady exchange between users and I/O experts, an identification of common problems should lead to better training resources. At the moment training resources are limited. Specialized offers by single projects or sites exist, but they are often coupled with certain events or are just a one-shot activity. The community lacks recurring general-purpose training needed to understand parallel I/O behavior.
- Lack of semantics documentation
  While an overlap exists in the functionality and reporting capabilities of I/O tools, the output data formats are often different and specific. The lack of documentation on the semantics of data formats makes direct comparison of the results of different tools difficult.
- Lack of an I/O tool catalog
  The community should provide a full catalog of available tools for understanding I/O behavior. The catalog should list the tools, state their main purpose (what the tool is good for, what the motivation behind its development is), provide some contact to the maintainer, and identify the current state of the tool (if it is a production tool or research prototype, or if it is no longer maintained).
- Lack of I/O awareness
  Users and nonexperts are often not aware of the implications coming with parallel I/O at scale because these issues do come not come up during the development on local machines.
- Lack of research and development platforms
  Scientific software projects commonly are hosted on site or project internal platforms (e.g., site internal GitLab instances). This situation could hinder collaboration between the I/O research community and, for example, tool developers. A valuable effort might be to provide an open site-independent platform to host source code repositories and issue tracking and documentation to improve collaboration and visibility of the different projects.

4.6.8 Recommendations

To bridge the gaps in the I/O community construction, we propose the following recommendations.

- **Community testbed** should be built to enable testing of existing I/O performance analysis tools and to provide an opportunity for testing during updating and development of new tools. Vendors should be welcome to participate in the community testbed. The integration of the testbed should be continuous. Ideal output format could be I/O SDK.
- **Benchmark suites to drive research and development** should be regularly maintained. When an old benchmark no longer matches the real use case, it should be removed from the suites. New benchmarks should be allowed to be added after comparison with the old ones in the suites. Some new benchmarks might be totally different from the old ones; but when they can give the I/O researcher new ideas or reflect a new tendency, they should be added to the suites.
- **Cross-center badges or description requirements** such as compute time application or academic paper artifacts should motivate participation in the community and contribute to the community.
- **Community software development** should be based on the “researcher, developer, user” cycle. I/O tool catalog and monitoring tools should be summarized by the community to accelerate the development of the software.
- **Education** should cover three types: (1) continuing education, through massive open online courses (MOOCs) such as Coursera or FutureLearn, which could attract more fresh blood to focus on parallel I/O during their academic life; (2) for university students,

common curricula such as courses and parallel I/O training to make them aware of the importance of I/O; and (3) life-long education, which could help developers improve the performance of their applications.

- **Establishment of a center of parallel I/O** should produce the standardization and long-term vision of parallel I/O and attract continuous funding, for example, for well-structured Dagstuhl seminars.
- **Volunteer recognition** in addition to the network should attract young researchers to join the community and choose parallel I/O as their further research field.

5 Recommendations and Conclusions

Several recurring themes emerged over the course of Dagstuhl Seminar 21332, *"Understanding I/O behavior in scientific and data-intensive computing"*. These high-level themes represent areas in which further research is expected to yield the largest impact across the field as a whole.

- **Tools and techniques are needed that span the full hierarchical scope HPC I/O behavior**: data centers, workflows, applications, processes, system software, and hardware. The current state of the art has produced silos of information that make it difficult to correlate low-level characteristics with high-level trends and link root causes to their broader impact.
- **Interoperability, and more specifically the ability to translate findings and techniques across the field at large, is hindered by lack of commonality and standardization.** This phenomenon is evident at multiple levels: terminology, data formats, monitoring granularity, and availability of infrastructure for storage and analysis of characterization data.
- **Lack of insight into the performance impact of monitoring tools, along with a lack of mechanisms to adjust that impact, has limited the deployment of performance monitoring infrastructure.** Large storage systems are high-value shared resources, and administrators will err on the side of caution if potential performance degradation or reliability impacts are not well understand.
- **Workloads used for evaluation and investigation of storage behavior are not representative of production applications.** They fail to capture salient characteristics of relevant applications, fail to capture novel workloads such as AI, or lack sufficient breadth to reproduce emergent system workload properties.
- **The ultimate impact of I/O behavior analysis is gated by our ability to transfer findings to end users and facility operators.** Current state-of-the-art tools cater primarily to researchers and systems experts. The gap between the two calls for better training, greater communication, more incentives to enact changes, clear risk/reward assessments, and simpler visual representations such as roofline models.
- **The diversity of platforms, workflow systems, high-level libraries, and applications makes it difficult to apply corrective action in a consistent way.** Many inconsistent tuning options are available to users and administrators. The field would benefit from unified models for data movement and more ways to reason about I/O properties that transcend the idiosyncrasies of individual scientific problem domains.
- **Advances in understanding HPC I/O behavior are difficult to sustain over time in production environments.** Root causes include rapid innovation in storage technology (which does not typically consider I/O instrumentation as a first-class citizen), difficulty in securing funding for software maintenance, lack of R&D platforms, challenges in workforce cultivation, and insufficiently robust system models.

State of the Field and and Future Outlook

Understanding HPC I/O behavior is crucial not just for today's systems; it is becoming even more important as scientific computing becomes more data centric and reliant on more diverse data sources. Comprehensive understanding of I/O behavior is the cornerstone technology that underpins any effort to optimize data access within applications and systems. Without it, we risk wasting valuable resources in research, procurement, and optimization of HPC systems and applications.

Extensive ongoing literature contributions continue to demonstrate the impact of advances in understanding HPC I/O behavior. In this context, the unique value of Dagstuhl Seminar 21332 was to produce a comprehensive, holistic view of the field by synthesizing the perspectives of a diverse collection of subject matter experts. This holistic viewpoint made it clear that our already impressive ability to understand and interpret I/O behavior would be even more impactful if we could address the most common shared roadblocks that have emerged across HPC facilities and application development teams. The community as a whole is well positioned to undertake research that addresses these challenges.

6 Abstracts of Lightning Talks

6.1 Analyzing POSIX I/O semantics of parallel applications

Sebastian Oeste (Technische Universität Dresden, DE)

Joint work of Sebastian Oeste, Michael Kluge, Andreas Knüpfer, Wolfgang E. Nagel

POSIX I/O and its restrictive access semantics are an already known bottleneck for the performance of distributed network file systems. Some parallel file systems relaxing specific POSIX semantics to provide better performance. While there are existing tools to test the POSIX compliance of the file system there is no tool to test the POSIX requirements of the application. In this talk I want to discuss a methodology for a tool to analyze the POSIX I/O requirements of parallel applications.

6.2 Andreas Knüpfer: I/O Aspects in the Center-Wide and Job-Aware Cluster Monitoring System PIKA

Andreas Knüpfer (Technische Universität Dresden, DE)

Joint work of Frank Winkler, Robert Dietrich, Andreas Knüpfer, Sebastian Oeste, Wolfgang E. Nagel

The lightning talk presented the basics about the center-wide, continuous cluster performance monitoring system "PIKA" which is in production at TU Dresden for over three years. It collects a set of performance metrics on all nodes in granularity of 1-2 samples per minute and maps them to batch system jobs. This allows a performance overview for live and past HPC jobs for users (for their jobs) and the computing center (for all jobs by all users).

PIKA contains also I/O metrics and the lightning talk focused on those metrics, how they can be visualized in various forms (timelines, histograms, scatter plots against other metrics, footprint summaries) and how one can search/filter jobs according to metrics.

Finally, the talk argued that I/O metrics are special and more difficult for automated judgement into sufficiently fast or too slow. Reasons for this are among others that (i) maximizing the I/O rates towards the nominal capacity of the infrastructure all the time is not desirable – unlike for CPU utilization or FLOP rates, (ii) peaks and gaps in I/O are expected, and (iii) even heavy I/O phases may be averaged out over the entire life span of long jobs. This was the starting point for a discussion how to assess PIKA's continuous I/O metrics in a more appropriate way.

Further details are given in [1] and the software is available under an Open Source license at [2].

References

1 Robert Dietrich, Frank Winkler, Andreas Knüpfer, Wolfgang E. Nagel; *PIKA: Center-Wide and Job-Aware Cluster Monitoring*, In proc. of 2020 IEEE International Conference on Cluster Computing (CLUSTER), Kobe, Japan, Sept. 2020, pp. 424-432, DOI 10.1109/CLUSTER49012.2020.00061.

2 GitLab repositpory of the PIKA project at `https://gitlab.hrz.tu-chemnitz.de/pika`

6.3 Can We Gamify I/O Performance?

Philip Carns (Argonne National Laboratory, Lemont IL, USA)

There is a fundamental manpower scalability problem in the arena of HPC I/O performance tuning: the number of scientific users is becoming larger and larger, while the availability of facility I/O experts is not. This talk explores the challenges and potential for better engaging the more "scalable" former group by *gamifying* I/O performance tuning. This will require methods for identifying competitors, a scoring system, competition rules, and incentives. None of these things are immediately straightforward to create, but the potential payoff for doing so is a way to not only improve system efficiency, but also to more deeply engage users in the process and develop novel technologies for reasoning about I/O strategies and their relative merits.

6.4 Lifting the user I/O abstraction to workflow level – a possibility or in vain?

Julian Kunkel (Universität Göttingen / GWDG – Göttingen, DE

This talk revisits the current workflow specifications. Mostly these are implicitly defined by task dependencies and hide the I/O characteristics. It then discusses the potential to exploit user workload specifications on a higher level. Lastly, the community could jointly work on such higher-level specifications.

Further details for the climate/weather community are given in [1].

References

1 Julian Kunkel, Luciana Pedro; *Potential of I/O Aware Workflows in Climate and Weather*, In Supercomputing Frontiers and Innovations, Series: Volume 7, Number 2, pp. 35-53, (Editors: Jack Dongarra, Vladimir Voevodin), Publishing Center of South Ural State University (454080, Lenin prospekt, 76, Chelyabinsk, Russia), ISSN: 2313-8734, 2020-04, DOI 10.14529/jsfi200203

6.5 An IO500-based Workflow For User-centric I/O Performance Management

Radita Liem (RWTH Aachen University, DE)

I/O performance in a multi-user environment is challenging to predict. It is hard for users to know what to expect when running and tuning their application for better I/O performance. In this project, we evaluate IO500 as a user-centric workflow to manage their expectations on their application's I/O performance and devise an optimization strategy specific to the target cluster's capability.

IO500 benchmark is a standard benchmark for HPC storages systems and is designed to create a balanced performance. In our workflow, we use the IO500 benchmark scenarios "easy" and "hard" to get the best and worst possible performance results of the cluster's bandwidth and metadata rate. Then, we create a bounding box of user expectations with these scenarios and map the application's I/O performance within this box. With the mapped I/O performance, we can understand which part of the application needs to be improved and the possible extent of this improvement.

Our experiments confirm that the bounding box of user's expectations can be created. In doing so, this project is a promising first step towards the mapping and improvement of the application's I/O performance.

Details are given in [1].

References

1 Dmytro Povaliaiev, Radita Liem, Jay Lofstead, Christian Terboven. An IO500-based Workflow For User-centric I/O Performance Management. Poster presented at: ISC High Performance 2021; June 24 – July 2, 2021.

6.6 IOMiner – A multi-level analysis to detect root causes of I/O bottlenecks

Suren Byna (Lawrence Berkeley National Laboratory, Berkeley, USA

Joint work of Suren Byna, Teng Wang, Suren Byna, Glenn Lockwood, Philip Carns, Shane Snyder, Sunggon Kim, Nicholas Wright

To understand the root causes of poor performing I/O jobs, we have conducted a zoom-in analysis and developed a set of analyses that were put together as a tool called IOMiner. In this presentation, we present an analysis of platform, application, and job level IO instrumentation logs using parallel coordinates and sweep-line analysis. We will also describe issues that caused performance bottlenecks in a few I/O use cases and solutions that resolved them.

Further details of IOMiner are described in [1] and [2], and the software is available with an Open Source license at `https://github.com/hpc-io/IOMiner` [3].

References

1 Teng Wang, Suren Byna, Glenn Lockwood, Nicholas Wright, Philip Carns, Shane Snyder, *IOMiner: Large-scale Analytics Framework for Gaining Knowledge from I/O Logs*. IEEE Cluster 2018.

2 Teng Wang, Suren Byna, Glenn Lockwood, Philip Carns, Shane Snyder, Sunggon Kim, Nicholas Wright, *A Zoom-in Analysis of I/O Logs to Detect Root Causes of I/O Performance Bottlenecks*, IEEE/ACM CCGrid 2019.

3 GitLab repositpory of the IOMiner tool at `https://github.com/hpc-io/IOMiner`

Participants

- Wolfgang Frings
Jülich Supercomputing Centre, DE
- Yi Ju
Max Planck Computing and Data Facility – Garching, DE
- Andreas Knüpfer
TU Dresden, DE
- Julian Kunkel
Gesellschaft für wissenschaftliche Datenverarbeitung – Göttingen, DE
- Erwin Laure
Max Planck Computing and Data Facility – Garching, DE
- Radita Liem
RWTH Aachen University, DE
- Frank Mueller
North Carolina State University – Raleigh, USA
- Sarah Neuwirth
Goethe-Universität Frankfurt am Main, DE
- Sebastian Oeste
TU Dresden, DE
- Martin Schulz
TU München, DE

Remote Participants

- Marcus Vincent Boden
Gesellschaft für wissenschaftliche Datenverarbeitung – Göttingen, DE
- Jim Brandt
Sandia National Laboratories – Albuquerque, USA
- André Brinkmann
Universität Mainz, DE
- Suren Byna
Lawrence Berkeley National Laboratory, Berkeley, USA
- Philip Carns
Argonne National Laboratory, USA
- Fahim Tahmid Chowdhury
Florida State University – Tallahassee, USA
- Hariharan Devarajan
Lawrence Livermore National Laboratory, USA
- Ann Gentile
Sandia National Laboratories – Albuquerque, USA
- Sivalingam Karthee
Huawei Technologies – Reading, GB
- Roland Laifer
KIT – Karlsruhe Institute of Technology, DE
- Jay Lofstead
Sandia National Laboratories – Albuquerque, USA
- Johann Lombardi
Intel Corporation – Meudon, FR
- Stefano Markidis
KTH Royal Institute of Technology – Stockholm, SE
- Sandra Adriana Mendez
Barcelona Supercomputing Center, ES
- Kathryn Mohror
Lawrence Livermore National Laboratory, USA
- Michael Ott
LRZ – München, DE
- Marc Snir
University of Illinois at Urbana-Champaign, USA
- Shane Snyder
Argonne National Laboratory, USA

- Mehmet Soysal
KIT – Karlsruhe Institute of Technology, DE

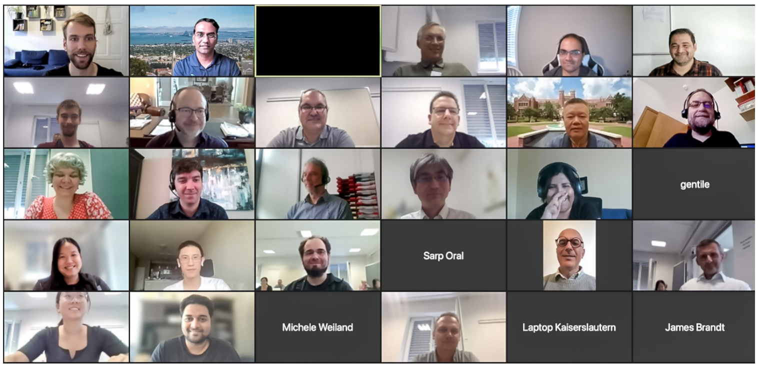
- Osamu Tatebe
University of Tsukuba, JP

- Devesh Tiwari
Northeastern University – Boston, USA

- Chen Wang
University of Illinois at Urbana-Champaign, USA

- Michèle Weiland
EPCC, The University of Edinburgh, GB

- Weikuan Yu
Florida State University – Tallahassee, USA



Report from Dagstuhl Seminar 21342

Identifying Key Enablers in Edge Intelligence

Edited by

Aaron Ding[1], Ella Peltonen[2], Sasu Tarkoma[3], and Lars Wolf[4]

1 **TU Delft, NL,** aaron.ding@tudelft.nl
2 **University of Oulu, FI,** ella.peltonen@oulu.fi
3 **University of Helsinki, FI,** sasu.tarkoma@helsinki.fi
4 **TU Braunschweig, DE,** wolf@ibr.cs.tu-bs.de

Abstract

Edge computing, a key part of the 5G networks and beyond, promises to decentralize cloud applications while providing more bandwidth and reducing latencies. The promises are delivered by moving application-specific computations between the cloud, the data-producing devices, and the network infrastructure components at the edges of wireless and fixed networks. However, the current AI/ML methods assume computations are conducted in a powerful computational infrastructure, such as a homogeneous cloud with ample computing and data storage resources available. In this seminar, we discussed and developed presumptions for a comprehensive view of AI methods and capabilities in the context of edge computing, and provided a roadmap to bring together enablers and key aspects for edge computing and applied AI/ML fields.

Seminar August 22–25, 2021 – http://www.dagstuhl.de/21342
2012 ACM Subject Classification Computer systems organization → Distributed architectures; Computing methodologies → Artificial intelligence; Networks
Keywords and phrases artificial intelligence, communication networks, edge computing, intelligent networking
Digital Object Identifier 10.4230/DagRep.11.7.76
Edited in cooperation with Atakan Aral, atakan.aral@univie.ac.at

1 Executive Summary

Aaron Ding
Ella Peltonen
Sasu Tarkoma
Lars Wolf

Research Area

Edge computing, a key part of the upcoming 5G mobile networks and future 6G technologies, promises to decentralize cloud applications while providing more bandwidth and reducing latencies. The promises are delivered by moving application-specific computations between the cloud, the data-producing devices, and the network infrastructure components at the edges of wireless and fixed networks. The previous works have shown that edge computing devices are capable of executing computing tasks with high energy efficiency, and when combined with comparable computing power to server computers.

Identifying Key Enablers in Edge Intelligence, *Dagstuhl Reports*, Vol. 11, Issue 07, pp. 76–88
Editors: Aaron Ding, Ella Peltonen, Sasu Tarkoma, and Lars Wolf
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In stark contrast to the current edge-computing development, current artificial intelligence (AI) and in particular machine-learning (ML) methods assume computations are conducted in a powerful computational infrastructure, such as a homogeneous cloud with ample computational and data storage resources available. This model requires transmitting data from end-user devices to the cloud, requiring significant bandwidth and suffering from latency. Bringing computation close to the end-user devices would be essential for reducing latency and ensuring real-time response for applications and services. Currently, however, these benefits cannot be achieved as the perspective of "edge for AI", or even "communication for AI", has been understudied. Indeed, previous studies address AI only limitedly in different perspectives of the Internet of Things, edge computing, and networks.

Clear benefits can be identified from the interplay of ML/AI and edge computing. We divide this interplay into edge computing for AI and AI for edge computing. Distributed AI functionality can further be divided into edge computing for communication, platform control, security, privacy, and application or service-specific aspects. Edge computing for AI centres on the challenge of adapting the current centralized ML and autonomous decision-making algorithms to the intermittent connectivity and the distributed nature of edge computing. AI for edge computing, on the other hand, concentrates on using AI methods to improve the edge applications or the functionalities provided by the edge computing platform by enhancing connectivity, network orchestration, edge platform management, privacy or security, or providing autonomy and personalized intelligence on application level.

Previous studies address accommodating AI methods for different perspectives of IoT, edge computing and networks. However, there is still a need to understand the holistic view of AI methods and capabilities in the context of edge computing, comprising for example predictive data analysis, machine learning, reasoning, and autonomous agents with learning and cognitive capabilities. Further, the edge environment with its opportunistic nature, intermittent connectivity, and interplay of numerous stakeholders present a unique environment for deploying such applications based on computations units with different degrees of intelligence capabilities.

The AI methods used in edge computing can be further divided into learning and decision making. Learning refers to building, maintaining and making predictions with ML models, especially neural networks. Decision making is the business logic, that is, the process of acting upon the predictions. This is the domain of decision theory, control theory and game theory, whose solutions and equilibrium are now often estimated with data by reinforcement learning methods.

Currently, AI's cloud-centric architecture requires transmitting raw data from the end-user devices to the cloud, introducing latencies, endangering privacy and consuming significant data transmission resources. The next step, currently under active research, is distributed or federated AI, which builds and maintains a central model in the cloud or on the edge but allows user devices to update the model and use it locally for predictions. We envision a fully decentralized AI which flattens the distributed hierarchy, with the joint model built and maintained by devices, edge nodes and cloud nodes with equal responsibility. The present challenges for AI in edge computing converge on 1) finding novel neural network architectures and their topological splits, with the associated training and inference algorithms with fast and reliable accuracy and 2) distributing and decentralized model building and sharing into the edge, by allowing local, fast-to-build personalized models and global, collaborative models, and information sharing. Finally, the novel methods need to be 3) integrated with key algorithmic solutions to be utilised in edge-native AI applications. The ground-breaking objectives and novel concepts edge-native artificial intelligence brings are:

- Edge-native AI can be used for obtaining higher quality data from massive Internet of Things, Web of Things, and other edge networks by filtering out large volumes of noise, context labelling, dynamic sampling, data cleaning, etc. High-quality data can thus be used to feed both edge inferencing and cloud-based data analysis systems, for example, training large-scale machine learning models.
- Edge computing provides low latency that is crucial especially for real-time applications, such as anything related to driving and smart mobility. AI applications on the edge and thus closer to the end-user will not only fasten existing applications but also provide opportunities for novel and completely new solutions.
- Edge-based computing provides data privacy when users are involved, and no need to share data to the cloud services but only the locally learned model.
- With edge-computing implemented for AI/ML model building, personalisation of such models can be done in local environments without unnecessary transmission overhead (when only local data is anyway considered for model building). Global models built in the cloud environment can be used to support these local models whenever a collaborative, large or more general model is requested.
- Edge-native AI/ML tasks provide mobility of the computation and cloudlet-like processing in the edge. In comparison to cloudlets, edge computing provides more flexibility and dynamic operations for load balancing, task management, distribution of the models, etc.
- Light-weight computation on the edge devices and local environments can enable energy savings.
- Ethical data management: edge-native AI can be used to keep data ownership control closer to the user, e.g., when computation is managed and task distribution controlled from the user’s own devices, and suitable security and privacy protection methods are in use.

2 Table of Contents

Executive Summary

Overview of Talks

Panel discussions

3 Overview of Talks

3.1 Edge intelligence for Environmental Monitoring and Protection

Atakan Aral (Universität Wien, AT)

Joint work of EU CHIST-ERA SWAIN project partners

Contemporary machine learning algorithms are not limited by the training data available to them, but the computing power needed to process that data. This is particularly critical for the systems that learn from streaming big data and take time-critical decisions because storing the data and processing in batches is not an option.

In this talk, I introduced a use case for edge intelligence in environmental monitoring and disaster prediction. Water resource contamination substantially threatens the environment. Rapid identification of chemicals and their emission sources in watersheds is crucial for sustainable water resources management. Despite studies on the measurement of micropollutants in the water resources around Europe, efficient utilization of the data in decision-making to protect water resources from detrimental chemical pollution is currently not available. Novel Internet of Things technologies, coupled with advanced Artificial Intelligence and Edge Intelligence strategies, may provide faster and more efficient responses to these challenges in real-time reactions as well as long-term planning.

3.2 NSF AI Institute for Edge Computing Leveraging Next Generation Networks (Athena)

Yiran Chen (Duke University – Durham, US)

As the world is embracing the fifth generation of mobile networks (5G), mobile network infrastructures are facing a double whammy. On one hand, they have made bold performance promises, anticipating previously impossible mobile apps and services, and foretelling a mass deployment of Internet of Things (IoT) devices. On the other hand, their massive computation needs, conventionally carried by dedicated, specialized hardware at the base station, are projected to have to be met by multi-tenant datacenters at the edge and in the Cloud, as network operators are aggressively cutting cost, especially capital investment, and bracing for a future of cloud-native mobile networks. We seek to kindle and fuel revolution for wireless communications by tapping the ongoing revolution in Artificial Intelligence (AI). In doing so, we will develop AI-powered transformative technologies for next-generation mobile networks at the edge as well as new algorithmic and practical foundations of AI, such that the new functionalities, efficiency, scalability, security, privacy, and fairness of the AI solutions can be adopted in the next-generation wireless communications.

3.3 Edge Intelligence

Schahram Dustdar (TU Wien, AT)

In this talk I discuss the challenges ahead when researching the confluence of Internet of Things, Edge Computing, Fog Computing, and Cloud Computing. In particular we discuss the topics related to research issues in the area of AI and Edge Computing.

Introduction

Today's systems infrastructure is composed out of three building blocks: People, Software Services, and Things. There are two fundamental approaches to discuss such infrastructures. The first one is the cloud centric perspective. This basically views the Cloud as the center of the world, and everything else is connected to the cloud. Some people call it the brain. In this case, everything is centralized and the brain is the most important thing. In this view, IoT is always connected to the Cloud as all machine learning and decision making is done on the Cloud, nothing goes unnoticed by the Cloud. The second perspective is the Internet-centric view. Here, decision making, learning, model building etc. is also done on the edge of a network; partially consolidated and transferred in a federated fashion to Cloud systems, if needed. In this talk we propose to look at the whole compute continuum and utilize IoT, Edge, Fog, and Cloud in all our systems development and engineering efforts. 5G or 6G base stations in the future, they are typically general-purpose computing infrastructures, sometimes they are telecom operator-controlled pieces of equipment, sometimes they are belonging to organizations or even to individuals. Currently, the fog infrastructure base stations, for example, is one example for that. The Cloud has essentially unlimited compute and storage resources, with the full spectrum of cloud services with high availability and lower costs is another important part of the compute continuum. Now, we know that there is a new family of applications, which require extremely low latency and different levels of privacy and security that actually cannot work only with the Cloud; autonomous driving is a good example for that. One needs to have extremely low latencies. Another example is telemedicine in real time surgery. Hence, this means that you have to make sense out of the whole spectrum from the IoT via the Edge, Fog, to the Cloud.

Compute Continuum

In this talk I suggest a software-intensive Edge systems focus. We completely rethink the design and the operation of such an environment. Why is that necessary? The main reason is that we have fundamentally conflicting factors concerning the system requirements, which need to be resolved. So on the one hand side, we have latency. There is this an inherent traditional division of the Cloud and IoT, which has different time factors and performance factors, which we can manage better when we look at it from a software intensive side. Secondly, we have computation as an edge resource, which basically means that we need to use edge infrastructures similarly to cloud infrastructures to perform complex infrastructure tasks, such as safety and security. And thirdly, we have the question of locality and mobility, where we can introduce novel solutions to privacy software configuration and system evolution. So the question is, which characteristics of edge computing systems then should be abstracted as first class citizens to the underpinning model?

Elastic Diffusion

In this talk we will first understand this from a hypothetical perspective. Is it possible to move the computation and decision making the model creation, etc, closer to where the data is actually being created? In other words, to take proximity, context or capability and energy more into account. That is definitely some an important area of research that people are working on. In this talk, we will focus on what I call the main principle, which is elastic diffusion. We will break it down into essentially two points. The first one is elasticity. Elasticity is a property that we know from physics, and it is basically a property, which says something about the resilience. We know elasticity from physics, it's a property of returning to initial form or a state following some deformation. In other words, you put force on a material, it changes its shape. When you take away that force, it goes back to its initial form. That is the principle of elasticity. In, in science in neuroscience for the brain, they call it plasticity. So you learn something and something changes its shape, so to speak. The second principle we discuss is Osmotic Computing. Osmosis is a principle from Chemistry. Molecules flow from higher to lower concentration. Similarly, we aim at mimicking the flow of microservices (functionality) from Cloud, Fog, and Edge devices from and to each other. In this talk we will discuss what Elasticity and Osmosis mean for the domain of Edge Intelligence and what the fundamental research question entail.

3.4 AWS Wavelength and Verizon 5G Edge

Janick Edinger (Universität Hamburg, DE)

Modern applications generate complex tasks that must be executed promptly and reliably. Edge computing in combination with 5G networks offer computing capabilities that can be accessed with ultralow latencies. However, unlike cloud computing, edge computing introduces a new type of infrastructure that needs to be installed, configured, and maintained. AWS Wavelength and Verizon 5G Edge provide a standard cloud infrastructure at the speed of the edge. Applications with strict latency and high bandwidth requirements benefit from this deployment, among them AR/VR rendering, 360-degree video streaming, real-time monitoring of connected cars, as well as the detection of quality issues on fast-moving assembly lines in smart factories.

3.5 Future Cloud Computing View – A Perspective from LRZ

Dieter Kranzlmüller (LMU München, DE)

The Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities is the IT service provider for science in Munich, Bavaria, Germany and Europe and thus a partner in many scientific projects. Recently, more and more demand is raised from Edge devices, which work in combination with LRZ´s own cloud infrastructure. Several examples demonstrate the workflow from the senders, through edge and network into the cloud. The crucial question is where to place the respective functionality , starting from processing and storing. The answer on this question depends on the capabilities of components along the edge-cloud continuum. Clouds serve as a means of accessoring largescale (federated) and "always-on" resources. In summary, clouds will continue to grow in terms of capacities and capabilities, limited only by availability of funding and power provisioning, which offering more and more heterogenous resources, from special AI devices to future QC accelerators. In any case, applications will benefit from using the entire spectrum corresponding to their specific needs and characteristics.

3.6 Performance and Security in Edge Video Analytics

Ling Liu (Georgia Institute of Technology – Atlanta, US)

The rapid growth of wireless mobile broadband communication networks has fueled new capabilities in scalable device-to-edge-to-cloud continuum, ranging from increased data rates, ultra-low latencies, larger coverage with massive number of devices connected 24x7. These advances have enabled new edge assisted applications, such as Augmented Reality/Virtual Reality (AR/VR) and video analytics. In this invited talk for Edge AI at Dagstuhl Seminar 21342, I will describe research challenges for performance and security in edge video analytics with dual goals. First, I will advocate high degree of resilience against systemic and adversarial disruptions for scalable video analytics on heterogeneous edge devices. Second, I will advocate combining multiple innovative techniques synergistically to provide the end-to-end resilience for next generation intelligent systems.

3.7 Towards Ubiquitous Intelligence in 6G

Sasu Tarkoma (University of Helsinki, FI)

Main reference Xiang Su, Xiaoli Liu, Naser Hossein Motlagh, Jacky Cao, Peifeng Su, Petri Pellikka, Yongchun Liu, Tuukka Petäjä, Markku Kulmala, Pan Hui, Sasu Tarkoma: "Intelligent and Scalable Air Quality Monitoring With 5G Edge", IEEE Internet Comput., Vol. 25(2), pp. 35–44, 2021.
URL http://dx.doi.org/10.1109/MIC.2021.3059189

The presentation focused on the motivation and development of ubiquitous Intelligence for beyond 5G systems towards 6G. Ubiquitous Intelligence pertains to the fusion and integration of sensing, AI, and connectivity in a hyper-local context. This emerging paradigm is expected to support many current and new vertical application areas, such as holographic interaction, tactile Internet, and massive-scale intelligent city services. Ubiquitous Intelligence requires the real-time discovery and interconnection of sensing components, context gathering, and storage with the algorithmic elements such as context estimation and prediction, positioning, and general AI algorithms. Due to the diversity of use cases, ubiquitous Intelligence requires asynchronous data-driven communications, serverless and function-based operation, and separation of concerns between the service types, such as communications, sensing, and AI.

The key aim of the paradigm is to support networks and applications that operate in the ubiquitous computing environment. Ubiquitous Intelligence aims to support a cognitive network architecture capable of accommodating the requirements of the verticals. In addition, it seeks to facilitate the design and deployment of vertical applications that can utilize the resources of the programmable world.

We envisage that the ubiquitous Intelligence environment is hierarchical in terms of geography and capabilities. The end-to-end environment consists of endpoints with opportunistic interactions through fog computing and various edge and core cloud processing tiers. In typical interactions, mobile devices and industrial devices utilize nearby capabilities for processing first and then send content for processing at more distant cloud levels. Learning and inference are distributed with partial models being generated and aggregated in the end-to-end environment.

3.8 Energy-efficient Energy-efficiency Calculations using Edge Intelligence

Michael Welzl (University of Oslo, NO)

Machine Learning (ML) methods can learn traffic patterns to make better decisions for energy efficiency in communications, e.g. for wireless devices. However, ML itself is quite energy-hungry. We can solve this dilemma by assigning the ML task to an edge device that is powered by renewable energy, e.g. via a solar panel.

4 Panel discussions

4.1 Breakout Session: Future Cloud View

Tobias Meuser (TU Darmstadt, DE)

Joint work of Based on the discussions between the remote attendees and the two panelists: Dieter Kranzlmüller and Schahram Dustdar

In the session on the future cloud perspective, we discussed technical and non-technical aspects of the interaction between edge and cloud. From the non-technical view, cloud and edge should aim to collaborate to improve the overall service quality. From the technical view, the resource constraints, energy efficiency, trust, and privacy have been discussed and considered as future research challenges.

4.2 Breakout Session: Beyond 5G View

Nitinder Mohan (TU München, DE)

Joint work of Based on the discussions between the remote attendees and the two panelists: Jari Arkko and Sasu Tarkoma

"Edge computing" is still a diffused term as the definition and possibilities of the "edge" are varied – which makes it more widespread and opens room for innovation. While academics have considered utilizing and deploying edge in devices (and servers) in home and office environments, the industry has proposed engraving the edge compute capabilities within their managed network infrastructure, e.g. cellular backbones, on-premise devices (e.g., home gateways), industry building (servers), enterprise, or smart city infrastructure (e.g., roadside units). Despite the "exact" availability, the benefits of the edge, and its synergy with supporting complex AI-based applications, are only possible if the cellular/networking fabric seamlessly connects users/sensors to such servers. Take, for example, the case of autonomous vehicles within a smart city environment. The most optimal operation of the vehicle is only possible if the AI models in the vehicle and the smart city are in sync with each other, i.e. the city learns of the driving destination and requirements from the vehicle to dynamically adjust its smart traffic control. Simultaneously, the vehicle feeds in data about congestion and road conditions to tweak its speed, lane, etc. Such an interaction is possible if the network interconnecting the two is itself self-learning and is able to adapt to dynamic environmental changes.

In the seminar, the participants discussed not just how to achieve such a vision within the up-and-coming 5G connectivity but also how to go (above and) beyond it. With future cellular access technologies, three pertinent opportunities can be explored. Firstly, the improved communication speeds along with reliable wireless communication will only enable novel IoT sensors and actuators that can help improve the granularity of data and control in AI applications. Secondly, in future cellular standards, there can now be a possibility to integrate AI within inherent control decisions (using smart switch and router hardware) that transparently integrates data generated in different ingress ports to improve not only the Quality-of-Experience of AI applications but also the Quality-of-Service of the network itself. Finally, there is a possibility to tailor computations in the network to be "human-driven" which integrates specific characteristics of humans (e.g. mobility patterns) into training its ML models. This way, for future cellular technologies, AI and edge computing will not just enable "human-in-the-loop" applications but also "human-centric" applications. Here, the participants agreed that a unified edge-in-the-network model should be worked out in the near future to account for the high cost of operation and management for supporting such an extensive and pervasive network of compute servers integrated deeply within the network fabric.

4.3 Breakout Session: AI/ML View on Edge Intelligence

Gürkan Solmaz (NEC Laboratories Europe – Heidelberg, DE)

Joint work of Based on the discussions between the remote attendees and the two panelists: Prof. Yiran Chen and Prof. Ling Liu

The remote breakout session on artificial intelligence (AI) and machine learning (ML) view focused on two main topics: 1) Edge infrastructure for distributed intelligence, 2) Distributed and federated learning models.

Currently, AI accelerators have the problem of applying AI designed for specialized hardware on another hardware. For instance, training an AI model on hardware can be fast, whereas training the same AI model would take longer on other machines. To satisfy requirements of distributed intelligence, it would be beneficial to have programmable devices instead of "black-box" devices and re-use existing techniques developed in edge and cloud computing. In recent years, GPUs have become larger and more expensive, whereas many small and embedded devices have become available. The new requirements for edge infrastructures may create new opportunities for industry stakeholders such as internet service providers, cloud providers, and local providers of hardware/software.

Distributed and federated learning has been of interest to academic and industrial research, especially for motivating edge computing scenarios such as autonomous driving. For distributed/federated learning, it is not straightforward to run advanced AI on embedded devices. A solution to this might be training lower granularity AI models on embedded devices and leveraging edge AI outcomes as features for more advanced models. Furthermore, researchers need to explore the trade-offs between accuracy vs. fairness and generalization vs. performance of AI/ML for edge intelligence. Moreover, it is not clear how different AI models can be adapted, re-used, and integrated. There has been ongoing research on multi-task learning and multi-modal data sources; on the other hand, there is a need for a unified AI library that implements joint loss functions for different AI models.

Participants

- Christian Becker
Universität Mannheim, DE
- Schahram Dustdar
TU Wien, AT
- Janick Edinger
Universität Hamburg, DE
- Lauri Lovén
University of Oulu, FI
- Tri Nguyen
University of Oulu, FI
- Ella Peltonen
University of Oulu, FI
- Jan Rellermeyer
TU Delft, NL
- Martijn Warnier
TU Delft, NL
- Lars Wolf
TU Braunschweig, DE

Remote Participants

- Atakan Aral
Universität Wien, AT
- Jari Arkko
Ericsson – Jorvas, FI
- Yiran Chen
Duke University – Durham, US
- Eyal De Lara
University of Toronto, CA
- Aaron Ding
TU Delft, NL
- Fred Douglis
Peraton Labs –
Basking Bridge, US
- Diego Ferran
Telefónica Research –
Barcelona, ES
- Thomas Hiessl
Siemens AG – Wien, AT
- Dewant Katare
TU Delft, NL
- Dieter Kranzlmüller
LMU München, DE
- Ling Liu
Georgia Institute of Technology –
Atlanta, US
- Madhusanka Liyanage
University College Dublin, IE
- Ivan Lujic
TU Wien, AT
- Setareh Maghsudi
Universität Tübingen, DE
- Nitinder Mohan
TU München, DE
- Iqbal Mohomed
Samsung AI Research –
Toronto, CA
- Roberto Morabito
Ericsson – Jorvas, FI
- Petteri Nurmi
University of Helsinki, FI
- Jörg Ott
TU München, DE
- Francesco Regazzoni
University of Amsterdam, NL &
Università della Svizzera italiana
– Lugano, CH
- Olga Saukh
TU Graz, AT
- Stefan Schulte
TU Hamburg, DE
- Henning Schulzrinne
Columbia University –
New York, US
- Maarten Sierhuis
Nissan Research Center –
Sunnyvale, US
- Stephan Sigg
Aalto University, FI
- Pieter Simoens
Ghent University, BE
- Gürkan Solmaz
NEC Laboratories Europe –
Heidelberg, DE
- Sasu Tarkoma
University of Helsinki, FI

- Wiebke Toussaint
TU Delft, NL

- Antero Vainio
University of Helsinki, FI

- Marten Van Dijk
CWI – Amsterdam, NL

- Maarten van Steen
University of Twente, NL

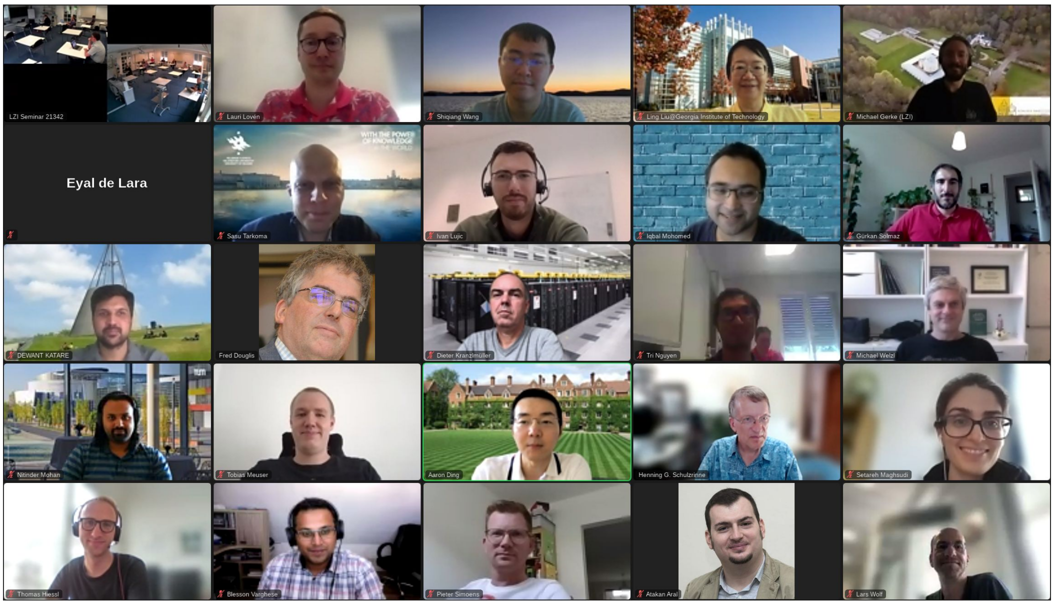
- Blesson Varghese
Queen's University of Belfast, GB

- Shiqiang Wang
IBM TJ Watson Research Center – Yorktown Heights, US

- Klaus Wehrle
RWTH Aachen, DE

- Michael Welzl
University of Oslo, NO

- Chenren Xu
Peking University, CN



Report from Dagstuhl Seminar 21351

Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics

Edited by

Timothy Baldwin[1], William Croft[2], Joakim Nivre[3], and Agata Savary[4]

1 **The University of Melbourne, AU**
2 **University of New Mexico – Albuquerque, US**
3 **Uppsala University, SE**
4 **Université de Tours – Blois, FR**

Abstract

Computational linguistics builds models that can usefully process and produce language and that can increase our understanding of linguistic phenomena. From the computational perspective, language data are particularly challenging notably due to their variable degree of **idiosyncrasy** (unexpected properties shared by few peer objects), and the pervasiveness of non-compositional phenomena such as **multiword expressions** (whose meaning cannot be straightforwardly deduced from the meanings of their components, e.g. red tape, by and large, to pay a visit and to pull one's leg) and constructions (conventional associations of forms and meanings). Additionally, if models and methods are to be consistent and valid across languages, they have to face specificities inherent either to particular languages, or to various linguistic traditions.

These challenges were addressed by the Dagstuhl Seminar 21351 entitled "Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics", which took place on 30-31 August 2021. Its main goal was to create synergies between three distinct though partly overlapping communities: experts in typology, in cross-lingual morphosyntactic annotation and in multiword expressions. This report documents the program and the outcomes of the seminar. We present the executive summary of the event, reports from the 3 Working Groups and abstracts of individual talks and open problems presented by the participants.

Seminar August 30–31, 2021 – http://www.dagstuhl.de/21351
2012 ACM Subject Classification Computing methodologies → Artificial intelligence
Keywords and phrases computational linguistics, morphosyntax, multiword expressions, language universals, idiosyncrasy
Digital Object Identifier 10.4230/DagRep.11.7.89
Edited in cooperation with Artur Kulmizev, Meagan Vigus

Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics, *Dagstuhl Reports*, Vol. 11, Issue 07, pp. 89–138
Editors: Timothy Baldwin, William Croft, Joakim Nivre, and Agata Savary
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Timothy Baldwin (The University of Melbourne, Australia, tbaldwin@unimelb.edu.au)
William Croft (niversity of New Mexico, Albuquerque, USA, wcroft@unm.edu)
Joakim Nivre (Uppsala University, Sweden, joakim.nivre@lingfil.uu.se)
Agata Savary (University of Tours, France, agata.savary@univ-tours.fr)

This Dagstuhl Seminar was initially planned as a 1-week event in June 2020 (with number 20261) with the following objectives:

- **Theoretical**: To deepen the understanding of language universals, and of how they apply to linguistic idiosyncrasy, so as to further promote unified modelling while preserving diversity.
- **Practical**: To improve the treatment of idiosyncrasy in treebanking frameworks, in computationally tractable ways and, thus, to foster high quality NLP tools for more languages with greater typological diversity.
- **Networking**: To promote a higher degree of convergence across typology-driven initiatives, while focusing on three main aspects of language modelling: morphology, syntax, and semantics.

Due to the COVID-19 pandemic, the event was first rescheduled and finally reduced to a 2-day online event on 30-31 August 2021, with two 3-hour sessions, repeated for better inclusiveness of various time zones (which corresponds to about 20% of the initially planned duration).

Prior to the event, participants submitted discussion issues, based on which working groups and the program were formed, as described in our Wiki space[1].

More precisely, the program of the event followed the Dagstuhl model:

- A list of recommended **readings** was published prior to the event
- **Introductory talks**, given by the 4 organizers, ensured common understanding of the scope and challenges to address.
- **Personal introductions** of all participants helped achieve a community building effect, despite the online setting.
- **Working groups** (WGs) were built on the basis of the discussion issues submitted by the participants. Each WG had 4 co-leaders, at least one of which could attend repeated sessions, so as to ensure consistency between the 2 time-zone sub-groups. The following WGs were created:
  - WG1: What counts as a word?
  - WG2: What counts as a MWE and as a construction?
  - WG3: Syntax vs. semantics
- **Discussion issues** were addressed in WGs by the proposers' short introductions followed by brainstorming.
- Plenary **reporting** sessions from WGs took place twice for every time zone.

The event attracted 51 participants, who judged it successful and expressed the need for a full-size onsite follow-up event. All the organizational details and outcomes of the seminar are gathered in our Wiki space[2].

[1] https://gitlab.com/unlid/dagstuhl-seminar/-/wikis/home
[2] `https://gitlab.com/unlid/dagstuhl-seminar/-/wikis`

Despite its very reduced and fully online format, the seminar achieved part of its objectives, stressed the importance of some initially-defined research questions, gave rise to new questions, and showed the efficiency of some instruments.

- On the **networking** side, the intended convergence effect was clearly apparent. While the initial proposal and invitee list was dominated by NLP-oriented members of the UD and PARSEME communities, strong contributions came notably from the less numerous typology and UniMorph experts. The four communities interacted actively, and reinforcing these interactions is intended for the near future. Notably, steps were taken towards:
  - integrating typology experts in the PARSEME core group
  - accompanying a seminal work in typology (Croft, to appear) with a "companion volume" about practical implementation of morphosyntactic concepts in UD.
- On the **theoretical** side, the event showed:
  - The importance of the research question *How to identify words across languages?* (item I.A in the seminar proposal), to which the whole of Working Group 1 was dedicated. In particular, new insights from lesser-studied languages, brought by typology experts, allowed us to broaden the perspective on this issue.
  - The need for capturing the relationship between the two fundamental notions in this proposal: a multiword expression and a construction, studied by Working Group 2. From the linguistic and typology perspective, a MWE is a special case of a construction, which is rarely made explicit in current NLP models. But the notion of a construction needs a more formal definition to be implementable in NLP, notably as far as the type-token opposition is concerned (question II.B in the seminar proposal). Thus, the typology-NLP interactions are essential in the quest for an optimal model.
  - The scope of the syntax-semantics interface issues (question II in the proposal) addressed by Working Group 3. On the one hand, the interests of the community in this respect exceeded the scope intended by the event organizers. Namely corpus-lexicon interlinking for all language units, not only for MWEs, was targeted. On the other hand, MWEs are exemplars of condensed syntax-semantic interface issues, and as such provide good case studies in this domain.
- On the **practical** side, some initial proposals emerged as to harmonizing UD treebank annotation guidelines with: (i) modelling morphological properties at the subword level (heavily studied by UniMorph), (ii) labelling MWEs (core activity of PARSEME).

Each multidisciplinary approach like ours bears heavy risks of intractability. This is because different communities often have different objectives and points of view on the same phenomena, and they may fail to agree on a unified approach, or even on the usefulness of working towards such a unification. In our case, there is a tension between:

- diversity and descriptive detail required in linguistics,
- necessary simplifications for the sake of robustness in NLP.

In other words, it is legitimate to question the usefulness of universality-driven initiatives (in NLP) if idiosyncrasy and diversity are basic properties of language data. Yet even typologists seek language universals which abstract away from the idiosyncrasy.

We feel that the event allowed us to mitigate this tension. Namely, even if a universality-based treebank fails to render the diversity of possible analyses of a language phenomenon, it is still useful not only for NLP applications but also for linguistic and typological analyses. This is because relevant examples are easy to extract (and to further re-interpret), as long as the annotation is consistent and well-documented.

Another barrier-lifting effect of the event concerned the relation between UD and PARSEME. It seems that the MWE categories defined by UD and PARSEME are less incompatible than initially expected, simply because the definition of an MWE in itself is different in UD and PARSEME. This could have been a source of major incompatibility but since a MWE does not really have a status in the UD annotation process, the discrepancies could (at least in some cases) be overcome relatively easily.

In conclusion, the event provided, in our opinion, a proof of concept for the framing objectives set up in the original Dagstuhl seminar proposal. However, since the effective framework and duration was severely reduced as compared to the initially intended setting, only part of these objectives could be achieved. Thus, we are currently putting efforts to ensure follow-up events. In particular, a new Dagstuhl seminar with roughly the same objectives has been submitted.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

Open problems

3 Overview of Talks

3.1 Multiword Expressions

Timothy Baldwin (The University of Melbourne, AU)

A multiword expression ("MWE") satisfies the following two conditions: (1) it is decomposable into multiple simplex words; and (2) it is lexically, phonetically, phonologically, morphosyntactically, semantically, and/or pragmatically idiosyncratic. Given the focus of this workshop on MWEs, Universal Dependencies, and Linguistic Typology, we primarily focus on lexical, morphosyntactic, and semantic idiosyncrasy in this talk, in addition to noting that our definition relies crucially on the concept of "simplex word". Lexical idiosyncrasy occurs when an MWE has one or more elements which do not have a usage outside of MWEs, such as *ad* in *ad hoc*, or *fro* in *to and fro*. Morphosyntactic idiosyncrasy occurs when the morphosyntax of the MWE differs from that of its components, as happens in the case of the transitive *wine and dine [SOMEONE]* "entertain [SOMEONE] with wine and food", as distinct from the intransitive *dine* (and also *wine*, although the simplex verbal usage of *wine* is, in itself, uncommon). Semantic idiosynrasy occurs when the meaning of the MWE is not simply the sum of its parts, either due to there being a mismatch in semantics (e.g. *blow hot and cold* "alternate between two polar-different moods or attitudes", which is completely divorced semantically from its component words) or there being extra semantics encoded in the MWE not found in the component words (e.g. *designated driver* being associated with the specific situation of a group going out to drink alcohol, with the designated driver making sure they drink in such a way as to be able to legally drive the group home).

Particular complications with MWEs as pertain to this workshop include: (a) What is a "word" in our definition, noting complications with non-segmenting languages, and also languages without a pre-existing writing system (Walpiri, Mohawk, ...)? (b) How to proceduralise/text for the different forms of idiosyncrasy in a way which generalises across different languages? (c) What is an MWE and what is (purely) constructional? (d) How should MWEs be represented to capture their (cross-linguistic) idiosyncrasies (but also their compositionality)?

Determinerless prepositional phrases (i.e. PPs where the head noun is a singular count noun, and lacks a determiner, such as *in gaol* or *per student*) are an excellent case study of the complexities of determining whether a given expression is an MWE or not. Two properties of particular interest with determinerless PPs are: productivity (i.e. how productive is a given preposition in combing with different nouns), and modifiability (i.e. can the noun in the determinerless PP be pre- or post-modified). In English, determinerless PPs populate the full spectrum from non-productive, non-modifiable constructions such as *ex cathedra* to fully-productive, fully-modifiable constructions such as *per*, e.g. *per recruited student that finishes the project.* Most determinerless PPs, however, lie between these extremes, and have limited productivity, and also idiosyncratic modifiability properties.

3.2 Multiword expressions: constructional and typological perspectives

William Croft (University of New Mexico – Albuquerque, US)

Multiword expressions have played a large role in construction grammar (Goldberg 1995, 2006; Croft 2001), and construction grammar provides a different and possibly useful perspective on the treatment of MWEs in computational linguistics. In computational linguistics, MWEs are often described as “words with spaces”, that is, MWEs are assimilated to the lexicon. In construction grammar, one can describe constructions as “MWEs without words”, that is, syntax is assimilated to MWEs. Syntactic constructions are organized in a lattice, in which the highest nodes are schematic and general syntactic structures, and the lowest nodes are the most specific and restricted constructions – that is, prototypical MWEs (Croft and Cruse 2004). The real issue in analyzing MWEs from a constructional perspective is: how general is the construction/MWE and its parts?

The seminal paper in construction grammar (Fillmore et al. 1988) classifies idioms in a way that demonstrates the continuum of generality. Idioms are made up of unfamiliar pieces: words that occur nowhere else, or familiar pieces: words that occur in other constructions. The pieces are unfamiliarly arranged: a syntactic pattern occurring nowhere else, or familiarly arranged: a syntactic pattern found in other constructions. Both occur to varying degrees.

The most MWE-like are fully substantive: every part of the construction is a specific word. An MWE like the rhetorical question *Who’s gonna make me?* consists of familiar pieces familiarly arranged. An idiom like *all of a sudden* consists of familiar pieces but they are familiarly arranged. The idiom *kith and kin* includes an unfamiliar piece, *kith*, but in a familiar NP coordination construction. For a truly unfamiliar arrangement of unfamiliar pieces, one must turn to borrowed phrases such as *joie de vivre*.

The problematic cases for MWE analysis are constructions which are partly general (“familiar”) in some way or another. In the *The Xer, the Yer* construction, as in *The bigger, the better*, the form *the* is unfamiliar – it comes from an Old English oblique demonstrative form, not the definite article – and so is the parallel paratactic construction for comparatives. The unfamiliar piece *heed* occurs only in *pay heed to* and *take heed of*, which are familiar constructions (Nunberg, Sag and Wasow 1994). Likewise, comparative *than* occurs in only the comparative construction, which fluctuates between the older elliptical subordinate clause construction *She is taller than I am* and the innovative oblique phrase construction *She is taller than me.* With respect to familiar pieces, the construction *Nth cousin M times removed* represents an otherwise unfamiliar syntactic pattern. So do the English Auxiliaries, which have unique syntax in negative and interrogative constructions, as in *Isn’t she nice?* (Bybee and Thompson 1997).

Finally, familiar pieces familiarly arranged may also have unfamiliar semantics and differing degrees of flexibility (generality) in syntax. The idiomatically combining expression *pull strings* is a substantive argument structure construction, consisting of only the verb *pull* and the noun *strings* as well as a schematic Subject role. It occurs in a variety of other constructions, including the Passive and the Object Relativization constructions: *Strings were pulled for me; the strings that she pulled for me….* Other substantive constructions such as the idiomatic phrase *kick(ed) the bucket* are more restricted, occurring only in different tense-aspect constructions.

These constructions raise the question: where do we draw the line for MWEs in this continuum of syntactic and semantic specificity? Should we draw the line at all? Even the most general, most “compositional” constructions, the modifier-noun construction and argument structure constructions have semantic idiosyncrasies. *Red pen* could refer to the color of the surface of the pen, stripes on the pen, the ink, and so on. The hue described by *red* varies with wine, hair, beans and so on. Argument structure constructions have verb-specific meanings for Subject, Object and so on; compare *I dried the dishes, I saw the hawk, She entered the room, This switch calibrates the temperature.*

Typologically, MWEs display certain patterns. Some MWEs evolve into single words, for example Proto-Basque **gu-re kide-a-n* 'in the company of us' > Basque *gu-rekin* 'with us' (Trask 1996:115-16) or Old English *ear wicga* 'ear one_that_moves' > *earwig* (Brinton and Traugott 2005:50). As a result, all of the idiomatic patterns described above occur with morphemes in words as well as in multiword expressions. This raises the question of how important is it to draw the "word level" line, not to mention how difficult is it? (Zingler 2020)

The syntax of MWEs displays some common patterns across languages, based on their diachronic origin. MWEs that grammaticalize to inflections are typically syntactically fixed phrases, or become so. The commonest of these are flags (case markers), TAMP (tense-aspect-modality-polarity) markers, and conjunctions. MWEs that lexicalize to referring phrases use the same strategies as modification constructions (Pepper 2020), and tend to be syntactically fixed, though often morphologically flexible. MWEs that evolve to complex predicates are the most varied in origin. They include light verbs (*They had a drink; You paid attention to me!*), serial/compound verbs (*Go fetch the paper*), copulas (*She is a teacher*), verb-argument phrases (*Strings were pulled for me, Butter wouldn't melt in Pat's mouth*) and verb-particle constructions (*She cut it up, The pond froze over*). Secondary predicates are also MWE-like (*The pond froze solid, They shot him dead/to death*). Complex predicates of all types tend to be syntactically flexible, and sometimes partly general.

The typology of multiword expressions remains to be explored in greater detail. They raise their own questions: Are there "universals of idiosyncrasy"? How do we find and formulate them?

References

1 Laurel J. Brinton and Elizabeth Closs Traugott. *Lexicalization and Language Change*. Cambridge University Press, Cambridge, UK, 2005

2 Joan L. Bybee and Sandra A. Thompson. *Three frequency effects in syntax*. Proceedings of the 23rd Annual Meeting of the Berkeley Linguistics Society, ed. Matthew L. Juge and Jeri O. Moxley, 378-88. Berkeley Linguistics Society, Berkeley, CA, 1997

3 William Croft. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press, Oxford, UK, 2001

4 Croft, William and D. Alan Cruse. *Cognitive Linguistics*. Cambridge University Press, Cambridge, UK, 2004

5 Charles J. Fillmore, Paul Kay and Mary Catherine O'Connor. *Regularity and idiomaticity in grammatical constructions: the case of* let alone. Language 64:501-538, 1988

6 Adele E. Goldberg. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press, Chicago, IL, 1995

7 Adele E. Goldberg. *Constructions At Work: The Nature of Generalization in Language*. Oxford University Press, Oxford, 2006

8 Geoffrey Nunberg, Ivan A. Sag and Thomas Wasow.*Idioms*. Language 70:491-538, 1994

9 Steve Pepper. *The Typology and Semantics of Binominal Lexemes: Noun-Noun Compounds and their Functional Equivalents*. PhD thesis, University of Oslo, Oslo, NO, 2020

10 R. L. Trask. *Historical linguistics*. Arnold, London, 1996

11 Tim Zingler. *Wordhood Issues: Typology and Grammaticalization*. PhD dissertation, University of New Mexico, Albuquerque, NM, 2020

3.3 Principles of the UD Annotation Framework

Joakim Nivre (Uppsala University, SE)

Universal Dependencies is a framework for cross-linguistically consistent morphosyntactic annotation and a project to create annotated corpora in this framework for as many languages as possible. The project started in 2014 when version 1 of the annotation guidelines was released together with 10 treebanks. Since then, data sets have been released roughly every six months, with the most recent release (v2.8) featuring 202 treebanks representing 114 languages. A major milestone was the release of version 2 of the guidelines in 2016. For more information about the guidelines and resources, we refer to [5] for version 1 and to [6] for version 2. The linguistic theory underlying the annotation framework is laid out in [4].

The goal of UD is to enable cross-linguistically consistent morphosyntactic annotation to support multilingual research in natural language processing and linguistics. Ideally, UD should therefore facilitate meaningful linguistic analysis within and across languages and support morphosyntactic processing in monolingual and cross-lingual settings. To facilitate adoption of the framework, it is based on pre-existing de facto standards and common usage, in particular an evolution of (universal) Stanford dependencies [1, 2, 3], Google universal part-of-speech tags [7], and the Interset interlingua for morphosyntactic tagsets [8]. It is important to emphasize that UD is meant to complement – not replace – language-specific annotation schemes. For researchers interested in the finer details of a single language, UD may not be ideal since it is designed for cross-linguistic comparison and therefore by necessity has to abstract over some of these details.

A fundamental design principle of UD is a commitment to lexicalism, which in this context essentially means that the fundamental annotation units are words. Words have internal morphological properties and enter into syntactic relations with other words. Both of these aspects should be reflected in the annotation, but the principles are different and the annotation is therefore organized into two layers: a morphological layer and a syntactic layer. It is important to note, however, that the relevant notion of word here is that of a syntactic word – not a phonological or orthographical word – and UD therefore permits a two-level segmentation in order to recognize syntactic words over and above basic tokens.

In the morphological annotation layer, each word is assigned a lemma, a part-of-speech tag and (optionally) a set of features. Part-of-speech tags are taken from a revised and extended version of the Google universal part-of-speech tag set containing a fixed inventory of 17 categories. Morphological features are taken from a larger inventory that can be extended as the need arises, but where the names of features and feature values are standardized across languages.

In the syntactic annotation layer, words are connected by grammatical relations into a dependency tree, normally rooted in the main predicate of a sentence. The backbone of this structure consists of direct relations between predicates, arguments and modifiers, which are normally realized as content words. Function words are attached to the content word that they specify. This means, for example, that determiners are attached to nouns and that auxiliaries are attached to main verbs. Even adpositions are essentially treated as case markers of nominals rather than heads of prepositional phrases. The rationale for this conception of syntactic structure is to maximize parallelism across structurally different languages. By and large, relations between content words are more likely to be parallel across languages, while function words in one language often correspond to morphological inflection or nothing at all in other languages. UD provides a taxonomy of 37 universal relations for the classification of syntactic relations, with optional language-specific subtypes. The taxonomy is organized by two main principles, a distinction between core arguments and oblique modifiers at the clause level, and a distinction between three main types of linguistic structures: clauses, nominals and modifiers.

The annotation of multiword expressions (MWEs) is a challenge for UD, since they transcend the traditional morphology-syntax distinction assumed in UD. The current policy is to give a special treatment of MWEs only when they are morphosyntactically irregular. The clearest example is the special relation *fixed*, which is used to connect the components of a completely fixed grammaticalized MWEs such as *in spite of* or *by and large*. In addition, the relations *compound*, for any kind of word-level compounding, and *flat* for any kind of headless construction, can be used to annotate certain types of MWEs. However, the guidelines for handling multiword expressions in UD, and for relating the morphosyntactic UD annotation to specialized MWE annotation, is in need of further elaboration.

References

1 Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

2 Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford Typed Dependencies Representation. In *Proceedings of the COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

3 Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A Cross-Linguistic Typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*.

4 Marie-Catherine de Marneffe, Christopher Manning, Joakim Nivre, Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2), 255–308.

5 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

6 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*.

7 Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*.

8 Daniel Zeman. 2008. Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

3.4 Multilingual modelling of verbal multiword expressions in the PARSEME framework

Agata Savary (Université de Tours – Blois, FR)

Joint work of the PARSEME community
Main reference Agata Savary, Marie Candito, Verginica Barbu Mititelu, Eduard Bejček, Fabienne Cap, Slavomír Čéplö, Silvio Ricardo Cordeiro, Gülşen Eryiğit, Voula Giouli, Maarten van Gompel, Yaakov HaCohen-Kerner, Jolanta Kovalevskaitė, Simon Krek, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Lonneke van der Plas, Behrang QasemiZadeh, Carlos Ramisch, Federico Sangati, Ivelina Stoyanova, Veronika Vincze: "PARSEME multilingual corpus of verbal multiword expressions", Language Science Press, 2018.
URL http://dx.doi.org/10.5281/zenodo.1471591

PARSEME is a community which emerged from the European COST Action on *Parsing and Multiword Expressions* funded by the European Commission in 2013–2017 [4, 3]. It gathered 31 countries, 30 languages and 6 dialects from 10 language genera. It had a number of outcomes (publications, language resources, tutorials, methodologies and a book series[3]). Notably, a collaborative effort of 25 language teams resulted in annotation guidelines for verbal multiword expressions, unified across 25 languages, as well as in a manually annotated corpus for these languages. This resource has seen 3 releases so far, and is distributed under open licenses.

By the PARSEME definition, a multiword expression (MWE) is a continuous or discontinuous sequence of words which: (i) contains at least two *lexicalized components*, including a head word and at least one other syntactically related word, (ii) displays some degree of idiosyncrasy. A component of a MWE is said to be lexicalized[4] if replacing it by semantically related words results in a meaning shift which goes beyond what is expected from the replacement (as in ***turn the tables*** "change from a weaker to a stronger position" vs. *turn the chairs*, *rotate the tables*). Thus, by contrast with Croft's introductory talk in this workshop, the notion of lexicalization applies not only to phrases but to their components as well. PARSEME admits lexical, morphological, syntactic and/or semantic idiosyncrasy as a defining criterion for MWEs. Thus, by contrast with [1], collocations, i.e. expressions exhibiting statistical idiosyncrasy only, are not included in the scope of MWEs.

In the PARSEME guidelines and corpus, focus is on verbal MWEs (VMWEs). A VMWE is a MWE such that: (i) its *canonical form* builds a (weakly) connected graph whose head is a verb, (ii) it passes the idiosyncrasy tests from the PARSEME guidelines. A canonical form is the least syntactically marked syntactic variant which preserves the idiomatic reading, e.g. a finite verb, active voice, non-negated form and a form with no extraction are considered less syntactically marked than infinitive/participle, passive voice, a negated form, and a form with an extraction, respectively. For instance, if we come across the occurrences on the left-hand side of Fig. 1, we first transform it into a canonical form (like the one on the right-hand side), and this is the forms which we test against the guidelines.

The PARSEME guidelines were conceived with 3 main objectives in mind: (i) to formalise idiomaticity in a cross-linguistically unified and computationally tractable way, (ii) to unify what is truly similar, thus emphasizing what is language-specific, (iii) to make the annotation reproducible. These objectives imply a series of principles and constraints. Namely,

[3] *Phraseology and Multiword Expressions* at Language Science Press: `https://langsci-press.org/catalog/series/pmwe`

[4] In examples, hexicalized components are highlightes in bold.

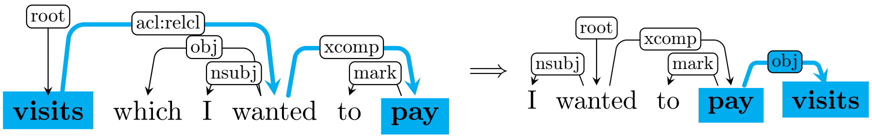


Figure 1 Transforming candidate VMWEs into their canonical forms.

annotation follows a decision diagram (with a unique starting point) and atomic decisions are binary (although non-compositionality is a matter of scale). Semantic non-compositionality is considered the major property to capture but is hard to test directly. Therefore it is approximated by lexical and morpho-syntactic inflexibility. For instance in French, *la porte s'ouvre* (lit. "the door opens itself") "the door opens" contains a combination of a verb (*ouvre* "opens") and of a reflexive clitic (*s'* "itself"), which might be idiomatic. However, this expression means roughly the same as *quelqu'un ouvre la porte* "someone opens the door", which proves that this is a regular middle passive (or inchoative) use of the reflexive clitic, rather than a VMWE. Such inflexibility tests are driven by the syntactic structure, which creates a strong dependence on the underlying syntactic theory. For the sake of cross-lingual validity, PARSEME annotation largely relies on the morpho-syntactic annotation provided by Universal Dependencies [2].

The PARSEME guidelines put forward a typology of VMWEs with categories of 3 kinds. Firstly, *universal categories* (i.e. occurring in all 25 languages under study) consist of: (i) *verbal idioms* (VIDs), like *to* ***call it a day***, and (ii) *light verb constructions* (LVCs) with two subtypes: LVC.full (*to* ***give*** *a* ***lecture***) and LVC.semi (***grant rights***). Secondly, *quasi-universal categories* occur in many languages but not all: (iii) *inherently reflexive verbs* (IRVs), like as in *to* ***help oneself*** "to take something freely", (iv) *verb-particle constructions* (VPCs) have two subtypes: VPC.full (*to* ***do in*** "to kill") and VPC.semi (*to* ***eat up*** "to eat completely", (v) *multi-verb constructions* (MVCs) like ***copy-paste***. Finally, *language-specific categories* are allowed and one has emerged so far: *inherently clitic verbs* (LS.ICV) in Itialian, like ***prenderle*** (lit. "to take it") "to be beaten".

The PARSEME quest for universality-oriented modeling of VMWEs opens many questions to which typology experts could greatly contribute. For instance, we would like to know if the two VMWE categories identified as univeral (VIDs and LVCs) truly occur beyond the 25 languages which we have studied. The annotation guidelines for MVCs also need more insight. There, Indo-European verb-verb constructions like ***make do*** or ***copy-paste*** in English are classified along with serial verbs like ***kar le-na*** (lit. "do take") "do something for one's own benefit" in Hindi. It remains unclear if this reflects true similarities rather than "false friends". Moreover, the statistics of the PARSEME corpus also reveal intriguing distributional phenomena. Notably, combinations with a verb and a reflexive clitic (as *I wash myself, she bought herself a present*) are frequent in many languages. Still, IRVs are frequent in Slavic and Romance, as well as in German, but rare or non-existent in other languages (e.g. English).

Future work in PARSEME also calls for stronger synergies with Universal Dependencies (UD). We already benefit from the UD definition of a word (a pre-requisite for defining a MWE) and from the UD morpho-syntactic annotations (pre-requisites for syntax-driven inflexibility tests). Thus, the universality of the UD categories and tags enables the universality of the PARSEME guidelines. However, joint challenges still need to be addressed. Firstly, the UD and the PARSEME definitions of a MWE are partly redundant and competing. For instance ***let alone*** in the following example is annotated as a MWE both at the level of UD dependencies (with the `fixed` label) and in the PARSEME layer (as a VID): *they never gave him a present,* ***let alone*** *a cake.* Moreover, some UD relations only partly overlap with the PARSEME categories. For instance, UD defines *inherently reflexive verbs* (marked with the `expl:pv` label) as those which never occur without the reflexive clitic. This corresponds to only part of the PARSEME criteria for IRVs.

Future work in the PARSEME corpora initiative includes: (i) extending the annotation guideines to new MWE categories (nominal, adjectival, adverbial, functional, . . .), (ii) unifying PARSEME and UD annotation guidelines, (iii) validating them by experts in typology, (iv) including new languages and language families, (v) continuous corpus enhancements with regular releases.

References

1 Timothy Baldwin and Su Nam Kim. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, 2 edition, 2010.

2 Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07 2021.

3 Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online, December 2020. Association for Computational Linguistics.

4 Agata Savary, Manfred Sailer, Yannick Parmentier, Michael Rosner, Victoria Rosén, Adam Przepiórkowski, Cvetana Krstev, Veronika Vincze, Beata Wójtowicz, Gyri Smørdal Losnegaard, Carla Parra Escartín, Jakub Waszczuk, Matthieu Constant, Petya Osenova, and Federico Sangati. PARSEME – PARSing and Multiword Expressions within a European multilingual network. In *7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015)*, Poznań, Poland, November 2015.

4 Working groups

4.1 Working Group 1 (What counts as a word?)

Francis Tyers, Ekaterina Vylomova, Daniel Zeman, Tim Zingler

Identifying words cross-linguistically

Tim presented a typological view (Zingler 2020). There is no single definition of word, applicable to all phenomena in all languages. However, there are clues that can help with a vast majority of cases. Phonological word vs. morphological word. Phonological clues: word = domain of stress/tone assignment, vowel harmony, phonologically conditioned allomorphy. Morphological clues: fixed order (relative position of each morpheme within a word is fixed), non-selectivity (words can co-occur with different word classes, affixes cannot). Example where phonological and morphological clues yield different results: English genitive *'s*. It is not a phonological word because it is prosodically dependent on the previous segment. However, it is syntagmatically independent (non-selective), i.e., a morphological word:

(1) a. [*The boy*] *'s dog*

b. [*The boy who ran away*] *'s dog*

Wordhood issues in Universal Dependencies

Dan presented several practical issues that emerged in Universal Dependencies. In UD, the surface segments identifiable by spaces and other special characters are orthographic words, while the nodes in dependency trees are syntactic words (essentially corresponding to morphological words mentioned above). The UD issues included the following (we only give more details about the first issue in this report):

- Diverging views of word boundaries in Japanese and Korean
- Incompatible treatment of definite markers in Arabic and Hebrew
- Incompatible treatment of subject agreement morphemes in Arabic and Amharic

Japanese vs. Korean

Japanese has no orthographic words, hence there is more freedom in deciding what should count as a word-level unit for annotation and language processing. There are several different traditions how to define words in Japanese, and the one used in UD is very fine grained, making Japanese look essentially as an analytical language.

Korean has space-delimited orthographic units, and these are treated as words in UD. However, these words are rather coarse-grained. In result, Korean looks very different from Japanese, although the two languages are usually considered typologically similar. (On the other hand, Korean UD bears some similarities with Turkic languages, which are also supposed to be typologically similar to Korean and Japanese.)

The following three examples are three possible segmentations of the same Japanese phrase, meaning "I went to the beauty salon of/in Kyodo". The segmentation (2a), where verbal morphemes are treated as auxiliary words, is currently used by the Japanese UD team. (2b) would be advocated by some to be a better fit, but it is currently not used. In (2c) the postpositions are treated as case suffixes; this approach is not advocated for Japanese at all, yet it is probably the closest one to the current approach taken in Korean UD.

(2) a. *Kyōdō no miyōshitsu ni it te ki mashi ta*

b. *Kyōdō no miyōshitsu ni itte kimashita*
Kyodo of beauty.salon to going come

c. *Kyōdōno miyōshitsuni itte kimashita*

UniMorph issues related to clitics

Ekaterina presented some open problems in UniMorph, which has mostly data automatically dug out of Wiktionary (currently just the English edition of Wiktionary). The paradigm tables in Wiktionary often include analytical forms, such as the Polish conditional in (3a), or even extra paradigms for reflexive verbs, such as *podróżować się* in (3b).

(3) a. *byłybyście podróżowały*

"you would have dyed pink"

b. *byłybyście się podróżowały*

"you would have turned pink" ("you would have dyed yourself pink")

Open question: Should we keep such multi-word units in the inflectional database of UniMorph? (Related issue: The data are not consistent with respect to this. Different languages are treated differently in different language editions of Wiktionary.)

Reut: Often the morphological features of individual words participating in a periphrastic verb form (main verb + auxiliaries) do not straightforwardly show the features of the resulting form. For example, the German future *wir werden sehen* "we will see" contains a present auxiliary form and an infinitive of the main verb, but none of them alone can be described as `Tense=Fut`. So it would be actually useful to have some kind of "phrase-level features" where the future could be annotated.

Impact on studies carried on the data

Natalia presented two typological studies with UD-annotated (automatically parsed) data. In both studies, she ran the same experiment several times, each time with a slightly different definition of word (derivable automatically from the UD annotation). While the results did not vary too much for most of the languages, for some of them the difference was significant. Recommendation: manually annotated UD data should employ transparent and well documented tokenization decisions. In the ideal case, the user could switch between different levels of "word granularity". (Levshina 2020, 2021)

Artur presented another study his group did to see whether contextual neural representations have internal preference for a particular dependency scheme. The results were often in line with typological assumptions about the languages, however, some anomalies occurred, which may have been caused by inconsistent approaches to tokenization in various UD treebanks. This leads to a similar recommendation as in the study presented by Natalia. (Kulmizev et al. 2020)

Defining the word in little-known languages, with morphological singularities

Emmanuel: When describing languages that are lesser-known (and rarely written, e.g., creoles), the situation is similar to languages whose writing system does not delimit words overtly. It is often the case that various authors use various ad hoc conventions, which are not mutually compatible. Peculiar morphosyntactic properties of the language may further complicate the situation, for example, when a morphological agreement morpheme is separated from the verb whose agreement with an actant the morpheme encodes.

Dependency analysis of noun incorporation in polysynthetic languages

Fran: In polysynthetic languages, a large part of the interesting structure is hidden inside words, at the sub-word level. Example (4) is from the Chukchi UD treebank.

(4) *Qonpə nəwiswetsəqiwqinetʔəm nəmanewanɬasqewqenat*

"They (children) constantly went to play, constantly asked for money."

The colored word is an example of a nominal object incorporated in a verb. Linguists agree that this should be one word. There are phonological clues such as vowel harmony, and also morphological clues: the yellow morphemes are verbal inflectional affixes. The blue morpheme is the verbal stem "to ask", while the red mane is the incorporated object "money". The verb uses intransitive inflection; if the object were not incorporated and appeared instead as an independent word (which is also possible in Chukchi), the verb would use its transitive inflection pattern. UD sticks to the word as the basic unit, corresponding to a node in a dependency tree. Consequently, the tree of (4) does not reveal that "money" is the object of "to ask".

Dependency structure vs. word-internal morphology

David: UniMorph is currently expanding to derivational/compound morphology. Here it might be useful to annotate word-internal structure similarly to how relations between words are modeled in Universal Dependencies. Then the German compounds (5), which are typically one word in UD, could have a tree similar to what UD does with English compounds, which are typically written as multiple words.

(5) *Donau/dampf/schiff/fahrts/gesellschafts/kapitän*
lit. Donau steam ship journey company captain

Even in English, some compounds may be written with or without space (steam ship vs. steamship, or white space vs. white-space vs. whitespace), and these somewhat arbitrary orthographic choices would create asymmetric analyses in UD. Even if the space slightly changes the meaning of the compound, we want to have similar analyses:

(6) a. *We hired a dish washer*
b. *We hired a dishwasher*

Agata: Splitting German compounds is also important for annotation of multi-word expressions. For example, *Rolle spielen* "to play a role" is considered a MWE, and "role" can be modified by a word that is not part of the MWE. In English, the modifier is likely to be a separate word, thus it is easy to exclude it when annotating the MWE. However, in German the modification is likely to be realized as a compound: *Hauptrolle spielen* "to play the main role".

Possible solutions to some of the issues

Defining word cross-linguistically

There might be “good enough” criteria that work 95% (or more) of the time. One such criterion is “fixed order”: If no morpheme within a string of morphemes S can move without changing the meaning of S, then S is a word. This criterion should work most of the time, although there is at least one known exception from Huallaga Huánuco Quechua (Weber 1989: 221), where *huknayllamannaw* and *huknayllanawman* have the same meaning in (7a) and (7b):

(7) a. *Ishka-n tikra-sha huknaylla-man-naw*
lit. two-3P turn-3PERF

“The two of them have become as though one.”

b. *Ishka-n tikra-sha huknaylla-naw-man*
lit. two-3P turn-3PERF

“The two of them have become as though one.”

Annotating word-internal structure

A new layer of stand-off annotation over UD trees could be defined. This new layer would annotate word-internal structure in a fashion as similar to UD trees as possible. (Yet it would still sit clearly outside the principles of basic UD annotation, therefore it has to be a separate layer.) Such a layer could help both with annotating compounds and with the incorporated nouns in polysynthetic languages. The exact nature and taxonomy of the word-internal relations has yet to be discussed. Potential future collaboration between UD and UniMorph is foreseen here. Many of the word-internal dependency annotations could be precomputed and stored in the lexicon, with minimal context-sensitive ambiguity.

In a similar spirit, one could also think of a layer above the orthographic words, which would enable defining morphological features of periphrastic forms, or of multi-word expressions.

References

1 Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, Joakim Nivre (2020). Do Neural Language Models Show Preferences for Syntactic Formalisms? In Proceedings of ACL. https://aclanthology.org/2020.acl-main.375.pdf

2 Natalia Levshina (2020). How tight is your language? A semantic typology based on Mutual Information. In Proceedings of TLT. https://aclanthology.org/2020.tlt-1.7.pdf

3 Natalia Levshina (2021). Corpus-based typology: applications, challenges and some solutions. In: Linguistic Typology. https://doi.org/10.1515/lingty-2020-0118

4 Francis M. Tyers, Karina Mishchenkova (2020). Dependency annotation of noun incorporation in polysynthetic languages. In Proceedings of UDW. https://universaldependencies.org/udw20/papers/2020.udw2020-1.22.pdf

5 Tim Zingler (2020). Wordhood issues: Typology and grammaticalization (PhD dissertation). University of New Mexico. https://digitalrepository.unm.edu/ling_etds/71

4.2 Working Group 2 (MWEs and Constructions)

Steve Pepper, Lori Levin, Aline Villavicencio

What counts as an MWE and how are they classified?

How to define MWE

The 'standard' definition, "lexical items that can be decomposed into multiple lexemes, and display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity" (Baldwin & Kim 2010), was broadly accepted by the group, but the following questions were raised:

- Is statistical idiomaticity alone sufficient to qualify as an MWE? Tim now says no, others disagree.
- Should "lexical items" be replaced with the term 'complex constructions' (or "complex units"), in order to avoid a strict separation between lexicon and grammar?
- Why "syntactic" and not "morphosyntactic"?
- There are interdependencies between different kinds of idiosyncrasy, e.g. syntactic/semantic, etc. What are the others?
- How to determine the size and core components of an MWE?
  - Do we consider *Customer service 101* to be an MWE?
- Per Croft: Should we draw the line [between MWEs and 'vanilla' constructions]?
- Proposed revised definition: **"Multiword expressions are complex constructions that display significant lexical, morphosyntactic, semantic, pragmatic and/or statistical idiosyncrasy."**

How to classify MWEs

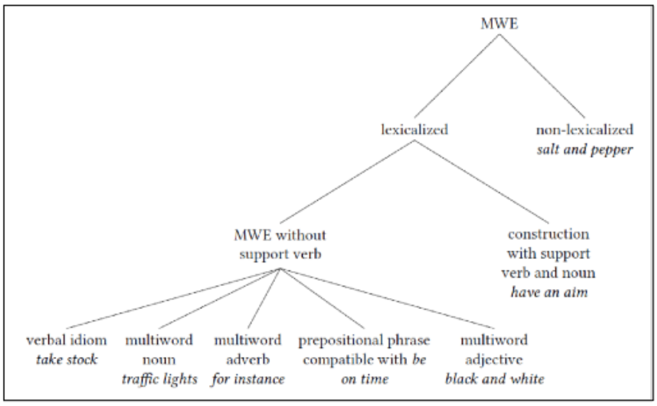


Figure 2 Proposed classification of MWEs.

Laporte (2018) observes that MWEs are "a heterogeneous set with a glaring need for classifications". However, his classification leaves much to be desired: it identifies only seven types of MWE; the top-level division is based on lexicalization – a gradient feature; the second privileges support verb constructions; and the third is a ragbag. There are similar classifications in Sag & al (2002) and Baldwin & Kim (2010). Questions raised:

1. How many different kinds of MWE are there? Can we enumerate them?
2. What types of MWE are widely (and less widely) observed cross-linguistically?

3. What classificatory principles should be applied in order to structure the domain of MWEs (grammatical functions, parts of speech, propositional acts)?
   a. Croft's take: "A primary division between **lexicalization of content words** and grammaticalization of function words. Within content words, a division between **complex predicates** (which are etymologically diverse, and often syntactically flexible) and **complex arguments and modifiers** (which usually are syntactically fixed, though often morphologically flexible). Within function words, a division between **relational function words** (case markers/flags and conjunctions), and **"other"**. Some cases that are problematic/don't fit in are idiosyncratic verb + argument structure combinations (e.g. verb + preposition) – but these are problematic in general – and pragmatically idiosyncratic expressions such as rhetorical questions."
4. Is it possible to infer the types from bottom-up annotation of corpora, for example using embedding-based approaches?

Annotating the morphemic level

1. Is the two-level word segmentation facility in UD, that is, into text (e.g. Fr. *aux*) and words (Fr. *à les*) sufficient to handle issues like the productivity of Bul. *-ica* [fem/dim] and the different orthographic systems (disjunctive and conjunctive) of closely related Bantu languages Northern Soto (Sotho) and Zulu (Nguni)?

	Orthography	Morphological analysis				
Northern Soto	ke a ba rata	*ke*	*a*	*ba*	*rat-*	-a
Zulu	ngiyabathanda	*ngi-*	*-ya-*	*-ba-*	*-thand-*	-a
	"I like them"	SC.1SG	pres	OC.CL2	verb root	inflectional ending

2. Is there a difference regarding the role of productivity (under the word level) and idiomaticity (above the word level)?

MWEs and constructions

How to define "construction"

Participants appear to have rather different notions of "construction", as evidenced by the question *is X an MWE or a construction?* In the absence of a common understanding of the term, communication is much more difficult. From the perspective of Construction Grammar, every MWE is by definition a construction:

- A construction is "the basic unit of morphosyntactic analysis ... a conventional pairing of form and function; its form is morphosyntactic structure, and its function is a combination of meaning (semantic content) and information packaging" (Croft 2022).
- What definitions of 'construction' do other people operate with?

MWEs as a subtype of construction

- If we accept that every MWE is a construction, do they constitute a separate subtype of construction?
- Or is it more reasonable to regard each MWE type a subtype of one or more other more general constructions, some of whose properties they share?
- Does an MWE "correspond to" one construction or (potentially) multiple constructions?

Comparing MWEs across languages

Which principles should be applied when comparing MWEs across languages? For instance, what allows us to perform the following two comparisons:

- The "Excess Construction", as in "so drunk he fell over": English the *so X [that] Y*, Chinese *X dao ('until') Y*, Japanese *Y hodo ('as.much.as') X*
- Eng. *in the black* vs. Port. *in the blue*

Annotation schemes for MWEs

Additional annotation schemes for MWEs

For some applications, some kinds of MWEs require more fine-grained annotation schemes than those offered by UD and PARSEME:

- Example 1: The semantic relation in noun-noun compounds and their functional equivalents is catered for only by **nmod** (and perhaps compound, which is something of a ragbag anyway). Hatcher-Bourque as a proposed annotation scheme.
- Example 2: The distinction between adverbial intensifiers and mitigators in Greek.
- Example 3: Degrees of productivity in English Verb-Particle Constructions.

Capturing types (or degrees) of idiosyncrasy

It is theoretically possible to specify how MWEs deviate from the prototypical constructions whose morphosyntactic strategies they recruit. Whether this is useful or not depends on what we are annotating (say, a corpus vs. a dictionary), and the needs of the application (e.g., parsing vs. text generation or translation). Some questions:

- Should we develop a "taxonomy of idiosyncrasies" that allows to annotate the ways in which MWEs deviate from the constructions they are instances of? For example, *how black and tan*, *salt and pepper*, *by and large* and *to and fro* deviate from the prototypical (conjunctive) coordinate nominal [N and N]N, coordinate adjectival [ADJ and ADJ]ADJ, and coordinate adverbial [ADV and ADV]ADV constructions:
- Does UD annotate enough to capture the full morphosyntactic idiosyncrasies of MWEs in context?
- What about other idiosyncrasies (lexical, semantic, pragmatic, statistical)?
- Should register be annotated?
- Can the STREUSLE approach of distinguishing between "strong" and "weak" semantic opacity be integrated into UD?
- More fundamentally, should semantic annotation (of MWEs or constructions in general) be part of UD, or should this be left to PARSEME?

Issues with UD

Some issues have been identified with UD:

- Please can we have better guidelines for "mischievous nominal constructions": names, dates, numbers; compounds; adverbial NPs?
- And also multiword connectives (*"out of"*, *"along with"*, *"based on"*, etc.)?

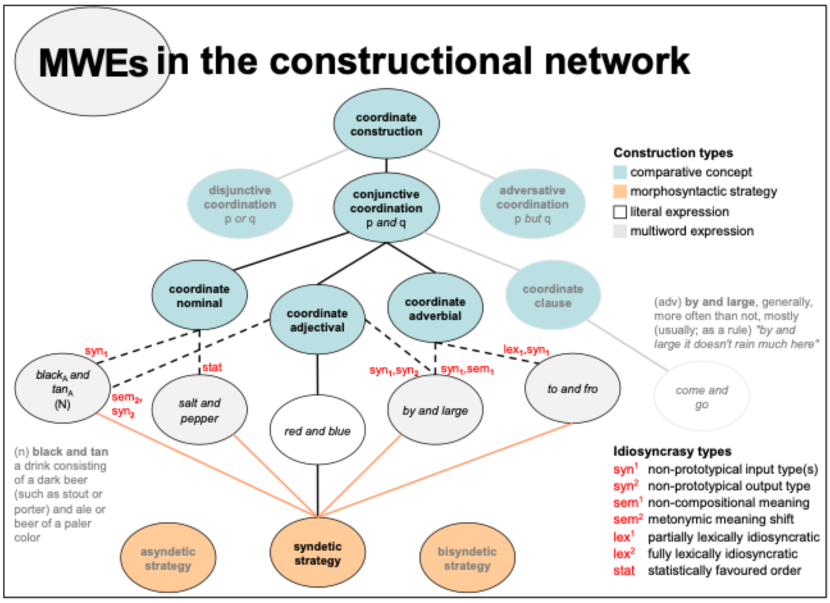

MWEs in the constructional network
coordinate construction
disjunctive coordination p or q
conjunctive coordination p and q
adversative coordination p but q
Construction types
comparative concept
morphosyntactic strategy
literal expression
multiword expression
coordinate nominal
coordinate adjectival
coordinate adverbial
coordinate clause
(adv) by and large, generally, more often than not, mostly (usually; as a rule) "by and large it doesn't rain much here"
syn$_1$
stat
sem$_2$, syn$_2$
syn$_1$,syn$_2$
syn$_1$,sem$_1$
lex$_1$,syn$_1$
black$_A$ and tan$_A$ (N)
salt and pepper
red and blue
by and large
to and fro
come and go
(n) black and tan a drink consisting of a dark beer (such as stout or porter) and ale or beer of a paler color
Idiosyncrasy types
syn^1 non-prototypical input type(s)
syn^2 non-prototypical output type
sem^1 non-compositional meaning
sem^2 metonymic meaning shift
lex^1 partially lexically idiosyncratic
lex^2 fully lexically idiosyncratic
stat statistically favoured order
asyndetic strategy
syndetic strategy
bisyndetic strategy

Figure 3 MWEs in the constructional network, with annotation of idiosyncrasies.

Issues with PARSEME

Some issues have been identified with the PARSEME framework when annotating Basque corpora:

- PARSEME expects LVCs to consist of LV+Noun. In Basque they can consist of LV+Adj (when the adjective has a morphologically identical eventive noun, like in Hindi), but also LV+Adv. We can expect the same phenomenon to occur in other languages. Is there any reason why this cannot be easily fixed?
- Basque annotators had trouble treating make a call as an LVC while receive a call should be treated as a collocation (and thus should not be annotated). Since causal verbs like give a headache are accepted in LVCs, could/should more kinds of verbs be included as well?

Some questions for discussion:

- If we want the guidelines to be universal, how acceptable/necessary are language-specific notes? Should we preferably use more general definitions?
- How detailed should definitions be in order to make sure that annotators refer to the same phenomenon/concept in multiple languages, while accepting that these phenomena might vary across languages?
- Should collocations be treated as a purely statistical phenomenon? Why should causal verbs be accepted inside LVCs but other similar cases be discarded?

References

1 Baldwin, Timothy, and Su Nam Kim. "Multiword expressions." Handbook of natural language processing 2 (2010): 267-292.
2 Croft, William. "Morphosyntax: constructions of the world's languages." (2022).
3 Sag, Ivan A., et al. "Multiword expressions: A pain in the neck for NLP." International conference on intelligent text processing and computational linguistics. Springer, Berlin, Heidelberg, 2002.

4.3 Working Group 3 (Syntax vs. Semantics)

Emily M. Bender, Jan Hajič, Marie-Catherine de Marneffe, Maria Koptjevskaja Tamm

Syntax vs. Semantics: Issues and Objectives of the Discussions

In line with the topic and goal of the whole seminar, the presentations and followup discussions have concentrated on the design and properties of the syntax/semantics interface and its components in a cross/multilingual setting. The issues are not new, but with the existence and experience of massively multilingual resources at the morphological and syntactic levels (SIGMORPHON Shared Tasks (Coterell et al. 2020) and resources, in particular UniMorph (Syllak-Glassman, 2016) and the Universal Dependencies (Nivre et al. 2016, de Marneffe et al. 2021) syntactic annotation), it is natural that the focus now turns to semantics.

There are many semantic representations, projects and environments (and some were presented here as well), and some are venturing into multilingual annotation schemes and resources. However, there is no common framework such as the one for Universal Dependencies. It was thus natural to ask fundamental questions such as whether it is ever feasible (to have a common scheme), how to design it (if ever yet) to be useful and understandable for both linguists and technologists, but also for the barely initiated (such as Ph.D. students).

Semantic Representations, Grounding and Lexicons

Semantic representations have been tackled, looking at them “top down”, from two points of view: annotation-based ones (e.g., the Prague Dependency Treebank (Hajic et al. 2020), UCCA (Abend et al., 2013) and others have been presented or mentioned) and the grammar-based ones (the Delph-in family, HPSG and MRS, ERG, etc.; see e.g (Bender et al., 2015, Copestake et al., 2005, Sag et al., 2003)). While the semantic representations based on an annotation scheme are available in more languages (albeit not many – see e.g. the MRP 2020 Shared Task (Oepen et al., 2020), where each formalism has been represented by just one additional language, besides English), the grammar-based approaches concentrate mostly on English – with some attempts to look at multilingual issues, such as at the Ling567 course at Univ. of Washington. This seems natural, as (hypothetically) unification of annotation guidelines seems to be simpler than building a grammar (necessarily?) different for each language, even if the resulting representation were uniform in their fundamental features.

However, the discussions brought up several interesting points where common schemes across languages might be possible. One example is grounding using language-neutral (or multilingual) ontologies – for example, DBpedia or Wikidata for organizations, people, locations of all sorts, domain ontologies etc. It has been mentioned that proper grounding might help to view things in context and possibly design a more uniform representation – for example, in the area of representing multiword entities (see also the WG2 report), representing synonyms, and dealing with ambiguity, both for single words (whatever a “word” is, see the WG1 report). Attempts have been mentioned to construct or convert existing verb-oriented lexical resources to an event type ontology, which is not well covered by current resources, such as WordNet or FrameNet. Examples have been shown, however, that every categorical scheme – whether grounding-based or more “lexically”-based – will necessarily have to solve several problems, such as granularity, style distinctions (how can they be grounded?), strength/positivity/negativity (and “scales” of all sorts).

Additionally, two other general and omnipresent issues have been mentioned: underspecification (and how to deal with it in the representation) and inferencing (how far to go in actually "interpreting" the utterance (and its "language") while constructing the representation. Relatedly, albeit not discussed very broadly, there was the issue of whether the semantic representation should in fact equal a "knowledge representation" (in the broader sense), or whether such a representation (either in the narrow interpretation of KR as a logic of some sort) will be still added on top of such semantic representation – but this is far away from the original "syntax vs semantics" topic of the WG, and probably deserving a seminar of its own.

Syntax-semantics Interface

While for grammar-engineering-based representations the relation to syntax is an inherent part of the grammar, it is much less clear how this interface is tackled in the annotation-based approaches. In some, there are layers which are linked together at a word (and sometimes subword) level; in some cases, at a MWE level (e.g., the Prague treebanks). In some others, there are no such links – AMR is a prime example, with others somewhere in between. There is a conceptual issue (how is it possible to connect syntax and semantics, represent this connection, at which level, etc.) and a representation (and format) issue (one schema for all – e.g., as in Enhanced UD, or two (or more) independent representations, with or without links). Several constraints are present here and have been discussed (see also the next paragraph about Cost of Access): fit to a certain background theory (of both syntax and semantics in terms of representation), boundary between the syntactic and semantic layers, harmonization across languages, simplicity of the representation(s), maintenance, error-proneness, and many more. It is also important to define the purpose – ideally the same annotation scheme should serve theoretical and computational linguists (syntactitians, semanticists, typologists, grammarians, but also researchers in the fields of pragmatics, language acquisition, cognitive science, speech and language deficiencies, etc. etc.) as well as NLP/LT developers. Some specific issues have been presented in this area, e.g., harmonization between the UD and PARSEME annotation schemes, and a consistency of the UD scheme in relation to semantics.

Cost of Access and Maintenance

The design of any representation is inevitably an exercise in compromising between a cleanliness and the effort to be able to represent the real-world language in its full breadth (cf. the Manning Laws in Universal Dependencies). One of the requirements, namely the understandability of such a representation (without an extensive training "course") for, for example, students (both linguists and computer scientists, or technologists at software development companies, etc.) – i.e., the "cost of access". There is a related issue: given that building a large multilingual collection of semantically annotated resource is costly, and therefore only possible with a community effort, how difficult is it to understand the representation and guidelines to actually build an annotated corpus (or convert it from an existing one)? And what is the cost of its maintenance? It has been also mentioned – and agreed – that for such large community efforts to be successful, people must feel a sense of community, be able to get help and guidance, and to contribute not only data, but ideas and provide input and feedback for decisions at the top level.

Conclusions and Open Questions

It is hard to come to a definite conclusions at any seminar, the less so during a two-day online seminar which did not allow, for one, for a real plenary sessions due to time zone differences; it is hard even for a full-length typical face-to-face Dagstuhl meeting. However, in WG3, we believe we have at least identified some of the most pressing questions in the area of the syntax-semantics interface and semantic representation itself from the point of view of the long-term goal, namely a multilingual universal representation of the idiosyncrasies arising in human languages. While necessarily simplifying the contents of the discussions, and perhaps even missing some important topics which might have been mentioned only briefly (and despite having 18 pages of detailed notes taken in turns by the co-leads of this WG...), here is an attempt to summarize some of the points which are worth further investigation in this area:

- What are or should be the units of semantic representation (the “concepts”), where they come from, what is their granularity, how to ensure consistency especially in the universal (multilingual) setting
- Which approach is perhaps more suitable – the grammar-engineering approach vs. the “annotation” approach – what are the advantages of one against the other, or could they be merged or can the strength be combined somehow
- Where is the sweet spot between language-specificity (and adequate representation of the semantics of that particular language) vs. cross-lingual comparability using a simplified, but “universal” or “uniform” representation
- Should the syntax-semantic interface be captured, and how: in separate layers or one annotation scheme, how tight the “linking” between layers should be (whether implicit or explicit)
- What about redundancy in the representation(s) – some is probably inevitable, but how strong should the effort be to avoid it (across layers, within the semantic layer, in the linking or grounding)
- Issues of scaling or discretization/categorization: how fine/grained it should be, how to capture differences among language
- How to represent underspecification and (immediate) inferencing – should it be captured in the representation, and if yes, how far or deep
- Cost of access – for users, contributors, technologists: how to minimize this cost already in the design of such a representation, which compromises it entails

Finally, we would like to thank the organizers for the suggestion to hold this meeting, for their perseverance to organize it under the pandemic restrictions and all the difficulties it brought – it has been very dense two days, but with extremely inspiring presentations and discussions – both in the main talks and in the WGs themselves.

References

1 Abend, O., Rappoport, A. 2013. Universal Conceptual Cognitive Annotation (UCCA). In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). ACL 2013. Sofia, Bulgaria. https://aclanthology.org/P13-1023. Pp. 228-238.

2 Bender, Emily M., Dan Flickinger, Stephan Oepen, Woodley Packard and Ann Copestake. 2015. Layers of Interpretation: On Grammar and Compositionality. In Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015), London. pp. 239-249.

3 Copestake, A., Flickinger, D., Pollard, C., Sag, I. 2005. Minimal recursion semantics: An introduction. Research on Language and Computation. 3(4). 281-332.

4 Flickinger, Dan, Stephan Oepen and Emily M. Bender. 2017. Sustainable Development and Refinement of Complex Linguistic Annotations at Scale. In Ide, Nancy and James Pustejovsky (eds), Handbook of Linguistic Annotation Science. Springer. pp. 353-377.

5 Hajič J. et al. 2020. Prague Dependency Treebank – Consolidated 1.0 (PDT-C 1.0). 2020. LINDAT/CLARIAH-CZ digital library. http://hdl.handle.net/11234/1-3185

6 Hajič J., Bejček E., Hlaváčová J., Mikulová M., Straka M., Štěpánek J., Štěpánková B. 2020. Prague Dependency Treebank – Consolidated 1.0. In: Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020), European Language Resources Association, Marseille, France, ISBN 979-10-95546-34-4, pp. 5208-5218.

7 Nicolai, G., Gorman, K., Cotterell, R. (Editors). 2020. Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology. ACL 2020.

8 de Marneffe, M.-C., Manning, C., Nivre, J., Zeman D. 2021. Universal Dependencies. In: Computational Linguistics, ISSN 1530-9312, vol. 47, no. 2, pp. 255-308.

9 Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C., Pyysalo, S., Schuster, S., Tyers, F., Zeman, D. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020), pp. 4034-4043, European Language Resources Association, Marseille, France, ISBN 979-10-95546-34-4.

10 Oepen S., Abend O., Abzianidze L., Bos J., Hajič J., Hershcovich D., Li B., O'Gorman T., Xue N., Zeman D. 2020. MRP 2020: The Second Shared Task on Cross-Framework and Cross-Lingual Meaning Representation Parsing. In: Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing, Copyright © Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-952148-64-4, pp. 1-22.

11 Sag, I. A.. Wasow, T., Bender, E. M. 2003. Syntactic Theory: a formal introduction, Second Edition. Chicago: University of Chicago Press.

12 Syllak-Glassman, J. 2016. The Composition and Use of the Universal Morphological Feature Schema (UniMorph Schema). Report, Working Draft v.2. CLSP. Johns Hopkins University.

5 Open problems

5.1 Harmonizing semantic representations in multilingual grammar engineering & otherwise

Emily M. Bender (University of Washington – Seattle, US)

Semantic harmonization refers to the process of designing semantic representations such that they are similar across languages, to the extent possible, while still staying true to the ".*" specific to each language. In that context, the following are points for discussion:

- What are the constraints that lead us to want to harmonize/what are the use cases for harmonized representations? Possible answers here include resources like the Grammar Matrix, Grammar-informed machine translation and downstream tasks that can handle different languages.

- What are the constraints that lead us to keep things language specific? Possible answers here include considerations of grammar design, considerations of annotation schema design and the annotation process itself.
- Finally, there are questions that can help situate the broader goal of harmonizing semantic representations and thus potentially help achieve it or at least approach it productively: At what point are we making these design decisions? Is it sensible to try to have one set of conventions? Across what domain (a multilingual project, multiple multilingual projects)? Who might benefit from such a set of conventions?

5.2 Idiosyncrasy and Derivational Morphology: From Distribution to Annotation

Cem Bozşahin (Middle East Technical University – Ankara, TR)

Joint work of Cem Bozşahin, Ozkan Aslan, Burcu Can, Arzu B Guven, Umut Ozge, Tzu-Ching Kao, Utku C Kunter, Gizem N Ozdemir, Barbaros Yet

Derivational morphology is commonly considered to be one morphological formant of the lexicon. Even in languages in which it is very productive, it is considered not as free as phrasal combination, and less compositional than inflection. It can become lexicalized, and idiosyncratic. For example, the Turkish word in (a) is formed by a very productive affix, -*lık*, with allomorphs -*lik,* -*lük,* -*luk*, but nowadays considered to be idiosyncratic; cf. its productive use in (b), and phrasal scope in (c), both of which are compositional. (DV: denominal verb; IRR: Irrealis; NN: noun to noun derivation.)

(1) a. *bakanlık*
bakan-lık
attend-NESS
'ministry' lit. bak-an-lık: see-REL-NESS Turkish

b. *gözlemlenebilirlik*
göz-lem-le-n-ebil-ir-lik
eye-NN-DV-PASS/REFL-ABIL-IRR-NESS
'observability'

c.

[*her*	*konuda*	*resmi*	*görüş-lü*]*-lük*	*sana*	*yakışmadı.*
every	topic	official	view-with-NESS	you-DAT	fit-NEG-PAST

'Officialese in every matter is not you.'

The question of unpredictability of combined meaning is pervasive crosslinguistically. Finnish is known for its agglutinating word structure. [4] report a case study in which Finnish children were asked about meaning of complex words, which is scored by a panel of linguists. Third graders scored better in low-frequency complex words if they happen to have highly productive affixes. Sixth graders were better in low-productive affixes if root frequencies were higher. It seems that idiosyncrasy in word structure needs time to settle in.

We can put this finding in a more general perspective in the microcosm of agglutination with results from Turkish. It has been known since [2] that Turkish children rarely go against adult's ordering of affixes in a word. [3] has found that in a database of 18–36 month old children [10], frequency of affixes in child-speech are proportional to that in child-directed speech. Whether the child uses the derivational ones as conventional/idiosyncratic as the adult's is a question of interest.

In an effort to add adult performance to these findings, we have made two pilot studies on annotated Turkish corpora, reported in [5, 7]. Turkish has more than one hundred derivational affixes (see [6]/2014 for a full list). We chose the ten most frequent from the annotations, judged from annotation's selection of the analyses of word forms. We morphologically disambiguated a set of complex words with the ten derivational affixes. This step gives us correct stem forms of the chosen affixes.

In order to make maximal use of meaning annotations, we analyzed the verbal stems of the affixes to explore semantic properties of the verbal stem in annotation banks, to understand the affixs' selectivity in a stem (agent, patient, beneficiary etc. for thematic annotations such as Proposition Bank, kinds of scenes and participants in UCCA of [1], dependencies in UD frameworks). One such verb-to-verb derivation that we analyzed is (it also has deverbal noun interpretation):

(2)	gel	come	gel-iş	grow/develop	Turkish
	kaç	escape	kaç-ış	run away	
	sığ	fit	sığ-ış	accomodate	
	böl	divide	böl-üş	share	
	koş	run	koş-uş	scurry	
	kok	smell	kok-uş	rot	

As a first approximation we performed unsupervised clustering in the Proposition Bank of [8] and UD Bank of [9], using three methods: k-means, agglomerative clustering, and Gaussian mix. We then translated the verbal stems to English, in an effort to find more results in larger proposition banks of English, with the assumption that semantically the verb stem itself may have similar cross-linguistic conceptual ontology, theme, or dependency properties, although affixing to verb forms is Turkish-specific. We have also performed similar analyses on UCCA-annotated Turkish database.

We have observed no clustering of features. Manual dimension reduction techniques, for example conflating theme and experiencer, did not help.

In retrospect, it seems that trying to infer the meaning of complex words from annotations by unsupervised methods is too much to expect from labels, given current trends in annotation, in which clause and phrase meanings are the targets, and in approximation. Derivational meanings are more conventional than thematic or compositional, and annotators can hardly be expected to pay attention to subleties of conventional use, or have access to guidelines for their annotation.

One case in point is the convention of using "run' above in its derived forms. Consider a scenario where the schoolbell rings and kids run to their classes. As native speaker, I would not use "scurry' alone and say *çocuk-lar koşuş-tu* (kids scurried), because it would mean disorganized and unpurposeful action. I would use *çocuk-lar koş-uş-tur-du* (kids run-COLL-CAUS-PAST), to mean kids ran around for a reason, with the collective and causative imposing some kind of purposefulness. Notice that now the derived verbal stem is compositional, considered to be collective run.

One can hardly expect annotators attending to clausal meaning to distinguish the two uses of *koşuş*. It is also questionable whether guidelines can be set up for the task. It seems that we need an entirely different mechanism or sources to capture these differences manifesting themselves as conventions, that is, higher-order uses of grammatical elements, which are idiosyncratic references to events.

References

1 Abend, O. and A. Rappoport (2013). Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 228–238.

2 Aksu-Koc, A. A. and D. I. Slobin (1985). The acquisition of Turkish. In D. I. Slobin (Ed.), *The Crosslinguistic Study of Language Acquisition, vol.I: The Data*. New Jersey: Lawrence Erlbaum.

3 Avcu, E. (2014). Nouns-first, verbs-first and computationally easier first: A preliminary design to test the order of acquisition. Master's thesis, Cognitive Science department, Middle East Technical University (ODTÜ), Ankara.

4 Bertram, R., M. Laine, and M. M. Virkkala (2000). The role of derivational morphology in vocabulary acquisition: Get by with a little help from my morpheme friends. *Scandinavian Journal of Psychology 41*(4), 287–296.

5 Kunter, U. C., G. N. Özdemir, and C. Bozşahin (2020). Distributional and lexical exploration of semantics of derivational morphology. In *Proc. of Int. Symp. on Brain and Cognitive Science, ISBCS 2020*, Ankara.

6 Oflazer, K., E. Göçmen, and C. Bozşahin (1994). An outline of Turkish morphology. Technical report, METU and Bilkent Univ. re-issued in 2014.

7 Özdemir, G. N. (2021). Distributional investigation of some frequent Turkish derivational affixes for exploring their semantics. Master's thesis, Middle East Technical University. Cognitive Science Dept., Ankara.

8 Şahin, G. G. and E. Adalı (2018). Annotation of semantic roles for the Turkish proposition bank. *Language Resources and Evaluation 52*(3), 673–706.

9 Türk, U., F. Atmaca, Ş. Özateş, G. Berk, A. Köksal, and A. Özgür (2020). Resources for Turkish dependency parsing: Introducing the BOUN treebank and the BoAT annotation tool. *Arxiv preprint 2002.10416.*

10 Ural, A. E., D. Yuret, F. N. Ketrez, D. Koçbaş, and A. C. Küntay (2009). Morphological cues vs. number of nominals in learning verb types in Turkish: The syntactic bootstrapping mechanism revisited. *Language and Cognitive Processes 24*(10), 1393–1405.

5.3 How to best account for the semantics of MWEs?

Voula Giouli (Athena Research Center, GR)

Joint work of Voula Giouli, Aggeliki Fotopoulou
Main reference Aggeliki Fotopoulou, Voula Giouli: "MWEs and the Emotion Lexicon: Typological and cross-lingual considerations", Language Science Press, 2018.
URL http://dx.doi.org/10.5281/zenodo.1182591

The identification of MWEs involves lexical, morphosyntactic and semantic criteria (Gross 1982; 1998b; Lamiroy 2003), to be taken into account, namely: *non-compositionality*, i.e., the meaning of the expression cannot be computed from the meanings of its constituents; *non-substitutability*, i.e., at least one of the expression constituents does not enter in alternations at the paradigmatic axis; and *non-modifiability*, in that they enter in syntactically rigid structures, posing further constraints over modification, transformations, etc. However, the criteria mentioned do not apply in all cases in a uniform way, and the variability attested brings about the notion of *degree of fixedness* (Gross 1996). In this regard, idiomatic expressions bear a meaning that cannot be computed based on the meaning of their constituents and the rules used to combine them. Light verb constructions (LVCs), on the other hand, have a rather transparent meaning due to the presence of the predicative noun (Npred) which retains its original sense.

The problem posed can be defined as follows:

(a) What is the best way to represent the semantics of MWEs – more precisely VIDS (verbal idiomatic expressions) and LVCs – in a uniform way?

(b) One step further, the limits between LVCs and VIDs are in some cases fuzzy: despite the semantic transparency in LVCs (entailed by the Npred) the overall structure is often susceptible to a number of constraints as shown in the example, and the overall semantics of the final expression is less transparent:

(1) πετάω από χαρά

petao	*apo*	*chara*
fly.1SG	from	happiness.ACC.SG

“to be very happy”

What are the criteria to account for these fuzzy cases?

(c) Is mapping of a MWE to a concept sufficient for representing its sense efficiently? esp. when a (near-)synonymous single-word expression exists? For example, how to account for the VID in (2) and its near-synonymous single-word verb?

(2) κάνω σκόνηά

kano	*skoni*
make.1SG	dust.ACC.SG

“to defeat thoroughly”

(d) What other types of semantic representation are feasible, i.e, semantic role labelling? And how to account for a sound annotation of MWEs? What are the best practices for assigning semantic roles to syntactic constituents (non-fixed elements) of MWEs (syntax-semantics interface), esp. with regard to cross-linguistic issues?

(3) Τον τρώει η ζήλιαά

Ton	*troi*	*i*	*zilia*
Him.3.SG.ACC.EXPERIENCER	eats.3.SG.NM	the.SG.NOM	jealousy.SG.NOM

“He is very jealous.”

References

1 Gross, Maurice. 1982. Une classification des phrases “figées”du français. *Revue Québécoise de Linguistique (RQL)* 11(2). 151–185.

2 Gross, Maurice. 1998a. La fonction sémantique des verbes supports. *Travaux de linguistique* 37. 25–46.

3 Gross, Maurice. 1998b. Les limites de la phrase figée. *Language* 90. 7–23.

4 Lamiroy, Béatrice. 2003. Les notions linguistiques de figement et de contrainte. *Lingvisticae Investigationes* 26(1). 1–14.

5.4 MWE Identification using Embedding-based Approaches

Tunga Güngör (Bogaziçi University – Istanbul, TR)

Joint work of Zeynep Yirmibeşoğlu, Tunga Güngör
Main reference Zeynep Yirmibesoglu, Tunga Gungor: "ERMI at PARSEME Shared Task 2020: Embedding-Rich Multiword Expression Identification", in Proc. of the Joint Workshop on Multiword Expressions and Electronic Lexicons, MWE-LEX@COLING 2020, online, December 13, 2020, pp. 130–135, Association for Computational Linguistics, 2020.
URL https://aclanthology.org/2020.mwe-1.17/

Rather than being directly related to the annotation of multiword expressions (MWE), this talk is on the computational side of processing of MWEs. We present a general schema for automatically identifying MWEs using embedding-based approaches. Normally, in all computational models based on the deep learning paradigm, the input is represented in terms of embeddings. An embedding is a short vector (a vector of dimensions typically between 100 and 500) that is used as a representation of a particular entity (e.g. a word). These embedding-based approaches can be used in a multilingual context.

We have used three embedding-based models in previous research for MWE identification in the scope of the PARSEME shared task (Ramisch, et al. (2020)). The basic model is named as ERMI (embedding-rich MWE identification). Two of the models are supervised, while one of them is a semi-supervised model. In that research, as the deep learning model, LSTM-CRF network was used. The details of the model are not important for this talk; other neural network models can also be used. We focus on the embeddings in these models. In the first model, the input for each word is formed of the concatenation of three types of information: the word itself, its part-of-speech, and its dependency relation to the head word in the UD treebank (Nivre, et al. (2016)). Each of these three parts is formed of embeddings. In this way, for a word, the input consists of both morphological and syntactic information. In the second model, a fourth component is added to the input, which is the head word of this word (i.e. its embedding). All these embeddings are learned from a corpus for a language. These two models use annotated corpus. The third model is different in the sense that there is an additional raw (not annotated) corpus, which is much larger than the annotated corpus. A MWE identification model is trained as in the other models, then this model is used to annotate the raw corpus. Then this much larger annotated corpus is used for learning a MWE identification model.

To conclude: These models can be used as multilingual NLP tools in computationally tractable ways. We have tested such tools in about 15 languages having different typologies. And they showed promising results in detecting MWEs.

The open question in this research is: What should be good input representations in terms of embeddings for different types of languages?

References

1 Ramisch, C., Savary, A., Guillaume, B., Waszczuk, J., Candito, M., Vaidya, A., Mititelu, V.B., Bhatia, A., Inurrieta, U., Giouli, V., Gungor, T., Jiang, M., Lichte, T., Liebeskind, C., Monti, J., Ramisch, R., Stymne, S., Walsh, A. and Xu, H., Edition 1.2 of the PARSEME Shared Task on Semi-supervised Identification of Verbal Multiword Expressions, Joint Workshop on Multiword Expressions and Electronic Lexicons (MWE-LEX 2020) at COLING 2020, Barcelona, p.107-118, December 2020.

2 Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D., Universal Dependencies v1: A Multilingual Treebank Collection, Proceedings of International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Portoroz, Slovenia, p.1659–1666, 2016.

5.5 Harmonizing Semantic Representations

Jan Hajič (Charles University – Prague, CZ) and Daniel Zeman (Charles University – Prague, CZ)

Joint work of Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajič, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, Daniel Zeman

Main reference Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, Daniel Zeman: "MRP 2020: The Second Shared Task on Cross-Framework and Cross-Lingual Meaning Representation Parsing", in Proc. of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing, CoNLL Shared Task 2020, Online, November 19-20, 2020, pp. 1–22, Association for Computational Linguistics, 2020.

URL http://dx.doi.org/10.18653/v1/2020.conll-shared.1

While there are very good examples of harmonized representations on lexical, morphological and syntactic levels (UniMorph, Universal Dependencies), work is still ongoing on various semantic representations (AMR/UMR, Prague Dependency Treebanks, UCCA, DMR, DRT, RMS, PMB, DM, ...), as exemplified, for example in the MRP Shared Tasks in 2019 and 2020. If ever possible, such a harmonization should work across formalisms and across languages. However, even the term "semantics" is understood wildly differently by the authors of the various existing representations – from just slightly above syntax (PDT, Enhanced UD) all the way to logic (e.g., PMB) or knowledge representations and ontologies. The related presentation(s) (see the WG3 slides and talks) look at it from different points of view. Perhaps a similar effort could be launched, like UD, for semantic representation and annotation...?

5.6 VMWE degree modification: somewhere between "compositional" and "idiomatic"

Stella Markantonatou (Athena Research Center, GR)

This discussion is about issues regarding the implications of the phenomenon of degree modification (achieved with modifiers) for the lexicographic codification and UD annotation of VMWEs. Degree modifiers may apply on the verb head (1a) or on a lexicalized phrase (fixed subject (1b), object (1c), complement of the copula).

(1) a. *δάγκωσα* ***για τα καλά/γϵρά/άσχημα*** *τη λαμαρίνα*
I.bit for good/strongly/ugly.**ADVERB** the tin

"I fell for somebody"

b. *ϑα σου πιουν* ***όλο*** *το αίμα*
will you.GEN they.drink all.**DET** the blood

"they will exhaust you;;

c. *βγήκϵ* ***μϵγάλη*** *βρώμα πριν λίγο στο Τρωκτικό για Σάββα*
came.out big.**ADJECTIVE** dirt before little in.the Rodent for Savvas

"very nasty news about Savvas has just been published in the "Rodent""

The distribution of adverbs such as κυριολεκτικά (“literally”), πραγματικά, πράγματι (“really”), ειλικρινά (“sincerely”) and PPs such as στην κυριολεξία (“literally”), στ′ αλήθεια (“truly”) seems to be determined by discourse only. These adverbs are used to confirm the true of the denotation of the utterance that contains the VMWE; it is precisely the speaker’s commitment about his utterance that creates the intensification effect (Mexa and Markantonatou 2020; Israel 2002; Paradis 2003; Bordet 2017).

The adverbs εντελώς, τελείως (“completely”) apply to VMWEs that are closed scale predicates (Kennedy and McNally 2005; Gavriilidou and Giannakidou 2016; Mexa and Markantonatou 2020). Manner adverbs are used for degree modification of VMWEs (and verbs) (Κλαίρης και Μπαμπινιώτης 2004: 857): άγρια (“wildly”), άσχημα (“ugly”), για τα καλά (“for good”), γερά, δυνατά (“strongly”), etc. Mexa and Markantonatou (2020) found them with VMWEs from the semantic domains of ANGER (2) and LOVE but not of SURPRISE. They possibly form collocations with certain VMWEs.

(2) *αρχίζω και φορτώνω πολύ/άσχημα/ϵπικίνδυνα/;;γϵρά*
I.start and I.load ugly/dangerously/??strongly

“I am getting dangerously angry”

Modification of a lexicalized NP is described below. Certain adjectives, such as ανήμερος (“untamed”) (3), τσουχτερός (“biting”) seem to form **collocations**.

(3) *έγινϵ ϑηρίο ανήμϵρο*
he.became beast untamed

“he got furious”

Μεγάλος (“large”, “big”), τεράστιος (“huge”) (4a), (4b) have less constrained distribution; other adjectives have a messy distribution probably determined by the ““literal” meaning of the noun. The definite and the indefinite article, the determiners όλος (“all”) (1b), πολύς (“much”, “a lot”), and the conjunction και (“and”) can be used as intensifiers:

(4) a. *έφαγα μϵγάλη/τϵράστια/τρϵλλή/άγρια/*βαριά φρίκη*
I.ate large/huge/mad/wild/*heavy horror

“I went through a horrendous experience”

b. *μου ήρϑϵ μϵγάλη/τϵράστια/βαριά/χοντρή/;;τρϵλλή κϵραμίδα*
to.me came large/huge/heavy/fat/?mad rooftile.NOM

“I experienced a big unpleasant surprise”

In sum: (i) Modification by adverbs of the type of “literally” is not determined by the semantics of the VMWE (ii). Certain, but not all, adjective+noun combinations strongly collocate (iii). In between are several adverb+verb, adjective+noun combinations.

A. What should be encoded in a lexicon?

- A collocation/combination as an independent VMWE?
- A collocation/combination as a variation of the “original” VMWE that could be derived from it via modification with (very often) prespecified adjectives/adverbs? Consider the ϑηρίο (“beast”) set of VMWEs:

(5) a. *έγινα θηρίο*
became.1.SG beast

"I got very angry"

b. *έγινα άγριο/σωστό/πραγματικό θηρίο*
became.1.SG wild/right/real beast

"I got very, very angry"

c. *έγινα θηρίο ανήμερο*
became.1.SG beast untamed

"I got furious"

B. Degree modification in UDs

- "Degree modification using modifiers (not using morphology)" ... but it is reminiscent of degree modification via morphology, e.g.:

(6) a. *χέρι μικρό χέρι/χερ-άκι*
hand

b. *μεγάλο χέρι/χερ-ούκλα*
large hand/hand-magnifier

UD at the moment have only one morphology feature Dim(inutive) and only for Afrikaans[5] that allows to connect the lemma of a noun e.g. χέρι with a diminutive form e.g. χεράκι. Probably something similar could be defined for the "beast"-set (5) and its kin.

References

1 Bordet, Lucile. "From vogue words to lexicalized intensifying words: the renewal and recycling of intensifiers in English. A case-study of very, really, so and totally." Lexis. Journal in English Lexicology 10 (2017).

2 Ernst, Thomas. "Grist for the linguistic mill: Idioms and "extra'adjectives." Journal of Linguistic Research 1.3 (1981): 51-68.

3 Gavriilidou, Zoe, and Anastasia Giannakidou. "Degree modification and manner adverbs: Greek: poli "very" vs. kala "well'." Selected papers on theoretical and applied linguistics 21 (2016): 93-104.

4 Israel, Michael. "Literally speaking." Journal of Pragmatics 34.4 (2002): 423-432.

5 Kennedy, Christopher, and Louise McNally. "Scale structure, degree modification, and the semantics of gradable predicates." Language (2005): 345-381.

6 Mexa, M., and S. Markantonatou. "Intensifiers/moderators of verbal multiword expressions in Modern Greek." EURALEX XIX (2021).

7 Paradis, Carita. "Between epistemic modality and degree: the case of really." Modality in contemporary English. De Gruyter Mouton, 2012. 191-222.

8 Κλάιρης, Χρήστος, ανδ Γεώργιος Δ. Μπαμπινιώτης. Γραμματική της νέας ελληνικής: δομολειτουργική-επικοινωνιακή. Το όνομα της νέας Ελληνικής. Ι. ὅλ. 1. Ελληνικά Γράμματα, 1996.

[5] https://universaldependencies.org/af/feat/Degree.html

5.7 Projects are humans – on the trade-off between complexity and diversity

Carlos Ramisch (Aix-Marseille University, FR)

Initiatives to create highly multilingual resources for morphological, syntactic and semantic processing abound nowadays in the computational linguistics community. Among these initiatives, three are represented in this seminar: UniMorph for morphology, Universal Dependencies (henceforth UD) for dependency syntax, and PARSEME for multiword expressions. These projects deal with the representation, especially in annotated corpora, of complex linguistic phenomena. To model these phenomena, one reasonable assumption is that one should use different layers to group phenomena according to their similarity into a single layer whereas phenomena that seem too distant are pushed to an upper/lower layer.

One of the most concrete examples of this approach is UD's CoNLL-U format. An annotated corpus file in CoNLL-U format contains 10 columns which can be seen as more or less independent layers representing the segmentation, form, lemma, POS, morphological features, dependencies etc. of naturally occurring text. This mechanism can be generalised further, as in the "CoNLL-U plus" format[6], which was adapted by PARSEME to add an 11th column representing MWEs in the CUPT format.[7]

Layers are very useful because they allow focusing on a single subproblem of language representation at each time inside an ambitious annotation project such as UD. In addition, different projects can then deal with different groups of phenomena, such as UniMorph, UD and PARSEME, thus manipulating autonomous layers in parallel without conflicts. Different layers can use different underlying structures to represent language: subword morphological features, word-level POS tags, dependency graphs, MWE subsequences (or sub-graphs), etc. Layers of different levels can be connected using unique IDs and references. This helps accounting for the complexity of language in a modular way that is also very convenient for computational processing.

On the other hand, a layered approach to language annotation and representation poses two major challenges. First, it is hard to define completely hermetic layers, especially in the light of cross-layer phenomena such as multiword expressions, which challenge the traditional borders between lexicon, syntax and semantics (or even morphology, e.g. idiomatic compounds). Therefore, different initiatives may decide to represent a single phenomenon in a language in different ways. For instance, while UD sees verb-particle constructions in English (e.g. *make up*) as a type of compound, PARSEME rather models the non-compositional nature of the particle modifier, providing different tests and scope to annotate the same phenomenon. This introduces redundancy when we bring layers together: many verb-particle constructions will be annotated twice, once in UD and once in PARSEME, with sometimes inconsistent decisions.

While consistency and redundancy are well known technical issues in large annotation projects, there is a second problem that arises from the complexity of multi-layered approaches. Suppose a new language wants to join UD and PARSEME, and that both projects are now completely integrated, with MWE annotation representing an extra layer over UD's morpho-

[6] https://universaldependencies.org/ext-format.html
[7] http://multiword.sourceforge.net/cupt-format/

syntactic layers. Even if all consistency and redundancy issues were solved, annotating all these 11 layers at once would still be extremely complex, especially for new annotators. Now suppose that not only MWEs but also finer morphological annotation is included as an extra layer (e.g. to introduce morpheme-based tags). Besides, more abstract semantic and pragmatic layers can be added such as abstract meaning representation semantics trees, named entities, terms, and so on, each on a separate layer. Reading, understanding and becoming familiar with the whole annotation guides of all these layers might sound like a scary task, so the **access cost** to this new integrated annotation project would increase.

The open issue at hand here could be summarised in the following question: *how can we account for the complexity of language in our multi-layered projects without increasing the access cost for new languages beyond what would be acceptable?* In other words, given the "universal" nature of these projects, we must keep in mind that the diversity of languages covered is a crucial aspect. Resource creation and enhancement cannot be limited to a small group of initiated project members who have been there for enough time to master the maze of annotation layers and guides.

To decompose the question into more focused ones, we can think about several aspects that can make a resource creation project attractive. First, the benefits for contributors must be made visible from the beginning. Most of these projects cannot fund their members, so the benefits are rather indirect. They include co-authoring papers on the topic, networking, improving the resources for their own languages, and being supported by a large international initiative in their local grant applications.[8]

Second, the management of the community must be well designed. It may require some structure, some specific roles, so that new members know who to contact when they get lost. New members should be able to get quick assistance, so that they do not feel discouraged by the weight of all the layers.

Third, collaborative projects usually require a deep sense of community. Members have to feel engaged, belonging to something pleasant, interesting, dynamic. For instance, how can we ensure that the discussions carried out on git issues are taken into account into the guidelines? Should the guidelines point to the discussions (e.g. git issues) that led to given a decision? How can we make guidelines evolve so that experts feel listened to, without requiring prohibitive updates to existing data? Creating and keeping this feeling of belonging, especially in the context of the current pandemic crisis, is a real challenge for these communities.

Fourth, a certain number of tools and resources can make integration of a new language less traumatic for newcomers. These include online forums, chat platforms, shared software infrastructure, tutorials, videos, zoom calls with more experienced members, and probably many more tools that we still have to imagine and develop. While free tools can be used in many cases, developing specific tools tailored to our needs can also make a difference, although it is not always easy to request funding for software development and engineering in research grant applications.

In short, the question of how to integrate (many) annotation layers in resources created by different communities to account for the complexity of linguistic phenomena poses real challenges for (computational) linguistics. In addition to the well known problems of redundancy and consistency, it is important to put some effort into keeping these projects welcoming and inclusive so that the **access cost** for new languages is kept to a reasonable minimum, helping keep diversity at the core of these initiatives. This provides us with a

[8] See an example from PARSEME: [9]

great opportunity to think out of the box and employ our creative energy to come up with innovative solutions that make people feel committed and engaged into connecting their resources and putting their linguistic expertise at the service of cross-linguistic connections and deeply multilingual computational applications.

5.8 Multiword expressions as multiword constructions

Manfred Sailer (Goethe-Universität Frankfurt am Main, DE)

Joint work of Manfred Sailer, Sascha Bargmann

The formal modelling of idioms has been alternating between phrasal/holistic and lexical/combinatorial approaches. At least in HPSG and SBCG, a combinatorial analysis seems to have been widely recognized recently. I think that, notwithstanding the important arguments in favor of a combinatorial modelling, the unit-like character of an MWE should be accounted for as well. I see two main flaws of combinatorial analyses: First, they fail to capture the fact that the MWE-specific reading of the components of the complex expression should not be represented outside the MWE. Second, the literal meaning of MWE components is accessible for metaphoric and other processes even in non-decomposable MWEs.

I will sketch an attempt of a solution to this dilemma, which may at least work for HPSG, though it is an open question what this would mean for other frameworks, for corpus annotation, or parsing.

Summary of the presentation

The formal modelling of idioms has been alternating between phrasal/holistic and lexical/combinatorial approaches. At least in HPSG and SBCG, a combinatorial analysis seems to have been widely recognized for all syntactically regular MWEs recently. I think that, notwithstanding the important arguments in favor of a combinatorial modelling, the unit-like character of an MWE should be accounted for as well.

Combinatorial modellings are very well equipped to capture the fact that MWEs of the same degree of decomposability differ with respect to their syntactic flexibility across languages. This observation has been made already in Nunberg et al. 1994, and more systematically in Schenk 1995. Bargmann & Sailer 2018 show that it can follow directly from a parallel specification of the lexical entries of the parts of the MWEs (*such as kick the bucket* ‘die’ and its German analogue *den Löffel abgeben* (lit: the spoon away.give)), combined with language-specific characterizations of the fronting constructions. In this approach, the constraints on the syntactic flexibility of MWEs follows from the interaction of the lexical entries of their component parts and the analysis of the critical syntactic constellations (passive, fronting, etc).

In so-called phrasal approaches, MWEs are encoded as phrasal units, which means that their “lexical” description contains both information on their component words/morphemes and on their syntactic combination (such as VP for kick the bucket). Such phrasal approaches face problems:

The discourse conditions can be so special that even usually syntactically non-flexible MWEs may appear in a particular constellation, such as non-decomposable MWEs in English passive:

(1) When you are dead, you don't have to worry about death anymore. ... The bucket will be kicked.
(internet example, reported in Bargmann & Sailer 2018:5)

There is interaction with other special constructions, such as the N-after-N construction:

(2) All those people behind them pulling string after string for them
(internet example, reported in Bargmann 2019: chapter 6)

One part of an MWE may be associated with several occurrences of the MWE:

(3) The beans have not been spilled yet, but will be spilled very soon.
(constructed, reported in Sailer & Bargmann 2021)

Parts of an MWE can be pronominalized:

(4) Eventually she spilled all the beans. But it took her a few days to spill them all.
(Riehemann 2001:207)

Webelhuth, et al. 2018 and Bargmann 2019 show how these data can be captured in an approach that reduces the description of an MWE to the description of its component words/morphemes, ignoring their concrete phrasal combination.

However, existing lexical approaches fail to capture the fact that the MWE-specific reading of the components of the complex expression should not be represented outside the MWE (see the criticism in Riehemann 2001). For example, some phenomena seem to link the literal and the MWE-specific (or idiomatic) reading of an MWE. Egan 2008 and Findlay, et al. 2019 discuss so-called extended uses of MWEs as in (5).

(5) If you let this cat out of the bag, a lot of people are going to get scratched.
(Egan 2008: 392)

Here, the MWE *let the cat out of the bag* is interpreted idiomatically with respect to the current world in the if-clause. To make sense of the main clause, we need to interpret the MWE literally, but with respect to some figurative or metaphoric world. Finally, the overall interpretation needs to be mapped back to current world by some analogy between the figurative world and the current world (i.e. revealed secret can hurt many people, just as a released cat can scratch many people). For such a reasoning to be available, we need (i) access to both the literal and the idiomatic interpretation of the MWE, (ii) access to the MWE as a unit.

Ernst 1981 and Bargmann, et al. 2021 show that so-called conjunct modification as in (6) is another instance which requires simultaneous availability of the literal and the idiomatic meaning of an MWE.

(6) With the recession, oil companies are having to tighten their Gucci belts.
(Ernst 1981:60)

In the talk, I sketched Sailer & Bargmann 2021, which treates MWEs as multiword constructions, i.e. as constructions that specify words they contain, but no concrete phrasal syntactic pattern. I suggested a way to expand this to integrate the analysis of data as (5) from Findlay, et al. 2019.

Summary of the discussion

The discussion brought up a number of interesting issues.

First, the notion of a construction was debated. If the current approach is on the right track, we should not only have constructions as complex phrasal patterns that may specify some lexical components, but we should also assume cases where we have fixed lexical components but no pre-determined phrasal pattern.

Second, the treatment in Sailer & Bargmann 2021 is in part similar to the IOB-encoding proposed in Schneider, et al. 2014, but may differ in cases like (3).

Third, the extent to which we find purely idiomatic uses of an MWE, purely literal uses, or combined uses such as those in (5) and (6) is an interesting topic that needs further investigation.

References

1 Bargmann, S. 2019. Chopping up idioms: Towards a combinatorial analysis. Goethe-University Frankfurt a.M. dissertation.

2 Bargmann, Sascha, Berit Gehrke, and Frank Richter. "Modification of literal meanings in semantically non-decomposable idioms." One-to-many relations in morphology, syntax, and semantics (2021): 245.

3 Bargmann, Sascha, and Manfred Sailer. "The syntactic flexibility of semantically non-decomposable idioms." Multiword expressions: Insights from a multi-lingual perspective 1 (2018): 1-29.

4 Egan, Andy. "Pretense for the complete idiom." Noûs 42.3 (2008): 381-409.

5 Ernst, Thomas. "Grist for the linguistic mill: Idioms and "extra'adjectives." Journal of Linguistic Research 1.3 (1981): 51-68.

6 Findlay, Jamie Y., et al. "Why the butterflies in your stomach can have big wings: combining formal and cognitive theories to explain productive extensions of idioms." Talk given at the EUROPHRAS 2019 Productive Patterns in Phraseology Conference. Vol. 24. 2019.

7 Nunberg, Geoffrey, Ivan A. Sag, and Thomas Wasow. "Idioms." Language 70.3 (1994): 491-538.

8 Riehemann, Susanne Zalta. A constructional approach to idioms and word formation. stanford university, 2001.

9 Sailer, M. & S. Bargmann. 2021. A phraseo-combinatorialanalysis of idioms. Talk presented at HPSG 21.

10 Schenk, André. "The syntactic behavior of idioms." Idioms: Structural and psychological perspectives (1995): 253-272.

11 Webelhuth, Gert, Sascha Bargmann, and Christopher Götze. "Idioms as evidence for the proper analysis of relative clauses." Reconstruction effects in relative clauses. De Gruyter (A), 2018. 225-262.

5.9 Word and resources for little-resourced languages

Emmanuel Schang (University of Orleans, FR)

The definition (its boundaries and the method to discover it) of word has been identified as a difficult topic for a very long time. These difficulties can be found in the Cours de Linguistique Générale (Saussure 1915):

> "En résumé la langue ne se présente pas comme un ensemble de signes délimités d'avance, dont il suffirait d'étudier les significatiosnet l'agencement ; c'est une masse indistincte où l'attention et l'habitude peuvent seules nous faire trouver des éléments particuliers. L'unité n'a aucun caractère phonique spécial, et la seule définition qu'on puisse en donner est la suivante : une tranche de sonorité qui est, à l'exclusion de ce qui précède et de ce qui suit dans la chaîne parlée, le signifiant d'un concept. [. . .] Cependant nous sommes mis immédiatement en défiance en constatant qu'on s'est beaucoup disputé sur la nature du mot, et en y réfléchissant un peu, on voit que ce qu'on entend par là est incompatible avec notre notion d'unité concrète.

De Saussure concludes in an optimistic way and gets around the problem of the definition of word by saying that it is not a necessary unit:

> "Lorsqu'une science ne présente pas d'unités concrètes immédiatement reconnaissables, c'est qu'elles n'y sont pas essentielles. En histoire, par exemple, est-ce l'individu, l'époque, la nation? On ne sait, mais qu'importe? On peut faire oeuvre historique sans être clair sur ce point."
> [. . .]
> "La langue présente donc ce caractère étrange et frappant de ne pas offrir d'entités perceptibles de prime abord, sans qu'on puisse douter cependant qu'elles existent et que c'est leur jeu qui la constitue. C'est là sans doute un trait qui la distingue de toutes les autres institutions sémiologiques."

More than a century later, Haspelmath (2017) concludes "that we do not currently have a good basis for dividing the domain of morphosyntax into morphology and syntax, and that linguists should be very careful with general claims that make crucial reference to a cross-linguistic "word' notion."

Having this in mind, any annotation of corpus based on the notion of word has to be taken with caution.

This is true for well-known languages, but crucial for little-known languages. In absence of a standardized writing system, the linguist makes theoretical choices which frequently conflict with the speakers "intuitions' and practice. This has been well described in Hazaël-Massieux (1993) for the Antillean Creoles for instance.

This becomes a real problem when building a treebank for little-known languages since the resources are rare and expensive, and the opportunities to recode it (new segmentation etc.) is difficult and costful. The necessities of a particular project often leads to a particular coding which too often blocks the reuse of the resource.

One can wish that a coding of a resource is not destructive and that an alternative coding could be considered.

References

1 Haspelmath, M. (2017). The indeterminacy of word segmentation and the nature of morphology and syntax. Folia linguistica, 51(s1000), 31-80.

2 Hazaël-Massieux, M. C. (1993). écrire en créole: oralité et écriture aux Antilles. éditions l'Harmattan.

3 Saussure, F. De (1915). Cours de linguistique generale (Payot, Paris).

5.10 Listen to the data: comprehensive MWE annotation and emergent challenges in English UD

Nathan Schneider

Overview

STREUSLE Corpus:

- Lexical semantic annotation of MWEs in English reviews
- Bottom-up, comprehensive: no preconceived notion of which categories/types we were looking for; original annotators did not see syntax:
  - Lots of variety! Not just verbal and nominal MWEs – also PPs, functional expressions, etc.
  - Also discovered some partially productive constructions in this process
- Strong (semantically opaque) vs. weak expressions (˜formulaic expressions, statistically idiomatic)
- Lexcat (lexical category): syntactic subcategorization of strong MWEs (and single-word expressions); adapted from UD UPOS; draws on PARSEME for VMWEs

UD issues investigated in English: better guidelines needed for

- "Mischievous nominal constructions": names, dates, numbers; compounds; adverbial NPs
- Multiword connectives ("out of", "along with", "based on", etc.)

Details and Links

- **STREUSLE:** MWEs can be annotated comprehensively ina corpus, without prefiltering for syntactic status, which unearths all sorts of interesting expressions and constructions. (Schneider et al. 2014)
  - PARSEME efforts thus far have done a great job for **verbal** MWEs across languages, but MWEs in general are syntactically open-ended.
  - STREUSLE annotators received general criteria for what counts as an MWE: **strong** (semantically opaque) or **weak** (formulaic expression). Developed guidelines for specific constructions as we went[10].

[10] see: original guidelines, prepositional verbs, PARSEME 1.1 VMWEs

    - Investigation was limited to English. Can this be done on a larger scale and multilingually?
    - Suggestion: for a corpus sample, annotate MWEs bottom-up, without predefined syntactic constraints. Then revise, taxonomize, and harmonize.
    - Description of data format (CONLLU-Lex), **lexcats** to sub-categorize strong expressions
  - Are there annotation tools that make it easy to alternate between consecutive and type-based annotation flows?
  - Tagger trained on STREUSLE, evaluated on MWE corpora
- UD guidelines need clarification/improvement with respect to various productive nominal constructions in English: **"mischievous nominal constructions"**:
  - names, dates, numbers, measurements; adverbial NPs; compounds
  - Better accommodate these with current top-level relations, clarifying boundaries of flat, compound, appos, nmod, nummod, etc. See proposals with Amir Zeldes.
- UD guidelines around **multiword functional connectives** need improvement
  - UD annotators need (at least a small) construction!
    - See: current version.
  - Double case analysis for "out of" etc. Why? See: Issue #795.
  - Deverbal connectives: "according to", "based on", etc. See: EWT issue #179.
    - Validator considers VERB/mark an error.
  - Conjunction-like connectives: "rather than", "instead of", "along with", etc. See: Issue #679.
  - "Next to". See: Issue #496.
- UD needs a way to annotate idioms where the internal head POS does not correspond to the phrase's syntactic distribution, e.g. see `ExtPos=Yes` and other ideas in Issue #807.
- UD unclear on complex determiners: "a few", "a little" (see EWT Issue #170), and in general whether "few" and "many" should be ADJ or DET (see Issue #786)
- Syntax vs. semantics: UD is a sufficiently established standard that we should clarify morphosyntactically tricky aspects of constructions, but keep semantics in a separate layer

5.11 Unifying UD and PARSEME frameworks

Sara Stymne (Uppsala University, SE)

Universal dependencies (UD) is a framework for consistent annotation of morphological features, aprt-of-speech tags and dependency syntax, across different human languages. Version 2.8 covers 114 languages. For each annotation layer, UD contains a pool of categories that languages can use, and it is also possible to add language specific extensions. For more information, see the UD web page[11].

PARSEME (PARSing and Multi-word Expressions), started out as a EU COST action (2013–2017), but the initiative remains. One of the PARSEME activities is to organize shared tasks on the identification of verbal multiword expressions (VMWEs). There has been three

[11] https://universaldependencies.org/

editions in 2017, 2018 and 2020. As part of the shared task, a large annotation effort has taken place for 26 languages (in edition 1.2). PARSEME has general guidlines targeted at all languages, with language specific extensions in a few cases.

Both these initiatives target harmoized annotation of language phenomena across languages, but with differnt phenomena in focus. PARSEME encouraged participants to annotate texts from UD, in order to have basic annotations as well, but there are also other texts annotated in the PARSEME corpora.

The main points for discussion proposed were:

- Is it desirable to unify UD and PARSEME?
- How should it be done technically?
  - The PARSEME CUPT format is an extension of the UD ConLLU format
  - Potentially PARSEME anntoations can be added to UD in a similar way as extended universal dependencies.
- What are the potential relations between the two projects?
- What can the projects learn from each other?

The maybe most important outcome of the discussion in WG3 was that it was seen as desirable to synchronize UD and PARSEME annotations. There seem to be advantages to having resources in the same location, especially when they are annotated on the same texts. While the PARSEME annotations are more semantic than UD, this was not seen as a major issue. There are also "extended universal dependencies" for some UD treebanks, which cover semantic aspects.

We noted that there are some overlap in annotations, especially as language-specific features in UD. Particles are attached to their main verbs with the edge label "compund:prt", and similarly the label "compound:lvc" is used for light verbs. However, these labels are only used for a small number of languages. While a language like English do have LVCs, such as "make a decision", it is not annotated as such in UD, but seen as a regular syntactic construction. We discussed that language-specific constructions will likley mainly be used in languages where such constructions are pervasive. This can be problametic for instance for corpus-studies, when the phenomenon is not represtend equally across languages. It is also the case that some syntactic constructions, like particle verbs, can be used idiomatically, but there are also cases where the semantics are regular, and they would not be annotated as VMWEs in PARSEME.

5.12 Are Chinese idioms Multi-Word Expressions (MWEs)?

Nianwen Xue (Brandeis University – Waltham, US)

Chinese has many (typically four-character) idioms that are based on some ancient stories. Overtime the moral of a story has become the meaning of the idiom. The question of how to identify MWEs are intricately linked to the question of wordhood. One key test for MWEs is non-compositionality, which is also a key test for wordhood in Chinese. An idiom is by definition non-compositional. Does that mean that all idioms are words (thus no longer MWEs)? Or are cases where idioms can still be MWEs? Answers to these questions are crucial to arriving at a definition of MWEs that are cross-linguistic applicable.

PARSEME definitions of Multiword Expressions (MWEs):

- Some degree of orthographic, morphological, syntactic or semantic idiosyncrasy with respect to what is considered general grammar rules of a language.
- Their component words include a head word and at least one other syntactically related word. Most often the relation they maintain is a syntactic (direct or indirect) dependence but it can also be e.g. a coordination.
- At least two components of such a word sequence have to be “lexicalized (fixed)” (others are “open slots”).
- How to recognize MWEs: Probably the most salient property of MWEs is semantic non-compositionality, but since non-compositionality is subjective, use inflexibility as proxy.

Chinese has many (typically four-character) idioms that are based on some ancient stories. Over time the moral of a story has become the meaning of the idiom:

(1) 请司空摘星拿主意,无异于缘木求鱼、刻舟求剑毫无 可操作性。
ask

“Asking Sikongzhaixing for ideas is no different from climbing trees to catch fish or carve a mark on boat to find the missing sword, and is not at all practical.”

Other idioms are metaphors that can be very long:

(2) 我哑巴吃黄连有苦说不出。
I.aphastic

“I feel like an aphasic who has taken coptis Chinesis, and cannot speak out even though I am wronged.”

In other cases apparent MWEs may be just discontinuous words:

(3) 我摔了一个大 跤。
I.fall

“I had a big fall / I fell hard.”

Questions for discussion:

- Are Chinese idioms multi-word expressions? To answer that, we need to know how many words these idioms have.
- How what counts as a word in Chinese, where text is not written with orthographic word boundaries?
- For languages like Chinese where there are no natural word boundaries, tests for MWEs need to be built on tests for wordhood.
- Wordhood in Chinese is a complicated issue, and there are many factors involved: morphophonological, syntactic, semantic, lexical, rhythmic, etc.
- Any definitions or classifications of MWEs that are cross-linguistically applicable would have to take into account languages that do not have orthographical word boundaries, and the determination of MWEs cannot be easily separated from the determination of wordhood.

- Coming up with consistent criteria to identify MWEs helps with consistent annotation of syntactic (e.g., UD) and semantic representations (e.g., AMR/UMR).
- Thinking about the syntactic / semantic structure can sometimes crytalize our judgments of MWEs.

5.13 Issues in UD Consistency, Phrases, and Semantics – With a Focus on Double Subjects and Nested Copula Predication

Amir Zeldes (Georgetown University – Washington, DC, US)

In this talk I give an overview of Universal Dependencies-related activities at Georgetown University, with a focus on annotation problems encountered while annotating the UD English Georgetown University Multilayer corpus (GUM) and some new UD data in Hebrew. I focus on the problematic guideline advocating the annotation of nesting copulas with a top-level clause headed by the copula and governing the nested clause as a complement clause (ccomp). I argue that this guideline is linguistically wrong, as it implies that verbs like "be' are transitive; that it is inconsistent, since other nesting functional structures (nesting PPs, adverbial clauses) do not behave this way; that it is incoherent, since it gives radically different sub-graph analyses to different cases of the same construction; and that it is cross-linguistically untenable due to languages with zero copula constructions, which may nest in the same way.

5.14 Wordhood issues: Toward a typology

Tim Zingler (University of New Mexico – Albuquerque, US)

Main reference Zingler, Tim: "Wordhood issues: Typology and grammaticalization". PhD dissertation, University of New Mexico, 2020
URL https://digitalrepository.unm.edu/ling_etds/71/

Researchers in both phonology and morphosyntax agree that the structural unit of the "word" is difficult to define on the basis of concrete formal criteria (e.g., Bickel et al. 2009; Haspelmath 2011). What is of particular interest is that this issue manifests itself in both language-specific and cross-linguistic work (e.g., Schiering et al. 2010; Tallman 2021). One response to this impasse has been to posit two different word units, a phonological (or prosodic) word and a grammatical (or morphological, or morphosyntactic) word (cf. Dixon 2010: ch. 10). In addition, it has become a convention to classify as "clitics" all those elements that show some kind of mismatch between phonological and grammatical wordhood criteria (cf. Haspelmath & Sims 2010: 198, 202). Yet, since phonological and grammatical words, as well as the mismatches within and between them, are language-specific and variegated, such a coarse classification does not shed much light on the problem or on its causes (cf. also Tallman 2020). The net result of this situation is thus a paradox. The majority of linguists would arguably agree that words are an important "building block" of language, and words have psychological reality even for users of traditionally unwritten languages (e.g., Evans 1995: 62; Mithun 2014: 73). Yet, every effort to define words seems to suggest that they do not in fact exist.

One way to accommodate this conundrum is to approach it from the opposite angle. That is, once it is acknowledged that mismatches between wordhood criteria (henceforth "wordhood issues") are inevitably found in the world's languages, it becomes a major desideratum to typologize these wordhood issues. To the extent that these mismatches tend to involve some combinations of wordhood criteria more often than others, this would suggest that these common mismatches would have to be treated in more depth by linguistic theories than the less common, language-specific outliers. Arriving at such a typology of wordhood issues was the central aim of Zingler (2020), which looked at exponents of definiteness, case, indexation ("agreement"), and tense across 60 unrelated languages from five geographical macro-areas. This focus on the grammatical domain was itself motivated by two major cross-linguistic insights. One the one hand, words tend to lose both their phonological and morphological word properties during the process of grammaticalization (e.g., Bybee et al. 1994; Hopper & Traugott 2003). On the other hand, the four functions selected bear a relatively low degree of relevance to the meanings of their respective nominal or verbal stems, which typically coincides with a lower degree of formal fusion between the stem and the grammatical marker (cf. Bybee 1985). In conjunction, these findings suggest that markers of those four functions are particularly likely to be formally ambiguous, that is, to constitute wordhood issues.

The methodological basis of Zingler (2020) was a set of four criteria of phonological wordhood and four criteria of grammatical wordhood. These eight criteria are displayed in Table 1 and were drawn from the relevant literature because they are sufficiently general to be applicable to a large number of different languages. Furthermore, these criteria were subsumed under the more general concept of "formal dependence," which was in turn defined as the degree to which the shape or distribution of a morpheme is determined by other morphemes. Hence, an element that falls short of any given criterion of wordhood is "dependent" in terms of that criterion. It also follows that an element that is dependent in terms of more criteria of phonological wordhood than of grammatical wordhood is more "prosodically" dependent than "syntagmatically" dependent, abbreviated here as "P > S." The converse, in which a morpheme is dependent on more criteria of grammatical wordhood than of phonological wordhood, is rendered as "S > P" in this work. One major benefit of this approach is that it does not require the vague and unhelpful "clitic" label. Rather, it permits the classification of any given element as a P > S or S > P issue, which then leaves open the possibility to further specify which criteria underlie the relevant mismatch.

The main finding of Zingler (2020) was that P > S issues are considerably more common than S > P issues. The 72 wordhood issues in the database come from 41 of the 60 languages in the sample, which suggests that there might be languages without wordhood issues in the grammatical domain. It is also important to highlight that the predominance of P > S issues holds in each of the four grammatical domains and in each of the five macro-areas investigated. So, the typological generalization seems to be that more prosodic dependence is the norm among grammatical exponents and that more syntagmatic dependence is the exception. Meanwhile, the grammaticalization perspective of this distribution is that the emergence of prosodic dependence typically precedes that of syntagmatic dependence, which can largely be explained by frequency-driven accounts of phonological change such as those by Bybee (2001, 2015). Finally, the S > P issues exclusively occur in contexts of high morphological synthesis, where the relevant exponent will often be syntagmatically fixed before it is fully prosodically reduced. This further suggests that wordhood issues are systematic. Yet, future research will have to establish why synthesis is a necessary rather than a sufficient condition for S > P issues.

While the quantitative discrepancy between P > S issues and S > P issues already helps to constrain the problem of wordhood, a further empirical fact that emerges from Zingler (2020) is of even greater interest. Specifically, 54 of the 63 P > S issues involve an item that has the syntactic distribution of a word (and thus meets the wordhood criterion of non-selectivity) but that falls short of phonological wordhood because it is integrated into a larger domain in terms of prominence and/or allomorphy (and thus violates the wordhood criteria of prosodic features and/or phonological rules). Overall, then, 54 of the 72 wordhood issues are defined by a specific interaction of only three of the eight wordhood criteria investigated. In order to account for this pattern, one might thus define a word as a single domain of prominence and allomorphy in which all non-root constituents are selective. However, it should be emphasized that this definition is no more than a heuristic to be used in cross-linguistic research on phenomena that require some definition of wordhood. Yet, while this definition obviously fails to account for notions such as vowel harmony, this omission derives precisely from the fact that vowel harmony proved to be a marginal indicator of wordhood in most of the languages sampled. Hence, the question of how to treat such language-specific wordhood issues will still have to be left for specialists working on the languages at issue.

Table 1 Wordhood criteria used in Zingler (2020).

phonological word	*grammatical word*
Free occurrence: Word constitutes a well-formed utterance	Cohesiveness: The constituent elements of a word always occur together
Segmental structure: Word is in the domain of phonotactic constraints, minimum weight	Conventionalized meaning: Word is the smallest psychologically real sign for users
Prosodic features: Word is the domain of stress / tone assignment, vowel harmony	Fixed order: The relative position of each morphological unit within a word is fixed
Phonological rules: Word is in the domain of phonologically conditioned allomorphy	Non-selectivity: Words can co-occur with different word classes (unlike affixes)

Table 2 Number and distribution of wordhood issues in Zingler (2020).

Dependence	*Definiteness*	*Case*	*Indexation*	*Tense*	*Total*
P>S	6	28	15	14	63
S>P	2	1	4	2	9

References

1 Bickel, Balthasar, Kristine Hildebrandt & René Schiering. 2009. The distribution of phonological word domains: A probabilistic typology. In Grijzenhout, Janet & Barış, Kabak (eds.), Phonological domains, 47-75. Berlin: De Gruyter.

2 Bybee, Joan. 1985. Morphology. Amsterdam: Benjamins.

3 Bybee, Joan. 2001. Phonology and language use. Cambridge: Cambridge University Press.

4 Bybee, Joan. 2015. Language change. Cambridge: Cambridge University Press.

5 Bybee, Joan, Revere Perkins & William Pagliuca. 1994. The evolution of grammar. Chicago: The University of Chicago Press.

6 Dixon, R. M. W. 2010. Basic linguistic theory, vol. 2. Oxford: Oxford University Press.

7 Dixon, R. M. W. & Alexandra Aikhenvald. 2003. Word: A typological framework. In Dixon & Aikhenvald (eds.), Word, 1-41. Cambridge: Cambridge University Press.

8 Evans, Nicholas. 1995. A grammar of Kayardild. Berlin: De Gruyter.

9 Haspelmath, Martin. 2011. The indeterminacy of word segmentation and the nature of morphology and syntax. Folia Linguistica 45, 31-80.

10 Haspelmath, Martin & Andrea Sims. 2010. Understanding morphology. 2nd ed. London: Hodder.

11 Hopper, Paul & Elizabeth Traugott. 2003. Grammaticalization. 2nd ed. Cambridge: Cambridge University Press.

12 Mithun, Marianne. 2014. Morphology: What's in a word? In Genetti, Carol (ed.), How languages work, 71-99. Cambridge: Cambridge University Press.

13 Schiering, René, Balthasar Bickel & Kristine Hildebrandt. 2010. The prosodic word is not universal but emergent. Journal of Linguistics 46, 657-709.

14 Tallman, Adam. 2020. Beyond grammatical and phonological words. Language and Linguistics Compass 14(2), e12364, 1-14.

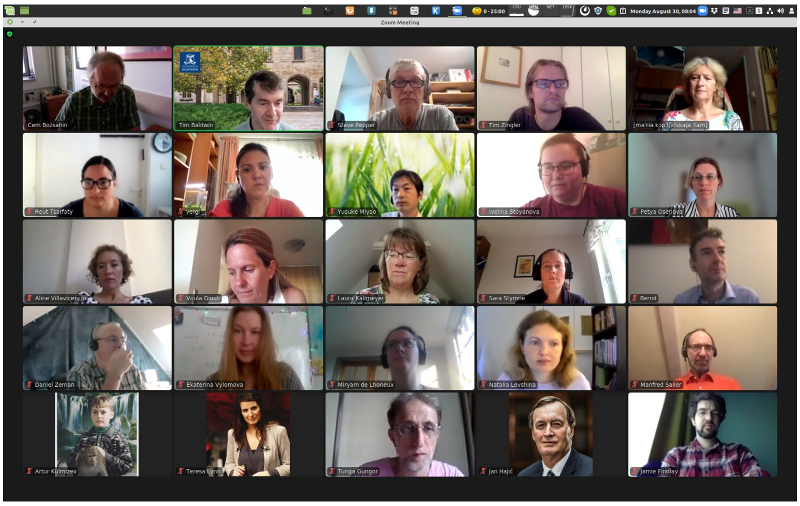
15 Tallman, Adam. 2021. Constituency and coincidence in Chácobo (Pano). Studies in Language 45, 321-383.

16 Zingler, Tim. 2020. Wordhood issues: Typology and grammaticalization. PhD dissertation, University of New Mexico.

Remote Participants

- Timothy Baldwin
The University of Melbourne, AU
- Verginica Barbu Mititelu
Research Institute for A.I. – Bucharest, RO
- Emily M. Bender
University of Washington – Seattle, US
- Archna Bhatia
Florida IHMC – Ocala, US
- Bernd Bohnet
Google – Amsterdam, NL
- Francis Bond
Nanyang TU – Singapore, SG
- Cem Bozsahin
Middle East Technical University – Ankara, TR
- Ryan Cotterell
ETH Zürich, CH
- William Croft
University of New Mexico – Alburquerque, US
- Miryam de Lhoneux
University of Copenhagen, DK
- Marie-Catherine de Marneffe
Ohio State University – Columbus, US
- Jamie Findlay
University of Oslo, NO
- Daniel Flickinger
Stanford University, US
- Kim Gerdes
University Paris-Saclay – Orsay, FR
- Voula Giouli
Athena Research Center, GR
- Tunga Gungor
Bogaziçi University – Istanbul, TR
- Jan Hajic
Charles University – Prague, CZ
- Dag Haug
University of Oslo, NO
- Uxoa Iñurrieta
Donostia, ES
- Laura Kallmeyer
Universität Düsseldorf, DE
- Christo Kirov
Google – New York, US
- Maria Koptjevskaja Tamm
Stockholm University, SE
- Artur Kulmizev
Uppsala University, SE
- Lori Levin
Carnegie Mellon University – Pittsburgh, US
- Natalia Levshina
Max-Planck-Institute for Psycholinguistics – Nijmegen, NL
- Teresa Lynn
Dublin City University, IE
- Stella Markantonatou
Athena Research Center, GR
- Nurit Melnik
The Open University of Israel – Raanana, IL
- Paola Merlo
University of Geneva, CH
- Yusuke Miyao
University of Tokyo, JP
- Kadri Muischnek
University of Tartu, EE
- Joakim Nivre
Uppsala University, SE
- Petya Osenova
Bulgarian Academy of Sciences – Sofia, BG
- Stephen Pepper
University of Oslo, NO
- James Pustejovsky
Brandeis University – Waltham, US
- Alexandre Rademaker
IBM Research – Sao Paulo, BR
- Carlos Ramisch
Aix-Marseille University, FR
- Manfred Sailer
Goethe-Universität Frankfurt am Main, DE
- Agata Savary
Université de Tours – Blois, FR
- Emmanuel Schang
University of Orleans, FR
- Nathan Schneider
Georgetown University – Washington, DC, US
- Ivelina Stoyanova
Bulgarian Academy of Sciences – Sofia, BG
- Sara Stymne
Uppsala University, SE
- Reut Tsarfaty
Bar-Ilan University – Ramat Gan, IL
- Francis M. Tyers
Indiana University – Bloomington, US
- Meagan Vigus
University of New Mexico – Alburquerque, US
- Aline Villavicencio
University of Sheffield, GB
- Veronika Vincze
University of Szeged, HU
- Ekaterina Vylomova
The University of Melbourne, AU
- Nianwen Xue
Brandeis University – Waltham, US
- David Yarowsky
Johns Hopkins University – Baltimore, US
- Amir Zeldes
Georgetown University – Washington, DC, US
- Daniel Zeman
Charles University – Prague, CZ
- Tim Zingler
University of New Mexico – Alburquerque, US

Joakim Nivre


Monday August 30, 08:04
Zoom Meeting
Cem Bozsahin
Tim Baldwin
Steve Pepper
Tim Zingler
[ma'ria kɔp'tʃɛfskaja 'tam]
Reut Tsarfaty
vergi
Yusuke Miyao
Ivelina Stoyanova
Petya Osenova
Aline Villavicencio
Voula Giouli
Laura Kallmeyer
Sara Stymne
Bernd
Daniel Zeman
Ekaterina Vylomova
Miryam de Lhoneux
Natalia Levshina
Manfred Sailer
Artur Kulmizev
Teresa Lynn
Tunga Gungor
Jan Hajič
Jamie Findlay

Zoom Meeting
ceramisch
Stella Markantonatou
Joakim Nivre
Agata Savary
Cem Bozsahin
Tim Baldwin
Steve Pepper
Tim Zingler
[ma'ria kɔp'tʃɛfskaja 'tam]
Reut Tsarfaty
vergi
Yusuke Miyao
Ivelina Stoyanova
Petya Osenova
Aline Villavicencio
Laura Kallmeyer
Sara Stymne
Bernd
Daniel Zeman
Ekaterina Vylomova
Miryam de Lhoneux
Natalia Levshina
Manfred Sailer
Artur Kulmizev
Teresa Lynn

Report from Dagstuhl Seminar 21352

Higher-Order Graph Models: From Theoretical Foundations to Machine Learning

Edited by

Tina Eliassi-Rad[1], Vito Latora[2], Martin Rosvall[3], and Ingo Scholtes [*4]

1 **Northeastern University – Boston, US,** `tina@eliassi.org`
2 **Queen Mary University of London, GB,** `v.latora@qmul.ac.uk`
3 **University of Umeå, SE,** `martin.rosvall@umu.se`
4 **Julius-Maximilians-Universität Würzburg, DE & Universität Zürich, CH,** `ingo.scholtes@uni-wuerzburg.de`

Abstract

Graph and network models are essential for data science applications in computer science, social sciences, and life sciences. They help to detect patterns in data on dyadic relations between pairs of genes, humans, or documents, and have improved our understanding of complex networks across disciplines. While the advantages of graph models of relational data are undisputed, we often have access to data with multiple types of higher-order relations not captured by simple graphs. Such data arise in social systems with non-dyadic or group-based interactions, multi-modal transportation networks with multiple connection types, or time series containing specific sequences of nodes traversed on paths. The complex relational structure of such data questions the validity of graph-based data mining and modelling, and jeopardises interdisciplinary applications of network analysis and machine learning.

To address this challenge, researchers in topological data analysis, network science, machine learning, and physics recently started to generalise network analysis to higher-order graph models that capture more than dyadic relations. These higher-order models differ from standard network analysis in assumptions, applications, and mathematical formalisms. As a result, the emerging field lacks a shared terminology, common challenges, benchmark data and metrics to facilitate fair comparisons. By bringing together researchers from different disciplines, Dagstuhl Seminar 21352 "Higher-Order Graph Models: From Theoretical Foundations to Machine Learning" aimed at the development of a common language and a shared understanding of key challenges in the field that foster progress in data analytics and machine learning for data with complex relational structure. This report documents the program and the outcomes of this seminar.

Seminar August 29–1, 2021 – http://www.dagstuhl.de/21352

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Machine learning theory; Mathematics of computing → Graph theory; Human-centered computing → Graph drawings; Human-centered computing → Social network analysis

Keywords and phrases (Social) Network analysis, Graph mining, Graph theory, Network science, Machine Learning, Statistical relational learning, Topological data analysis

Digital Object Identifier 10.4230/DagRep.11.7.139

Edited in cooperation with Unai Alvarez-Rodriguez, Leonie Neuhäuser, Vincenzo Perri, and Chester Tan

[*] Ingo Scholtes acknowledges financial support from the Swiss National Science Foundation through grant no. 176938.

Higher-Order Graph Models: From Theoretical Foundations to Machine Learning, *Dagstuhl Reports*, Vol. 11, Issue 07, pp. 139–178
Editors: Tina Eliassi-Rad, Vito Latora, Martin Rosvall, and Ingo Scholtes
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Ingo Scholtes (Julius-Maximilians-Universität Würzburg, DE & Universität Zürich, CH)
Tina Eliassi-Rad (Northeastern University – Boston, US)
Vito Latora (Queen Mary University of London, GB)
Martin Rosvall (University of Umeå, SE)

The network science and graph mining community has created a rich portfolio of data analysis and visualisation techniques that have become a cornerstone for knowledge extraction from relational data on complex systems. Most of those techniques build on simple graph abstractions, where nodes represent a system's elements, and links represent *dyadic interactions, relations, or dependencies* between those elements. This mathematical formalism has proven useful for reasoning, e.g., about the centrality of nodes, the evolution and control of dynamical processes, and the community or cluster structure in complex systems, given that we have access to *relational data* [17]. However, the graph abstractions used in those methods typically do not account for **higher-order relations** between nodes that are present in many real complex systems. Important examples for such data include:

- relational data that is inherently non-dyadic, such as (unordered) sets of authors co-authoring scientific articles, protein triplets in a cell that simultaneously interact with each other, or actors in social systems engaging in group collaborations,
- time-stamped data on social networks with chronologically ordered sequences of (dyadic) interactions, where specific sequences of nodes interact via *causal paths*
- sequential data on networked systems, such as user click streams, mobility trajectories, financial transaction sequences, citation paths, or directed acyclic graphs that give rise to a chronologically or topologically ordered sequences of nodes traversed by processes
- data on networked systems with multiple types or layers of links that cannot be reduced to a simple graph model

Over the past years, researchers have shown that the presence of such higher-order interactions can fundamentally alter our understanding of complex systems. They can change our notion of the importance of nodes captured by centrality measures, affect the detection of cluster and community structures in graphs, and influence dynamical processes like diffusion or epidemic spreading, as well as associated control strategies in non-trivial ways [24, 28, 33, 4, 5, 10, 35]. To further develop graph-based representations of data and broaden their potential application in pattern recognition, data analysis, and machine learning, over the past few years researchers have developed a rich portfolio of **higher-order network models and representations** that capture more than just dyadic dependencies in complex systems. The organisers of this seminar have recently summarised current research and open challenges in this area in three independent overview and perspective articles [1, 30, 15]. An incomplete list of approaches explored over the past few years include:

- hypergraphs, where each *hyperedge* can connect an arbitrary number of nodes [11]
- simplicial network models, where *simplices* represent d-dimensional group interactions [12, 10]
- d-dimensional De Bruijn graphs, where edges capture *ordered* sets of d dyadic interactions [28, 16]
- memory networks, where memory nodes capture Non-Markovian properties in time series data [24]

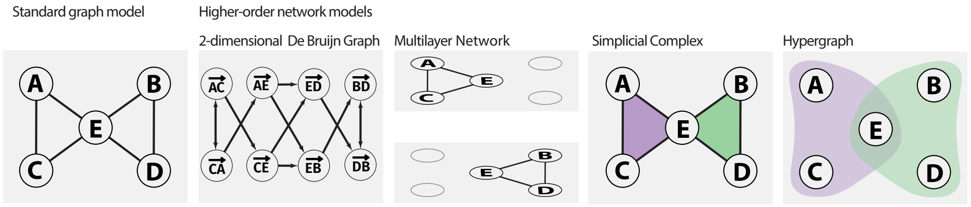


Figure 1 Illustration of standard graph model (left) and four modelling approaches capturing different types of higher-order interactions proposed in topological data analysis, network science, and computer science. Figure adapted from [15].

- higher-, variable-, and multi-order Markov models for temporal networks [33, 20, 4]
- multi-layer and multiplex networks with multiple types of links between nodes [13]
- applications of categorical sequence mining techniques to model patterns in sequences of node sets [7]

In Figure 1, we illustrate some of the higher-order graph models listed above. All these modelling approaches address the same fundamental limitation of graph models when studying complex systems: **we cannot understand a system's structure and dynamics by decomposing direct and indirect interactions between elements into a set of dyadic relations with a single type**. However, the similarities and differences between these different approaches are still not fully understood.

At a critical time for the community, this Dagstuhl seminar intended to improve our understanding of the strengths, weaknesses, commonalities, and differences of these different approaches along with their resulting computational and epistemological challenges. The seminar aimed to create a common foundation for developing graph mining and machine learning techniques that use recent advances in the study of higher-order graph models by gathering key researchers from different communities, including machine learning, information retrieval and data mining, complex systems theory, theoretical physics, network science, computational social science, and mathematics. The participants included senior and junior researchers focusing on four related and intersecting topics: (i) Topological and Graph-Theoretic Foundations, (ii) Higher-Order Models for Dynamical Processes, (iii) Higher-Order Pattern Recognition and Machine Learning, (iv) Computational Aspects in Higher-Order Graph Analysis and Graph Mining.

The organisers used the four topics to structure the seminar program and derive the participants' initial assignment to possible working groups. After an initial round of brief opening statements, participants introduced themselves and stated their specific interests for the seminar during five-minute lightning talks. During a match-making session taking place in the afternoon of day one, all interests expressed by the participants were consolidated into a set of working groups, addressing the following six areas: (i) Visualisation and Interpretability of Higher-Order Graph Models, (ii) Learning and Model Selection, (iii) Unification of Different Higher-Order Modelling Frameworks, (iv) Benchmark Data and Evaluation Practices, (v) Applications of Higher-Order Graph Models, and (vi) Societal Impact, Robustness, and Fairness. In the remaining time of the seminar, participants worked on those issues in the groups. This report includes summaries of the opening statements, the results of the working groups, and a summary of a panel discussion taking place on the evening of day two.

References

1 Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 2020.

2 Austin R. Benson. Tools for higher-order network analysis. *CoRR*, abs/1802.06820, 2018.

3 Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 2018.

4 Austin R. Benson, David F. Gleich, and Jure Leskovec. Tensor spectral clustering for partitioning higher-order network structures. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 118–126, 2015.

5 Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

6 Austin R. Benson, David F. Gleich, and Lek-Heng Lim. The spacey random walk: A stochastic process for higher-order data. *SIAM Review*, 59(2):321–345, 2017.

7 Austin R Benson, Ravi Kumar, and Andrew Tomkins. Sequences of sets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1148–1157, 2018.

8 Matthias Bolten, Stephanie Friedhoff, Andreas Frommer, Matthias Heming, and Karsten Kahl. Algebraic multigrid methods for laplacians of graphs. *Linear Algebra and its Applications*, 434(11):2225 – 2243, 2011. Special Issue: Devoted to the 2nd NASC 08 Conference in Nanjing (NSC).

9 Daniel Edler, Ludvig Bohlin, et al. Mapping higher-order network flows in memory and multilayer networks with infomap. *Algorithms*, 10(4):112, 2017.

10 Ernesto Estrada and Grant J. Ross. Centralities in simplicial complexes. applications to protein interaction networks. *Journal of Theoretical Biology*, 438:46 – 60, 2018.

11 Gourab Ghoshal, Vinko Zlatić, Guido Caldarelli, and Mark EJ Newman. Random hypergraphs and their applications. *Physical Review E*, 79(6):066118, 2009.

12 Iacopo Iacopini, Giovanni Petri, Alain Barrat, and Vito Latora. Simplicial models of social contagion. *Nature communications*, 10(1):1–9, 2019.

13 Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 07 2014.

14 Renaud Lambiotte, Martin Rosvall, Michael Schaub, Ingo Scholtes, and Jian Xu. Beyond graph mining: Higher-order data analytics for temporal network data. In *Hands-On Tutorial at the 24th ACM SIGKDD International Conference on Knowledge Discovery*, volume 38, 2018.

15 Renaud Lambiotte, Martin Rosvall, and Ingo Scholtes. From networks to optimal higher-order models of complex systems. In *Nature physics*, Vol. 15, p. 313-320, March 25 2019

16 Timothy LaRock, Vahan Nanumyan, Ingo Scholtes, Giona Casiraghi, Tina Eliassi-Rad, and Frank Schweitzer. HYPA: Efficient detection of path anomalies in time series data on networks. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 460–468. SIAM, 2020.

17 Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex Networks: Principles, Methods and Applications.* Cambridge University Press, 2017.

18 Naoki Masuda, Mason A. Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics Reports*, 716-717:1 – 58, 2017. Random walks and diffusion on networks.

19 Huda Nassar, Caitlin Kennedy, Shweta Jain, Austin R. Benson, and David F. Gleich. Using cliques with higher-order spectral embeddings improves graph visualizations. In *Proceedings of the 2020 Web Conference (WWW)*, pages 2927–2933, 2020.

20 Tiago P Peixoto and Martin Rosvall. Modelling sequences and temporal networks with dynamic community structures. *Nature communications*, 8(1):1–12, 2017.

21 Vincenzo Perri and Ingo Scholtes. Hotvis: Higher-order time-aware visualisation of dynamic graphs. In *Proceedings of the 28th International Symposium on Graph Drawing and Network Visualization (Graph Drawing 2020), Vancouver, BC, Canada (to appear)*, 2020.

22 Luka V. Petrovic and Ingo Scholtes. PaCo: Fast Counting of Causal Paths in Temporal Network Data. In *Proceedings of the 11th Temporal Web Analytics Workshop (TempWeb 2021)* held in conjunction with The Web Conference 2021, Ljubljana, Slovenia, April 2021

23 Luka V. Petrovic and Ingo Scholtes. Learning the markov order of paths in a network, 2020. arXiv 2007.02861.

24 Martin Rosvall, Alcides V Esquivel, Andrea Lancichinetti, Jevin D West, and Renaud Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5:4630, 2014.

25 Mandana Saebi, Giovanni Luca Ciampaglia, Lance M Kaplan, and Nitesh V Chawla. Honem: Network embedding using higher-order patterns in sequential data. *arXiv preprint arXiv:1908.05387*, 2019.

26 Mandana Saebi, Jian Xu, Lance M. Kaplan, Bruno Ribeiro, and Nitesh V. Chawla. Efficient modeling of higher-order dependencies in networks: from algorithm to application for anomaly detection. *EPJ Data Sci.*, 9(1):15, 2020.

27 Ingo Scholtes. When is a network a network? multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1037–1046, New York, NY, USA, 2017. Association for Computing Machinery.

28 Ingo Scholtes, Nicolas Wider, René Pfitzner, Antonios Garas, Claudio Tessone, and Frank Schweitzer. Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks. *Nature communications*, 5:5024, 2014.

29 Yingxia Shao, Shiyue Huang, Xupeng Miao, Bin Cui, and Lei Chen. Memory-aware framework for efficient second-order random walk on large graphs. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, page 1797–1812, 2020.

30 Leo Torres, Ann S Blevins, Danielle S Bassett, and Tina Eliassi-Rad. The why, how, and when of representations for complex systems. *arXiv preprint arXiv:2006.02870*, 2020.

31 Francesco Tudisco, Austin R. Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. *CoRR*, abs/2006.04762, 2020.

32 Tao Wu, Austin R. Benson, and David F. Gleich. General tensor spectral co-clustering for higher-order data. In *Advances in Neural Information Processing Systems 29*, pages 2559–2567, 2016.

33 Jian Xu, Thanuka L Wickramarathne, and Nitesh V Chawla. Representing higher-order dependencies in networks. *Science advances*, 2(5):e1600028, 2016.

34 Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 555–564, 2017.

35 Yan Zhang, Antonios Garas, and Ingo Scholtes. Higher-order models capture changes in controllability of temporal networks. In Journal of Physics: Complexity, Vol. 2, No. 1, January 29, 2021

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

Panel discussions

3 Overview of Talks

3.1 Inference of Time-ordered Multi-Body Interactions

Unai Alvarez-Rodriguez (Universität Zürich, CH)

Higher-order models are typically specialised in a single class of interaction. Multi-time, multi-system and multi-type modelling approaches have not yet been combined, and therefore there is no framework capable of describing processes that simultaneously manifest different classes of interactions. I argue a unification of higher-order models is necessary to bypass this limitation and to improve our understanding of complex systems. Along these lines, I present preliminary results for extracting time-ordered multi-body interactions from time series of systems composed by multiple interacting elements.

3.2 Cascade Processes in Machine Learning

Rebekka Burkholz (Harvard School of Public Health – Boston, US)

Joint work of Alkis Gotovos, Rebekka Burkholz, John Quackenbush, Stefanie Jegelka
Main reference Alkis Gotovos, Rebekka Burkholz, John Quackenbush, Stefanie Jegelka: "Scaling up Continuous-Time Markov Chains Helps Resolve Underspecification", CoRR, Vol. abs/2107.02911, 2021.
URL https://arxiv.org/abs/2107.02911

I have proposed to develop a unifying framework to represent higher order network information by parametrising a process that evolves on a network as graph neural network. This could be combined with the design of suitable covariate information that represents the higher order model information and would enable the inference of networks and processes based on data. Yet, in many situations, this unification approach is expected to suffer from overparametrisation leaving the question whether there are better and more parameter efficient representations of higher order structure for a given task. As motivating problem, I have presented recent work about learning the order in which mutations accumulate during cancer progression.

3.3 The Why, How, and When of Representations for Complex Systems

Tina Eliassi-Rad (Northeastern University – Boston, US)

Joint work of Leo Torres, Ann S. Blevins, and Danielle Bassett
Main reference Leo Torres, Ann Sizemore Blevins, Danielle S. Bassett, Tina Eliassi-Rad. The Why, How, and When of Representations for Complex Systems. SIAM Review (SIREV), 63(3): 435-485, 2021.
URL https://epubs.siam.org/doi/pdf/10.1137/20M1355896

Complex systems, which at the most fundamental level consist of entities and their interactions, describe phenomena in a wide variety of fields, from neuroscience to computer science to economics. The wide variety of applications has led to two key challenges: (1) the development of many domain-specific strategies for analysing complex systems, and (2) the compartmentalization of representation and analysis within a domain due to inconsistencies in the language for complex systems. In our work, we propose a domain-agnostic language to develop a more coherent vocabulary. We use this language to evaluate each step of the analysis of complex systems. We start with the system under study and its observations in terms of the collected data, and then go through different mathematical frameworks for encoding the observed data (i.e., graphs, simplicial complexes, and hypergraphs) and relevant computational methods for each framework. At each step, we consider different types of dependencies. These are properties of the system that describe how the existence of an interaction between a group of entities in a system can affect the possibility of the existence of another relationship. We discuss how dependencies can arise and how they can change the interpretation of results or the entire analysis pipeline. We conclude with two real-world examples.

3.4 Spreading and Centrality on Hypergraphs

Desmond J. Higham (University of Edinburgh, GB)

Joint work of Desmond J. Higham, Henry-Louis de Kergorlay, Francesco Tudisco
Main reference Desmond J. Higham and Henry-Louis de Kergorlay, Epidemics on hypergraphs: Spectral thresholds for extinction, Proceedings of the Royal Society, Series A, 2021
URL https://doi.org/10.1098/rspa.2021.0232

We typically interact in groups, not just in pairs. The use of hyperedges naturally allows us to model with a nonlinear rate of transmission, in terms of both the group size and the number of infected group members, as is the case, for example, when social distancing is encouraged. I am therefore interested in individual-level, stochastic disease models on a hypergraph [1, 2]. I am also interested in centrality measures that take account of group interactions, which leads to nonlinear eigenvalue problems, and nonlinear extensions of Perron-Frobenius theory and the power method [3].

References

1 Desmond J. Higham and Henry-Louis de Kergorlay, Epidemics on hypergraphs: Spectral thresholds for extinction, Proceedings of the Royal Society, Series A, 2021

2 Desmond J. Higham and Henry-Louis de Kergorlay, Mean field analysis of hypergraph contagion models, arXiv: 2108.05451, 2021

3 F. Tudisco and Desmond J. Higham, Node and edge eigenvector centrality for hypergraphs, Communications Physics, 2021.

3.5 Interacting Discovery Processes on Complex Networks

Gabriele DiBona (Queen Mary University of London, GB)

Main reference Iacopo Iacopini, Gabriele Di Bona, Enrico Ubaldi, Vittorio Loreto, Vito Latora: "Interacting Discovery Processes on Complex Networks", Phys. Rev. Lett., Vol. 125, p. 248301, American Physical Society, 2020.
URL http://dx.doi.org/10.1103/PhysRevLett.125.248301

In my talk, I focused on the influence of social interactions on collective processes, such as the exploration and the discovery of new content in different contexts. The challenge is now to include group interactions using higher-order methods, with a data-driven approach. This can have implications in phenomena as diverse as user interaction in online social networks, collective decisions in teams, team success and optimal structures, nonlinear random walks, brain analysis in social activities, brain creativity, and diffusion of innovation. A first step in this direction has been done in our recent paper [1].

References

1 Iacopo Iacopini, Gabriele Di Bona, Enrico Ubaldi, Vittorio Loreto, and Vito Latora. *Interacting Discovery Processes on Complex Networks*. Phys. Rev. Lett. 125, 248301, 10 December 2020.

3.6 Dynamical Processes on Higher-Order Networks: Beyond Dyadic Projections

Luca Gallo (University of Catania, IT)

Starting from the current literature about dynamical processes on higher-order networks, I formulate two theoretical questions. (i) Dynamical processes on hypergraphs and on simplicial complexes are usually studied in parallel [3, 2]. Can we produce a general theory of dynamical systems on higher-order networks? In particular, is it possible to point out if and how the absence or the presence of the inclusion requirement impacts the dynamics? (ii) To make the problem analytically feasible, previous efforts in the study of dynamical processes on higher-order structures have relied on the definition of suitable dyadic projections [1, 2, 3], i.e. equivalent weighted networks. However, this method can lose information about the higher-order structure, possibly preventing a complete study of the dynamics [4]. Can we produce an analytical framework that goes beyond projected networks?

References

1 Timoteo Carletti, Duccio Fanelli, and Sara Nicoletti. *Dynamical systems on hypergraphs*. Journal of Physics: Complexity 1(3), 035006 (2020)

2 Guilherme Ferraz de Arruda, Michele Tizzani, and Yamir Moreno. *Phase transitions and stability of dynamical processes on hypergraphs*. Communications Physics 4(1), 1-9, (2021)

3 Lucia Valentina Gambuzza, Francesca Di Patti, Luca Gallo, Stefano Lepri, Miguel Romance, Regino Criado, Mattia Frasca, Vito Latora and Stefano Boccaletti. *Stability of synchronization in simplicial complexes*. Nature communications, 12(1), 1-13, (2021).

4 Anastasiya Salova and Raissa M. D'Souza, *Analyzing states beyond full synchronization on hypergraphs requires methods beyond projected networks*. arXiv preprint arXiv:2107.13712. (2021)

3.7 New Data for Higher-Order Network Research

David F. Gleich (Purdue University – West Lafayette, US)

Joint work of Austin R. Benson, Nate Veldt, David F. Gleich
Main reference Austin R. Benson, Nate Veldt, David F. Gleich: "fauci-email: a json digest of Anthony Fauci's released emails", CoRR, Vol. abs/2108.01239, 2021.
URL https://arxiv.org/abs/2108.01239

We discussed some challenges we had in visualising a new higher-order dataset derived as a hypergraph representation of Anthony Fauci's emails regarding the COVID-19 pandemic [2]. Analytic studies of these data show how higher-order features were more stable than their graph counterparts [1]; but in the abstract presentation we highlighted how the lack of hypergraph visualisation tools limited our investigation of the data. This dataset is a more modern counterpart to the famous Karate Club dataset as well as the Enron email dataset. In our working paper, we provide a fully parsed version suitable to derive a number of graph, hypergraph, and other higher-order datasets.

References

1 Austin R. Benson, Nate Veldt, and David F. Gleich. 2021. fauci-email: a json digest of Anthony Fauci's released emails. *arXiv* cs.SI 2108.01239. Published online at `http://arxiv.org/abs/2108.01239` and data available from `https://github.com/nveldt/fauci-email`.

2 Natalie Bettendorf and Jason Leopold. Anthony Fauci's Emails Reveal The Pressure That Fell On One Man. *BuzzFeed News*. June 2021. Published online at `https://www.buzzfeednews.com/article/nataliebettendorf/fauci-emails-covid-response`.

3.8 Combining Higher-Order Graph Models with Expert Knowledge

Christoph Gote (ETH Zürich, CH)

Joint work of Christoph Gote, Giona Casiraghi, Frank Schweitzer, Ingo Scholtes
Main reference Christoph Gote, Giona Casiraghi, Frank Schweitzer, Ingo Scholtes: "Predicting Sequences of Traversed Nodes in Graphs using Network Models with Multiple Higher Orders", CoRR, Vol. abs/2007.06662, 2020.
URL https://arxiv.org/abs/2007.06662

Higher-order correlations facilitate unprecedented insights into system processes. However, to interpret and validate the results, we need both a thorough theoretical understanding of the underlying methods and expert subject matter knowledge. We conjecture that the overlap between groups with expertise regarding both aspects is low. Consequently, we ask how we can increase the visibility and applicability of higher-order methods in other scientific fields.

3.9 Benchmarking and Robustness of Higher-Order Graph Models

Stephan Günnemann (Technical University of Munich, DE)

Machine learning methods taking higher-order network structure into account have the potential to obtain richer and potentially more accurate results by modeling the underlying complex graph data better. To evaluate the real success of such higher-order graph-based ML models, however, fair evaluation and benchmarking principles are required – a non-trivial task even for standard graphs and graph learning models [1]. Indeed, beyond providing suitable benchmark datasets of higher-order graph models, such evaluation practices have to identify common tasks and appropriate baselines specifically tacking the higher-order nature into account and comparing them to standard graph approaches. Moreover, evaluations should not be limited to metrics such as accuracy but specifically the robustness of the models need be considered. While, e.g., standard graph neural networks have been shown to be non-robust [2], it is an open challenge whether higher-order graph structures can make the methods and analysis more reliable and, thus, leading to more robust ML models.

References

1 Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. *Relational Representation Learning Workshop, Neural Information Processing Systems*, 2018.

2 Stephan Günnemann. Graph Neural Networks: Adversarial Robustness. *Graph Neural Networks: Foundations, Frontiers, and Applications*, Springer, chapter 8, 2021.

3.10 (Knowledge | Hyper) Graphs in Social Media and Text

Andreas Hotho (Julius-Maximilians-Universität Würzburg, DE)

In my talk, I focused on three topics related to the seminar. First, hypergraphs are the underlying structure of FolkSonomies, which are behind tagging systems emerged in the Web 2.0 wave. At that time, we started our system BibSonomy which is still online[1, 7]. All the user generated data are freely available for research. Further, I pointed to a couple of results on hypergraphs, e.g. our analysis of the graph structure [2], the behaviour analysis together with Markus Strohmaier [5] and the emergent semantics in the systems [6]. We also developed new ranking and recommendation algorithm. The second topic is on the edge of graphs and natural languages processing (NLP). I show two showcases, the analysis of plots on German novels and dime novels and the emergent languages and communication patterns in the chat messages of the twitch.tv platform [3]. Third, knowledge graphs well known in the semantic web community and widely adopted in many other areas are another graph structure of interest. By integrating KGs with languages models like BERT or GPT, the graph structure is becoming even more interesting for the higher order graph community [4].

References

1 Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme. The social bookmark and publication management system bibsonomy. *The VLDB Journal*, 19(6):849–875, December 2010.

2 Ciro Cattuto, Christoph Schmitz, Andrea Baldassarri, Vito D. P. Servedio, Vittorio Loreto, Andreas Hotho, Miranda Grahl, and Gerd Stumme. Network properties of folksonomies. *AI Communications*, 20(4):245 – 262, 2007.

3 Konstantin Kobs, Albin Zehe, Armin Bernstetter, Julian Chibane, Jan Pfister, Julian Tritscher, and Andreas Hotho. Emote-controlled: Obtaining implicit viewer feedback through emote based sentiment analysis on comments of popular twitch.tv channels. *ACM Transactions on Social Computing*, 3(2):1–34, May 2020.

4 Janna Omeliyanenko, Albin Zehe, Lena Hettinger, and Andreas Hotho. Lm4kg: Improving common sense knowledge graphs with language models. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 456–473, Cham, 2020. Springer International Publishing.

5 P. Singer, D. Helic, A. Hotho, and M. Strohmaier. Hyptrails: A bayesian approach for comparing hypotheses about human trails. In *24th International World Wide Web Conference (WWW2015)*, Firenze, Italy, May 18 – May 22 2015. ACM, ACM.

6 Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *The Semantic Web – ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 615–631. Springer Berlin / Heidelberg, 2008.

7 Daniel Zoller, Stephan Doerfel, Robert Jäschke, Gerd Stumme, and Andreas Hotho. Posted, visited, exported: Altmetrics in the social tagging system bibsonomy. *Journal of Informetrics*, 10(3):732 – 749, 2016.

3.11 Inequalities and Higher-Order Interactions

Fariba Karimi (Complexity Science Hub – Wien, AT)

In my talk I propose to consider the types of inequalities that are hidden in higher order interactions that we would miss if we don't consider them. For example, the presence of multiple groups of various size and mixing patterns between groups may cause certain types of hypergraphs representations and result in specific group dynamics. I am interested in developing network models that would consider higher order interactions and use that to understand the emergence of inequalities in society and algorithms.

3.12 Higher-Order Processes in Complex Systems

Vito Latora (Queen Mary University of London, GB)

Dynamical processes on systems with higher-order interactions and/or systems with higher-order temporal dependencies can help to understand the neural and social components of creativity. In this talk I will show some examples of models of collective exploration [1], of social interactions [2] and social contagion [3] that can be generalised to take into account higher-order interactions and higher-order temporal dependencies [4]. I will also point to some examples of possible experiments to test the effects of higher-order interactions on the dynamics of social systems.

References

1 Iacopini, Di Bona, Ubaldi, Loreto and Latora, 2020, Interacting Discovery Processes on Complex Networks. Phys. Rev. Lett. 125, 248301.

2 Evolutionary dynamics of higher-order interactions in social networks (2021) Alvarez-Rodriguez, Battiston, Ferraz de Arruda, Moreno, Perc, Latora, Nat. Hum. Behav. 5, 586.

3 Iacopini, Petri, Barrat and Latora, 2019. Simplicial models of social contagion. Nat. Comm. 10(1), pp.1-9.

4 Mazzarisi, Lillo, Williams and Latora, Non-Markovian temporal networks with auto- and cross-correlated link dynamics, arXiv preprint arXiv:1909.08134.

3.13 Simplicial Network Analysis Based on Electrical Networks

Kang-Ju Lee (Seoul National University, KR)

Joint work of Woong Kook
Main reference Woong Kook, Kang-Ju Lee: "Simplicial networks and effective resistance", Adv. Appl. Math., Vol. 100, pp. 71–86, 2018.
URL http://dx.doi.org/10.1016/j.aam.2018.05.004

I introduce network invariants based on simplicial electrical networks. Effective resistance also known as resistance distance measures how well currents generated by an edge between two vertices as a battery are resisted. Under d-dimensional Kirchhoff's current and voltage laws, we introduce simplicial effective resistance among d+1 vertices [1]. We make use of our measure to propose a simplicial analogue of current-flow closeness centrality or information centrality. We define the simplicial Kirchhoff index as a robustness measure for simplicial networks [2]. We also propose a high-dimensional generalisation of the concept of the number of connected components.

One of the advantages of using simplicial complexes is that we can utilise tools from algebraic topology. Generalising studies in network theory for 1-cycles or flows to simplicial networks will take advantage of it. Finding data set concerning high-dimensional cycles or flows will support these studies.

References

1 Woong Kook and Kang-Ju Lee. Simplicial networks and effective resistance, *Advances in Applied Mathematics* 100 (2018) 71-86.

2 Woong Kook and Kang-Ju Lee. Kirchhoff index of simplicial networks, *Linear Algebra and its Applications* 626 (2021) 1-19.

3.14 Machine Learning for Networks

Lisi Qarkaxhija (Koper, SI)

I am a recent Master's graduate in the field of Data Science were my main priority was machine learning on networks. Before that, I completed a bachelor's degree in Mathematics. In my lightning talk, I introduced myself as a soon-to-be doctorate researcher in the field of Machine Learning for Complex Networks. As such, I established my interest in research concerning higher-order graphs and took the opportunity to familiarise myself with the topic and to form new connections.

3.15 Dynamical Processes on Higher-Order Models: Future Research

Leonie Neuhäuser (RWTH Aachen, DE)

Joint work of Leonie Neuhäuser, Michael Thomas Schaub, Renaud Lambiotte, Andrew Mellor
Main reference Leonie Neuhäuser, Andrew Mellor, Renaud Lambiotte: "Multibody interactions and nonlinear consensus dynamics on networked systems", Phys. Rev. E, Vol. 101, p. 032310, American Physical Society, 2020.
URL http://dx.doi.org/10.1103/PhysRevE.101.032310

In my talk, I outline two ways of extending higher-order model research, motivated by my previous work on the interplay of dynamics and multi-body topology [1, 2]. Firstly, we have to consider the practicability of higher-order models. The overall system is often determined by an interplay of many model aspects (topology, temporal ordering, type of dynamics) and we need to detect which of these interactions aspects are qualitatively impacting the specific research question of interest. For this, it is important to consider both domain expert knowledge and model expert knowledge. Another interesting question is the interplay of different higher-order dimensions. Current methods are mainly focusing on one specific higher-order aspect, but different aspects may interact. We have investigated the interplay of temporal and multi-way interactions in [3] and found effects, that differ from their projections. This call for more research on the combination of different higher-order model facets.

References

1 Leonie Neuhäuser, Andrew Mellor, and Renaud Lambiotte. Multibody interactions and nonlinear consensus dynamics on networked systems. Physical Review E 101, 2020.

2 Leonie Neuhäuser, Renaud Lambiotte and Michael T. Schaub. Consensus Dynamics and Opinion Formation on Hypergraphs. arXiv:2105.01369, 2021.

3 Leonie Neuhäuser, Michael T. Schaub and Renaud Lambiotte. Consensus Dynamics on Temporal Hypergraphs. arXiv:2109.04985, 2021.

3.16 How, when, and which Higher-Order Models can we use?

Vincenzo Perri (Universität Zürich, CH)

The use of machine learning tools provides a fruitful way for the analysis of network systems. Unlike standard network models, the application of these tools to higher-order models is neither unified nor straightforward. This difficulty comes from the existence of multiple ways to extend these tasks on higher-order networks and a lack of understanding of the commonalities between the different types of higher-order models. In light of this, I am interested in examining the commonalities between higher-order methods and their possibilities for applications.

3.17 The Role of Higher-Order Interactions in Complex Systems

Giovanni Petri (ISI Foundation – Torino, IT), Federico Battiston (Central European University – Vienna, AT), Ginestra Bianconi (Queen Mary University of London, GB), Vito Latora (Queen Mary University of London, GB), and Yamir Moreno (University of Zaragoza, ES)

Main reference Federico Battiston, Enrico Amico, Alain Barrat, et al: “The physics of higher-order interactions in complex systems”. Nat. Phys. 17, 1093–1098 (2021)
URL https://www.nature.com/articles/s41567-021-01371-4

Complex networks have become the main paradigm for modelling the dynamics of interacting systems. However, networks are intrinsically limited to describing pairwise interactions, whereas real-world systems are often characterised by higher-order interactions involving groups of three or more units. Higher-order (polyadic) structures are therefore a better tool to map the real organisation of many social, biological and man-made systems. Here I outline key challenges for the physics of higher-order systems.

See [1, 2, 3].

References

1 Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.G. and Petri, G., 2020. Networks beyond pairwise interactions: structure and dynamics. Physics Reports, 874, pp.1-92.

2 Petri, G., Expert, P., Turkheimer, F., Carhart-Harris, R., Nutt, D., Hellyer, P.J. and Vaccarino, F., 2014. Homological scaffolds of brain functional networks. Journal of The Royal Society Interface, 11(101), p.20140873.

3 Iacopini, I., Petri, G., Barrat, A. and Latora, V., 2019. Simplicial models of social contagion. Nature communications, 10(1), pp.1-9.

3.18 Data-efficient Model Selection of Higher-order Networks

Luka Petrovic (Universität Zürich, CH)

Main reference Luka V. Petrovic, Ingo Scholtes: "Learning the Markov order of paths in a network", CoRR, Vol. abs/2007.02861, 2020.
URL https://arxiv.org/abs/2007.02861

In my previous work I have focused on statistical inference of higher-order network models for paths. They generally have large parameter spaces, and therefore require large amounts of data for training. We leveraged the fact that many networked systems have topological constraints and devised a Bayesian method to improve data-efficiency of model selection for higher-order network models for paths [1]. We believe that this methodology can improve statistical inference for a broader class of higher-order network models.

References

1 Luka Petrovic, Ingo Scholtes. Learning the Markov order of paths in a network, arXiv:2007.02861, https://arxiv.org/abs/2007.02861

3.19 Efficient Variable-Order Markov Models of Network Flows

Martin Rosvall (University of Umeå, SE), Daniel Edler (University of Umeå, SE), Anton Eriksson (University of Umeå, SE), and Jelena Smiljanic (University of Umeå, SE)

Main reference Christian Persson, Ludvig Bohlin, Daniel Edler, Martin Rosvall: "Maps of sparse Markov chains efficiently reveal community structure in network flows with memory", CoRR, Vol. abs/1606.08328, 2016.
URL http://arxiv.org/abs/1606.08328

Researchers develop maps that reveal essential patterns in network flows to better understand the flows of ideas or information through social and biological systems. In practice, network flow models have implied memoryless first-order Markov chains. Recently, researchers have introduced higher-order Markov chain models with memory to capture patterns in multi-step pathways, including revealing actual, overlapping community structures. However, higher-order Markov chain models suffer from the curse of dimensionality: their vast parameter spaces require exponentially increasing data to avoid overfitting and therefore make mapping inefficient already for moderate-sized systems. Model selection based on Markov chain state lumping into variable-order Markov chains and cross-validation alleviates this problem but wastes plentiful data. We need more efficient methods for reliably describing higher-order network flows. Two central questions arise: Which algorithm best explores the space of variable-order Markov chain models? How do we incorporate Bayesian methods to select the model that best describes the higher-order network flows?

3.20 What are Higher-Order Models?

Michael Schaub (RWTH Aachen, DE)

Main reference Christian Bick, Elizabeth Gross, Heather A. Harrington, Michael T. Schaub: "What are higher-order networks?", CoRR, Vol. abs/2104.11329, 2021.
URL https://arxiv.org/abs/2104.11329

In this talk I outlined several different ways in which we may consider higher-order models emerging from considerations of modelling low dimensional geometric structure (modelling nonlinear spaces); higher-order models for modelling non-dyadic relational data (interactions between groups vs interactions between pairs of nodes); and higher-order models for complex data supported on (fixed) domains such as hypergraphs, complexes etc.

References

1 Schaub, M. T., Zhu, Y., Seby, J. B., Roddenberry, T. M., and Segarra, S. *Signal processing on higher-order networks: Livin' on the edge... and beyond.* Signal Processing, 187, 108149, 2021

2 Bick, C., Gross, E., Harrington, H. A., and Schaub, M. T. *What are higher-order networks?.* arXiv preprint arXiv:2104.11329, 2021

3.21 Higher-Order Models and Cultural Data Analytics

Maximilian Schich (Tallinn University, EE)

Main reference Maximilian Schich: "Cultural analysis situs", ART-Dok, Heidelberg, 2019
URL https://doi.org/10.11588/artdok.00006347

In my talk I first gave a brief intro to the research mission of the CUDAN ERA Chair for Cultural Data Analytics at Tallinn University in Estonia (cf. `https://cudan.tlu.ee`). Second, I have provided some insight into the common roots and shared potential of networks with multiple node and link types, of higher-order topology, and a systematic science of art and culture.

3.22 Opening Talk: The Three Ages of Network Science – A Historical Perspective on Higher-Order Graph Models

Ingo Scholtes (Julius-Maximilians-Universität Würzburg, DE & Universität Zürich, CH)

Main reference Renaud Lambiotte, Martin Rosvall, Ingo Scholtes: "From Networks to Optimal Higher-Order Models of Complex Systems", Nature Physics, Vol. 15, p. 313-320, March 25, 2019
URL https://doi.org/10.1038/s41567-019-0459-y

Starting from a historical perspective on what I propose to consider "three ages" of network science, in the opening talk I gave an overview of different modelling frameworks that address different types of higher-order information and dependencies in complex networks. Addressing the challenge of dyadic interactions with multiple types, a first category of higher-order models includes signed graphs, multiplex networks and multi-layer networks. The second category of models includes simplicial complexes, hypergraphs, and motif-based network models, which can be used to address the challenge of modelling data with polyadic, i.e. non-dyadic, relationships. A third category of models uses higher-order Markov chains, memory networks, or high-dimensional De Bruijn graphs to model higher-order dependencies in time-ordered and sequential data. Following this categorisation, I presented three cross-cutting challenges that require a collaboration between researchers who address these different modelling frameworks. The first challenge addresses the practicality of higher-order models for data science practitioners, e.g., considering computational complexity, data efficiency, model dimensionality, and the need for intuitive and efficient visualisations of high-dimensional models. A second challenge is due to the curse of dimensionality that is common in higher-order models, which introduces the challenge of generalisability and model selection. A third challenge is the development of a unified perspective that combines different higher-order modelling frameworks to address complex data sets like, e.g. time-ordered or multi-type polyadic relationships.

References

1 Jürgen Hackl, Ingo Scholtes, Luka Petrović, Vincenzo Perri, Luca Verginer, Christoph Gote. *Analysis and visualisation of time series data on networks with pathpy*. In Proceedings of the 11th Temporal Web Analytics Workshop (TempWeb 2021) in conjunction with The Web Conference 2021, Ljubljana, Slovenia, April 2021

2 Vincenzo Perri and Ingo Scholtes. *HOTVis: Higher-Order Time-Aware Visualisation of Dynamic Graphs*. In Proceedings of the 28th International Symposium on Graph Drawing and Network Visualization (GD 2020), Vancouver, BC, Canada, September 15-18, 2020

3 Renaud Lambiotte, Martin Rosvall, Ingo Scholtes. *From Networks to Optimal Higher-Order Models of Complex Systems*. In Nature Physics, Vol. 15, p. 313-320, March 25 2019

4 Ingo Scholtes. *When is a Network a Network? Multi-Order Graphical Model Selection in Pathways and Temporal Networks*. In KDD'17 – Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Nova Scotia, Canada, August 13-17, 2017

5 Ingo Scholtes, Nicolas Wider, René Pfitzner, Antonios Garas, Claudio Tessone and Frank Schweitzer. *Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks*, In Nature Communications, Vol. 5, Article 5024, September 24, 2014

3.23 Motifs for Processes on Networks

Alice Schwarze (University of Washington – Seattle, US)

Joint work of Alice Schwarze, Mason Porter
Main reference Alice C. Schwarze, Mason A. Porter: "Motifs for processes on networks", CoRR, Vol. abs/2007.07447, 2020.
URL https://arxiv.org/abs/2007.07447

The study of motifs in networks can help researchers uncover links between the structure and function of networks in biology, sociology, economics, and many other areas. Empirical studies of networks have identified feedback loops, feed-forward loops, and several other small structures as "motifs" that occur frequently in real-world networks and may contribute by various mechanisms to important functions in these systems. However, these mechanisms are unknown for many of these motifs. We propose to distinguish between "structure motifs" (i.e., graphlets) in networks and "process motifs" (which we define as structured sets of walks) on networks and consider process motifs as building blocks of processes on networks. Using the steady-state covariances and steady-state correlations in a multivariate Ornstein–Uhlenbeck process on a network as examples, we demonstrate that the distinction between structure motifs and process motifs makes it possible to gain quantitative insights into mechanisms that contribute to important functions of dynamical systems on networks.

3.24 Higher-Order Models of Group Formation

Frank Schweitzer (ETH Zürich, CH)

Joint work of Frank Schweitzer, Georges Andres
Main reference Frank Schweitzer, Georges Andres: "Social nucleation: Group formation as a phase transition", CoRR, Vol. abs/2107.06696, 2021.
URL https://arxiv.org/abs/2107.06696

I talk about the dynamics of group formation, for a good reason: Group structures can be represented as polyadic interactions and are thus accessible by higher-order network models. But group formation is inherently driven by social mechanisms: homophily, cost/benefit evaluation, restricted access to resources, competition, to name a few. The question is: how are these social mechanisms preserved in a higher-order representation? In my short presentation, I provide a model that works for first-order networks, combining agent-based modelling with rules for network formation. Would similar results be achievable with higher-order models? How should these models look like? Would we gain anything beyond what the first-order network model already provides?

3.25 Higher-Order Models and Responsible Machine Learning

Markus Strohmaier (RWTH Aachen, DE)

In my talk I am exploring issues and challenges related to Responsible Machine Learning on Social Networks. I will argue that traditional methods for evaluating machine learning models need to be expanded to include and consider social challenges such as polarisation, inequality, exclusion or discrimination that are potentially arising from the deployment of machine learning techniques in social settings. I conclude with an outlook of potential avenues for further research.

3.26 Network Evolution and Spacetime Networks as Higher-Order Graphs

Chester Tan (National University of Singapore, SG)

In this lightning talk, I propose the following two questions: (1) Can higher order random network evolution models (e.g. higher order preferential attachment) be represented and analysed as higher order path sequence networks and vice versa. (2) How can higher order networks be represented meaningfully and analyzed usefully in spacetime?

3.27 Generative Models for Higher-Order Interactions

Anatol Wegner (University College London, GB)

The talk briefly introduced generative models for higher order interactions that include interactions that can take the form of any simply connected motifs. These models include a wide variety of higher order structures that go beyond cliques while remaining analytically tractable. I discussed the use of these models in inference based methods that can be used to obtain higher order representations of networks and raised potential applications of such generative models in graph based machine learning.

4 Working groups

4.1 Unification of Higher-Order Models

Unai Alvarez-Rodriguez (Universität Zürich, CH), Ginestra Bianconi (Queen Mary University of London, GB), Natasa Przulj (Barcelona Supercomputing Center, ES), Maximilian Schich (Tallinn University, EE), Alice Schwarze (University of Washington – Seattle, US), Leo Torres (Northeastern University – Boston, US), and Anatol Wegner (University College London, GB)

Despite *unification* being a newcomer to the higher-order jargon, its popularity skyrocketed in Dagstuhl, to the point of being shortlisted as a key discussion topic for the higher-order interactions field. Indeed, it was the preferred option for eight researchers in the working group allocation round, making it the most voted with twice the support of its most successful competitor. These values indicate a substantial acceptance of the unification discourse introduced during the first part of the seminar, where unification was presented as a quest towards a Utopian model merging multi-type, multi-time and multi-body interactions in a single formalism. The minimal desirable purpose of unification would be to develop a shared perspective that clarifies the relation, mutual difference and gap of alternative paradigms, including hypergraphs, simplicial complexes, multi-layer networks and temporal higher-order networks.

The working group on unification was gathered with the goal of highlighting what we stand to gain from the unification of different higher-order models. Our initial exchange was a virtual round table where we took turns to share our views on unification. In this process we identified two opposite aspects to refine in further explorations. The first one is the purely theoretical challenge of merging different types of higher-order approaches for the sake of addressing rich classes of dynamics. The second one is finding practical applications where a unified formalism is preferred over already existing ones. In summary, unification should aim not just for mathematical elegance and instead prove to be useful also for practitioners.

For the remaining days the team organised an asynchronous brain storming (deviating from more-typical synchronous Dagstuhl-style discussion, due to the hybrid nature of the event) to incorporate on-line as well as on-site participants with different time zones. Every member of the group then started a search for potential benefits of unification. This search lead to the following findings:

One of the profitable byproducts of unification is model compatibility. A unified model containing current frameworks as particular cases would provide a common language for higher-order phenomena which may enable combining results obtained independently in different domains of research on higher-order networks.

Another idea that we discussed was the use of unification as a principle for workflow automation. For researchers interested in applying network science to case studies, one of the first important steps is to decide which is the type of higher-order model that best describes a data set. Choosing a wrong model may lead to misleading conclusions about the behaviour of systems. The standard procedure to tackle this problem is employing model selection techniques. A challenge in doing so is that many currently available models are incompatible with each other, because they do not allow mixtures between different features (multi-type,multi-time and multi-body). A unified model would overcome this rigid structure as it would include degrees of freedoms of different features. Furthermore, such model flexibility would also remove the otherwise necessary step of network-type selection and therefore simplify the work of applied researchers.

Knowledge graphs were another topic of the debate. Within the joint discussion, group members Ginestra Bianconi and Maximilian Schich have pointed out the relevance of knowledge graphs, from the perspective of applied mathematics/physics and socio-cultural domain expertise respectively. Within the unification working group, knowledge graphs, and by extension less generalised database models, such as relational databases, have been considered regarding their relevance towards unification of higher-order network research. From a mathematical perspective, knowledge graphs are relevant as their configuration and growth is likely out of sync with existing maximum entropy models of network growth (e.g. adding bespoke link motifs instead of n-simplices). From the perspective of domain experts, ranging from biology to socio-cultural disciplines, developing a deeper understanding of higher-order structure and dynamics of knowledge graphs is a desiderate that seems more or less obvious since about two decades. From this a joint challenge emerges that can now be tackled based on the recent advances of higher-order network science. Consequently a second notion of "unification" emerged in the discussion, where different approaches of higher-order network science can be tested against each other, while looking at knowledge graphs that permeate a broad spectrum of disciplines.

All in all, we were able to ground the original proposal by showcasing specific methodologies that would be improved by a unified model. Our working group concluded that there is a robust motivation for a unification of higher-order models, and that we can anticipate an increase in the research community's efforts towards unification in years to come.

4.2 Social Impact of Higher-Order Models

Leonie Neuhäuser (RWTH Aachen, DE), Fariba Karimi (Complexity Science Hub – Wien, AT), and Markus Strohmaier (RWTH Aachen, DE)

In the working group "Social impact of higher-order models", we discussed which unique challenges and opportunities arise when deploying higher-order approaches to modelling social systems.

First, we identify the potential of higher-order models to better capture the rich subtleties and nuances present in social systems that might be neglected or ignored when modelling social systems with lower order approaches. This might help address issues that can arise from lower order modelling approaches such as conveying or exaggerating biases existing in the data due to oversimplification. We give examples of different scenarios for each of the three main higher-order model streams: multi-way, multi-layer and temporal interactions.

Second, we also identify a potential of higher-order models to introduce new problems themselves that might have negative consequences on social systems, such as disadvantaging certain parts of a social system (groups, communities) or warping and changing the representation of social systems in undesirable ways. We identify two main challenges: data availability and model interpretation. With regard to the first point, additional degrees of freedom for a model creates additional possibility for bias and misrepresentation e.g. due to data availability.

Data resolution affects how well we can infer certain dimensions of a higher-order model in practice and how much the models generalises for certain groups. Additionally, there might be some dimensions that we particularly do not want to include in a model because they introduce bias. Secondly, the interpretation of the results of a higher-order model can be complicated by the higher-order model aspects, which have to be well motivated and backed up with theory. When constructing a model, we want to capture all aspects of a system which are relevant for a specific research question of interest. Additional model dimensions can possibly lead to more insights, but also to more misinterpretation of their meaning is not clear in a specific context.

In summary, higher-order models of social systems have the potential to help overcome limitations of existing lower order approaches, but also introduce new challenges which need to be addressed to avoid introducing unintentional harms that result from information captured by higher order approaches.

4.3 Applications of Higher-Order Models

Vincenzo Perri (Universität Zürich, CH), Gabriele DiBona (Queen Mary University of London, GB), Luca Gallo (University of Catania, IT), Christoph Gote (ETH Zürich, CH), Jürgen Hackl (University of Liverpool, GB), Desmond J. Higham (University of Edinburgh, GB), and Frank Schweitzer (ETH Zürich, CH)

This report summarises the discussion that has taken place during the breakout sessions of the working group focused on the topic of "applications of higher-order networks". The discussions covered a broad range of topics, which we report in what follows.

A practitioner's guide

In our examination we abstract from the details of an application to specific systems. We decided to draft a practitioner's guide such that practitioners, based on their domain knowledge, are equipped to determine if and when higher-order networks should be used and which question they can answer. We identify five questions to address before applying higher-order methods and outline possible answers.

Do I need higher-order models?

To use higher-order networks, we first need to understand if the question has higher-order characteristics. Higher-order patterns do not pertain to the network structure (i.e., which pairwise interactions occur) but emerge from more complicated relationships. We identify three dimensions that result in higher-order patterns:

1. sequential (causal) dependencies
2. group interactions
3. multiple node or edge types

The use of higher-order networks is beneficial only if the problem displays at least one of these characteristics.

Do my data allow the use of higher-order models?

Even if the problem has higher-order characteristics, the data at our disposal might not allow for the use of higher-order methods. The complexity of higher-order interactions leads to models with a higher number of parameters compared to standard methods. Such complexity raises constraints relative to data quantity, as more parameters have to be estimated, and quality, as inconsistencies in the data might lead to cascading effects. Model selection needs to be used to select the optimal model given the available amount of data. Possible cascading effects are still an open issue. We will re-encounter them in the last question.

How do I get the model?

Answering positively to the previous two questions establishes the conditions for a fruitful application of higher-order networks.

Now, the practitioner has to choose the more suitable higher-order formalism from the available ones. The choice might not be straightforward, and it will depend on both the question and the available data. Helping a practitioner answer this question requires the community to provide tools (tutorials, software, etc.) that allow practitioners to understand the use-cases of different higher-order formalisms and eventually compare or mix them.

How to analyse higher-order models

Analysing higher-order networks might not be as straightforward as for standard networks. The first key step is the choice of the data structure to use, which may affect both the flexibility and efficiency of a computational algorithm. Then, performing the analysis requires understanding the meaning of the interactions between the elements of the higher-order network. Additionally, we need to decide whether to use the higher-order information to predict higher-order or standard structures. Finally, one should also consider the problem of data quality underlined above (question 2). While steps forward have been made in considering the impact of incomplete or noisy data on standard networks, this topic has received little attention for higher-order networks.

How to interpret the results

The interpretation of higher-order methods' results is often not as straightforward as that of standard network methods. One challenge is to express patterns identified in the higher-order representation in terms of standard nodes. Depending on how we project from higher to lower order, we will retain more or different types of information. Additionally, the question of interpretation can not be separated from the other topics discussed above. Issues like data quality and quantity have to be considered when interpreting the results in order to be able to separate the model's sensitivity to changes in the data from the system's stability.

Conclusion

In our discussions, we identified the most challenging problems to be the ones regarding the interpretation of the results of a higher-order model in terms of a real-world problem. We suggest thinking of the minimal model that can explain the phenomenon, also considering simple networks. Even if this process has been undertaken when choosing the model, we should continue to question the choice of the model when interpreting the results.

4.4 Learning and Model Selection in Higher-Order Networks

Martin Rosvall (University of Umeå, SE), Rebekka Burkholz (Harvard School of Public Health – Boston, US), Timothy LaRock (Northeastern University – Boston, US), Vito Latora (Queen Mary University of London, GB), Kang-Ju Lee (Seoul National University, KR), Giovanni Petri (ISI Foundation – Torino, IT), Luka Petrovic (Universität Zürich, CH), Michael Schaub (RWTH Aachen, DE), Alice Schwarze (University of Washington – Seattle, US), and Michele Starnini (ISI Foundation, IT)

In the working group *Learning and model selection in higher-order networks*, we discussed the different roles models can have and how the specific task must decide the model choice. Our discussions focused on two model perspectives.

The first perspective concerned *the scope of what we aim to learn.* Are we interested in relational data – the structure of a network? Or are we interested in covariate data constrained by relational data – signals, metadata, and dynamical data? Or both?

Relational data describe interactions between at least two entities, the topology of a possibly higher-order network system. In a standard, dyadic, static setup, edges describe the system's topology, which we aim to model. For example, we observe a network sample and infer a probability distribution over graphs using a stochastic block model as a statistical model of the network.

In higher-order models, the types of relational data we recognise are significantly larger. Instead of considering only a single type of dyadic static relation between two nodes, we consider typed interactions (signed, multiplex, multi-layer), temporal interactions with a path-dependency, polyadic interactions, or any combination of them.

In covariate data, each data point is associated with a node, an edge, or a higher-order simplex. One can obtain covariate data from measurements at a given time or a sequence of time points. One can also obtain covariate data as a function of time from theoretical dynamical systems or computational simulations. Such state variables can capture a dynamical process that takes place on the interacting system. Examples include time series of electric signals at the different cortical areas of the brain measured through EEG brain imaging, infected individuals in the spreading of a disease across a population, or traffic flows in street networks.

The other perspective concerned the *objective of learning.* Are we interested in prediction or classification from incomplete data, or do we seek to discern important mechanisms of the system under study, or both? These objectives come with different trade-offs.

For predicting future interactions or classifying nodes based on incomplete data, we must balance model and data complexity to find a model that accurately describes the available data. When detecting the optimal order of a multi-order network model for pathway data, for example, we need to balance the increase in the likelihood of a more complex model with the increase in the complexity of the model. The richer the data we can collect, the more flexible models we can try to fit. With access to polyadic data, we may successfully fit a statistical hypergraph model. In contrast, we may need more data to infer the polyadic relations from dyadic relations. Similarly, rich covariate data may allow for a more detailed model.

To identify a system's important mechanisms, we should decide what assumptions to use for modelling relational and covariate data. We discussed a scenario in which we study the spread of an epidemic by observing time-stamped interactions between people, potentially augmented with information about the state of the nodes. We could consider these data in at least two ways. First, we may think about them as coming from an epidemic process that spreads on a temporal topology: the process runs continuously on top of each node, but the relations between people are only active for some time. Second, we may interpret our data as events generated by a point process on a latent but largely static interaction topology. In this example, our mechanism of interest and the respective model's ability to describe the data, should guide us in choosing a model. We may prefer a simple model with system-specific assumptions over a flexible model with a potentially better fit to the data than the simple model because a high model complexity obscures the mechanisms that we seek to identify.

We concluded that generalising statistical principles developed for networks to higher-order network models seems promising for trade-offs of model flexibility. By contrast, trade-offs of higher-order models that we develop to gain mechanistic insights are under-explored and require new computational and mathematical methods.

4.5 Benchmark Data and Evaluation Practices

Ingo Scholtes (Julius-Maximilians-Universität Würzburg, DE & Universität Zürich, CH), Stephan Günnemann (Technical University of Munich, DE), Andreas Hotho (Julius-Maximilians-Universität Würzburg, DE), and Jelena Smiljanic (University of Umeå, SE)

An important open issue that has been raised by several participants during their introductory statements is a lack of commonly used benchmark data and generally accepted practices to evaluate higher-order graph models. Mirroring the diversity of the higher-order graph community, this is a multi-faceted problem. The working group has identified three related challenges which are presented below. We used them to derive four opportunities for the higher-order graph community to improve their evaluation practices and – in turn – increase their impact on applications of network and data science.

4.5.1 Challenges for the Evaluation of Higher-Order Graph Models

Comparing Higher- vs. First-Order Models

An important first challenge is due to the common need to show what we gain by using higher-order graph models, as opposed to methods that are based on first-order graphs. To this end, researchers typically evaluate their models based on a variety of data mining, prediction, and modelling tasks. The choice of those tasks, as well as the choice of data set in which those tasks are addressed, is often informed by specific assumptions of the higher-order models that are being evaluated. This introduces potential issues for the *external validity*, i.e. it is not clear to what extent the obtained results generalise to other settings or data with higher-order characteristics that do *not* match the assumptions of a given modelling framework.

Comparison of Different Higher-Order Models

While the challenge above applies to each "paradigm" of higher-order graph models individually, a second challenge arises due to the growing number of different modelling paradigms that address the same higher-order characteristic of data or systems, e.g., the use of hypergraphs vs. simplicial complexes to model systems with polyadic interactions. To facilitate a fair comparison between such different modelling approaches, the community should establish standard benchmark data sets that exhibit higher-order characteristics, along with a set of clearly defined tasks and evaluation metrics that do not favour one or the other modelling paradigm. This would not only help practitioners to decide which modelling paradigm to choose for a specific system. It is also likely to improve our understanding of the advantages and disadvantages of different paradigms and the sometimes implicit assumptions they are based on.

Comparison of Models for Different Higher-Order Characteristics

As highlighted in the panel discussion, there is no single, *correct* type of higher-order graph model that could be used to model all networked systems. Instead, we are commonly confronted with systems that exhibit multiple higher-order characteristics at once such as, e.g., networks with temporally ordered, multi-typed, and polyadic interactions. A third important challenge for the community is thus to understand what we lose or gain by using models that capture only one of those characteristics. Given a modelling task in a system with temporally ordered polyadic interactions, is it preferable to use a hypergraph model that ignores the temporal ordering of interactions, or is it preferable to use a model that captures causal path while ignoring the fact that interactions are non-dyadic? To answer such questions, the community needs benchmark data and problems that support a comparison of higher-order graph models that address *different* higher-order characteristics in complex systems. We further need model-independent prediction or modelling tasks like, e.g., the prediction of interactions, node- or graph-level classification tasks, or forecasting the evolution of dynamical processes, that could be used to compare the performance of different higher-order graph models.

4.5.2 Opportunities to Improve Evaluation Practices

Based on the challenges outlined above, the working group has identified three opportunities for the higher-order graph modelling community, which we outline below.

Opportunity 1: Higher-Order Graph Benchmarks

A first opportunity is to establish benchmarks that can be addressed by different types of higher-order graph models, and which should be based on the following ingredients:

- data sets on networked systems with a given higher-order pattern (polyadic, multi-typed, temporal interactions, etc.)
- measure for model performance based on a given prediction or modelling task
- a baseline against which we compare model performance. Depending on the problem, this baseline can either be state-of-the-art techniques or, if we want to reason about the benefit of higher-order models, lower- or first-order versions of a given model.

We note that the following online repositories for network data contain data sets that may have the necessary characteristics for such benchmark data:

- **SNAP** `https://snap.stanford.edu/data/` (temporal, multi-layer, polyadic)
- **netzschleuder** `https://networks.skewed.de/` (temporal, multi-layer, polyadic)
- **Konect** `http://konect.cc` (temporal, multi-layer)
- **Sociopatterns** `http://www.sociopatterns.org` (polyadic, temporal)

Referring to the first challenge, a common goal in the study of higher-order graph models is to assess the advantage over techniques based on first-order graphs. The question which first-order graph model should be chosen to facilitate a fair comparison is non-trivial and has – in some cases – been addressed in an unsatisfactory fashion. As an example, consider a comparison of a model with weighted higher-order interactions with an unweighted first-order graph model. The results of such an experiment do not tell us a lot about the impact of higher-order interactions, since it mixes the effect of a projection to first-order interactions with the effect of reducing a weighted to an unweighted graph model.

How can we define baselines that enable a fair comparison? One possible approach is to apply higher-order graph models to a version of a data set, where the higher-order dependencies have been selectively removed. E.g. for memory networks or De Bruijn graph models of paths in temporally ordered interactions, we can use data where time stamps have been randomly reshuffled, which removes any temporal correlations in the ordering while preserving information on the temporal distribution and the frequency of interactions. Similar randomisation approaches that maintain first-order characteristics but destroy higher-order patterns may be possible for data on polyadic interactions or multi-typed relations.

Opportunity 2: Using Higher-order Models to improve on Standard Graph Mining Problems

We can evaluate higher-order graph models in standard graph mining problems. This allows us to compare higher-order models against state-of-the-art algorithms as well as to different higher-order graph modelling frameworks with each other. Examples that can be potentially addressed based on different types of modelling paradigms include:

- Node ranking, where the ranking is based on different higher-order generalisations of centrality measures
- Node classification, where classes are assigned to nodes in a way that incorporates higher-order patterns
- Link prediction, where dyadic interactions are predicted based on models incorporating higher-order characteristics of the data (e.g. time or multiple types)
- Graph clustering, where clusterings in higher-order graph spaces are projected to clusters in a first-order graph
- Vector-space embedding, where vector representations of nodes or links are derived from a higher-order generalisation of similarity/dissimilarity rankings.

Recently, an extensive evaluation platform for graph mining problems has been proposed in [1]. It would be a worthwhile effort to consider whether a similar set of problems and evaluation practices, as well as convenient solutions for standardised data splitting, sampling and shuffling, could be combined with some of the data sets above to establish a higher-order graph benchmark that is accepted by the community.

Opportunity 3: Defining Novel Benchmark Problems involving Higher-Order Patterns

Apart from evaluating higher-order graph models in terms of standard graph mining and learning tasks, an interesting prospect for the definition of novel evaluation practices is that some of those problems can be naturally and meaningfully translated to the higher-order primitives used by different modelling frameworks. Examples include:

- multi-layer link prediction, where we predict links given a layer, a layer given a link, or both the layer and the link
- hyperedge or k-simplex prediction, which can be easily defined for co-occurrence or co-authorship data
- hyperedge clustering, which can be used to identify, e.g. groups of similar collaboration patterns
- path ranking, where rather than identifying nodes we identify node sequences or sets that are most important, e.g, for spreading patterns or information propagation
- path clustering, where we identify sets of paths observed in a time series data set that are more similar to each other than to other paths
- path prediction or classification, which can be useful for applications in click stream data, information propagation, as well as end-to-end vs. next-element prediction in sequential data

Opportunity 4: Model Dimensionality and Data Sparsity

One of the key challenges that we face in the study of higher-order models is that we often increase the dimensionality of the model, i.e. we add degrees of freedom that – on the one hand – enable us to more accurately model systems but – on the other hand – potentially

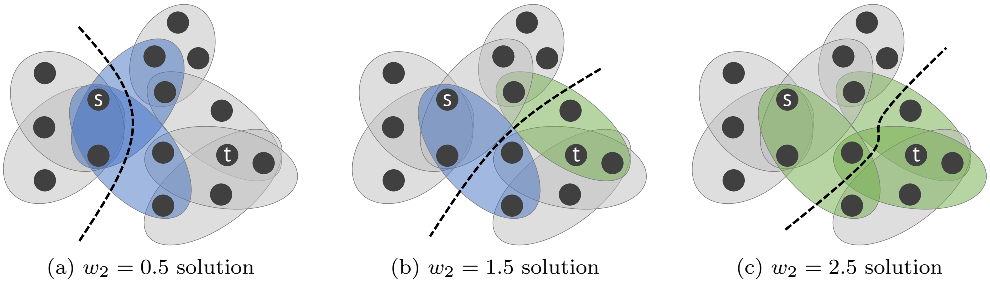


Figure 2 A figure from Veldt et al. [31] that illustrates how a hypergraph can be split into two pieces as a characteristic parameter changes in a simple scenario. Crucial to this drawing is showing which hyperedges are separated in the partition, which is simple in the convex-set drawing of a hypergraph used in the figure. It would be challenging to illustrate this figure with a node-and-edge drawing – even of the bipartite network representation of a hypergraph. This motivates our questions of how to visualise these hypergraphs.

increase computational complexity and pose challenges for the generalisability, robustness, and data efficiency of our models. However, this challenge also introduces opportunities for a definition of evaluation practices that go beyond mere model accuracy. Exemplary aspects that should be incorporated in the evaluation of higher-order graph models include:

- **model robustness**: How robust is a higher-order model against the introduction of noise and how does the inclusion of higher-order primitives specifically change the robustness compared to first-order graphs?
- **model size**: How much memory does a model consume, how many degrees of freedom does it have and how does the model size depend on key system parameters like the number of nodes or the density of (higher-order) interactions?
- **data efficiency**: How much data do we need to reliably model higher-order patterns in a given data set?
- **scalability**: How much time do we need to learn a model or to make predictions and how does the computational complexity depend on the size of the data (in terms of number of observations) or the size of the system?

References

1 Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, Jure Leskovec: *Open Graph Benchmark: Datasets for Machine Learning on Graphs*, arXiv 2005.00687, `https://arxiv.org/abs/2005.00687`

4.6 Visualisation and Interpretability of Higher-Order Networks

Chester Tan (National University of Singapore, SG), Daniel Edler (University of Umeå, SE), Anton Eriksson (University of Umeå, SE), David F. Gleich (Purdue University – West Lafayette, US), and Lisi Qarkaxhija (Koper, SI)

Visualisation is essential to understanding and *interpreting* data – it facilitates recognising norms and outliers in non-human readable data by representing data in more accessible forms such as graphs. Recognising its importance and infancy in its use with higher-order networks, we decided to form a working group to discuss exactly the Visualisation and Interpretability of Higher-Order Networks. In this working group we discussed the current state of the art and its limitations, and deduced some notable key opportunities for development.

4.6.1 Current literature

We first sought a common understanding of the state of the art by discussing known data visualisation tools and identifying if they support higher-order visualisations, and the type(s) of higher-order visualisations they facilitate.

Relevant Tools from Workshop Participants

Acknowledging while looking to leverage our biases, we first highlighted the following most relevant tools developed by participants in this Dagstuhl: **pathpy** [26] is a Python package which provides an automated framework to deduce the most likely Markov order for sequential data, and visualise such data as its most probable higher-order De Bruijn Graph. **Infomap Network Navigator** [12] is an interactive web application that generates a zoomable map for networks clustered with InfoMap. While it supports higher-order networks, it doesn't draw the raw network but instead the hierarchical modular network structure using existing tools for force directed layouts, augmented with new constraints, to support the higher order visualisation of state networks. **LocalGraphClustering** [15] is a Python (and Julia) package designed to identify local structures in networks and visualise how their local groups compress into low-rank representations, primarily to highlight differences in the way various algorithms *see or experience* network structures.

Desktop GUI Applications

We then discussed well-known Desktop GUI applications for visualising network data such as **Gephi** [2] and **Cytoscape** [28]. Neither requires any programming, and both support importing a network and associated metadata from various file types. Gephi is an open source cross-platform application that is able to visualise and analyse large networks, while Cytoscape is an open source software platform for visualising complex networks and integrating these with any type of attribute data, focusing on bioinformatics data. Notably, neither have dedicated support for higher-order network representations.

General Graph Toolboxes

While these relatively well-known GUI applications did not support visualising higher-order data, we noted that there are many software libraries designed to draw networks, or support programs that work with networks and produce visualisations, though they all require some programming familiarity: **MuxViz** [10] is an R package for the analysis and visualisation of interconnected multi-layer networks. **NetworkX** [17] is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. It has many functions to help draw networks and a number of functions to compute force directed-like layouts. **D3.js** [7] is a JavaScript library that transforms data to interactive visualisations in the browser. It includes force-directed graph layout algorithms. **iGraph** [9] is a network analysis and visualisation software written in C++ with bindings to R and Python. This includes tools to compute network layouts (coordinates) for each node from a variety of methods. These scale to large graphs with millions of nodes in reasonable time-frames (hours). **GraphViz** [13] is a free and open source graph visualisation program in DOT language scripts. Like **iGraph**, it supports a variety of layout algorithms. **Graph-tool** [22] is a Python package which is used to produce useful visualisations, statistical analysis and manipulation of networks. It known for its performance since its main algorithms and data structures are written in C++.

4.6.2 The Many Forms of Higher-Order Networks and their Visualizations

From these known tools and frameworks, we identified some common models of higher-order network models and visualisations which elucidated even more higher-order visualisation methods.

Hypergraphs

One existing means of visualising higher-order data is through sets of nodes. Berge used this technique in his book on hypergraphs [5]. A downside to this visualisation approach is that an inaccurate drawing may result in nodes appearing to belong to hyperedges that they do not contain.

Bipartite Networks

Another means of visualizing higher-order data is through a bipartite network. This visualisation corresponds to using the incidence matrix of a graph as the adjacency matrix of a bipartite graph, and is further related to what is called a *star* expansion of a hypergraph structure.

Space Embedded Networks

Some network embeddings [23, 16] produce a set of coordinates for a network by minimising an energy function over small sets sampled from the network. These have been extended to higher-order data as well [11, 29], where the output is typically set of coordinate in a high-dimensional space, with 1 coordinate per node. These can be subsequently processed with tSNE [30] or UMAP [20] or alternative dimension reduction techniques, though these dimension reduction techniques do have their biases and compromises [32]. Some methods to embed networks in spacetime [8] have also been explored.

4.6.3 Key Opportunities and Takeaways

From these discussions, we deduced that network visualisation serves two primary roles: **(1) to elucidate results in studied network data**, as a static figure or a short movie – a form of network data visualisation that has appeared on the cover image of many highly regarded interdisciplinary journals – and **(2) to facilitate the discovery of features in data**, where tools often have interactive graphical components that make it easy to manipulate diverse data.

In an ideal scenario, higher-order graph models and data can simultaneously be visualised and interpreted by existing tools and also pose new challenges and opportunities beyond them. This apparent contradiction follows because, while there are many ways to translate higher-order data into network-like representations, each interpretation has its biases which obscures some properties of the higher order data over others. To illustrate this point, see Figure 2. The contents of the figure require its higher-order data to be expressed in a new or different way. Similar figures, with new visualisation strategies, arise in many papers introducing higher-order topics [21, 25, 3]. Many of these figures use non-standard visual representations of higher-order data that are difficult or impossible to replicate with standard tools.

Dimensions of Higher-Order Network Visualisations

These two primary roles make just one of the many notable *dimensions* we found network visualisations to have, including: **Interactive vs Static** For data exploration and online showcases, interactive visualisations are preferred over static visualisations, while paper figures are typically static, and have stricter requirements for legibility. **Multi-layer vs Single-layer** With usually an ordinary 2D layout of the network in each layer, a multi-layer network visualisation can either be in two or three visual dimensions with each layer drawn as cards side by side. **10 vs 1000 vs millions of nodes** It is easy to visualise entire small networks, but significantly more difficult to interpret visualisations of large networks. **Annotated vs Non-annotated** While annotations in our visualisations aid comprehension of the data and provide additional information about topics that aren't evident to the human eye, having a lot of annotations make visualisations undesirably noisy and cluttered. **Raw Network vs Clusters** In most interactive maps, the level of detail shown depends on the zoom level. A hierarchical clustering algorithm can similarly help us navigate a very large network, overcoming potential graphical or computational limitations. **Node-edge Plots vs Feature Embeddings** A standard way of drawing a network is to plot both nodes and edges, and for that to look nice for a broad range of networks, it typically requires a force-directed layout algorithm, or similar, that minimises edge crossings. On the other hand, if we want to highlight some features of our data, we may, for example, embed nodes in a high-dimensional space of node features, and employ dimensionality reduction techniques to layout nodes in two or three visual dimensions. **Instantaneous vs Evolving** visualisations. **GUI vs Programming Interfaces** Many GUI apps contain various tools that are often more user friendly and less time consuming than their programming counterparts, which, instead, often offer greater customisability and reproducibility.

We note that the seemingly opposing poles in each of these dimensions listed above are not necessarily mutually exclusive – e.g. a GUI app could have a programming interface, and that the current state of the art supports only a small subspace of these dimensions. The following example may further elucidate how the current lack of tools and techniques hampers research: in Gleich's work this past summer studying a set of emails surrounding the US government's response to the COVID pandemic, we sought to use time-varying hypergraphs to represent the email information [4, 6]. Hypergraphs were key to the representation as often a single email will bridge a number of different organisational entities in the strongly hierarchical government agencies. Yet, without current tools supporting them adequately, the team had to implement rudimentary ideas as surrogates for investigations they wished to conduct. This make it significantly more difficult to interpret the data using the growing set of higher-order data tools produced by the community.

A List of Some of the Many Forms of Higher-Order Network Visualisations

As another key takeaway we briefly list some of the common forms of higher-order network visualisations we discussed: **hypergraphs**, **simplicial complexes**, **bipartite networks**, **multi-layer networks**, **multiplex networks**, **higher-order space embedded networks**. This list highlights, among other things, the variety in visualisation methods and language of, which made discussions especially challenging and interesting, and an apparent recurring and arguably most interesting theme throughout this Dagstuhl.

4.6.4 Key Opportunities

Finally, we concluded that the following are two key opportunities in the field for further study and development: **(1) Interactive tools for higher-order data** There is ongoing work on tools to work with higher-order representations of processes on network data by the workshop participants in the Infomap Network Navigator (see the paragraph below). Additional tools have identified similar weaknesses and aspects. See, for instance, open issues on the `Gephi` and `graphviz` software to support hypergraph drawings. **(2) Revisiting fundamental ideas** Many existing network visualisations involve a variety of studies closely related to many applied algorithms. For instance, spectral network drawing was originally proposed as an energy minimisation technique [18] that predated Fiedler's work on Laplacian eigenvectors [14]. There are now higher-order generalisations of many similar ideas [19] (and references therein). Many successful node placement techniques for graph visualisation are based on force simulations (e.g. Force-Atlas, etc.) Higher-order data present novel opportunities to evolve this research. For instance, recent work on force directed placement [1]. Related work includes efficient molecular dynamics simulations [24], which suggest novel types of possible forces for higher-order data.

References

1 Naheed Anjum Arafat and Stéphane Bressan. Hypergraph drawing by force-directed placement. In Djamal Benslimane, Ernesto Damiani, William I. Grosky, Abdelkader Hameurlain, Amit Sheth, and Roland R. Wagner, editors, *Database and Expert Systems Applications*, pages 387–394, Cham, 2017. Springer International Publishing.

2 Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*, 2009.

3 Austin Benson, David F. Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

4 Austin Benson, Nate Veldt, and David F. Gleich. fauci-email: a json digest of anthony fauci's released emails. *arXiv*, cs.SI:2108.01239, 2021. Code and data available from `https://github.com/nveldt/fauci-email`.

5 Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

6 Natalie Bettendorf and Jason Leopold. Anthony fauci's emails reveal the pressure that fell on one man. BuzzFeed News, `https://www.buzzfeednews.com/article/nataliebettendorf/fauci-emails-covid-response`, June 2021.

7 Mike Bostock. D3.js – data-driven documents, 2012.

8 James R. Clough and Tim S. Evans. Embedding graphs in lorentzian spacetime. *PLOS ONE*, 12(11):e0187301, Nov 2017.

9 Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

10 Manlio De Domenico, Mason A Porter, and Alex Arenas. Muxviz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, 3(2):159–176, 2015.

11 Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 135–144, New York, NY, USA, 2017. Association for Computing Machinery.

12 D. Edler, A. Eriksson, and M. Rosvall. *The Infomap Software Package*, 2021.

13 John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen North, Gordon Woodhull, Short Description, and Lucent Technologies. Graphviz — open source graph drawing tools. In *Lecture Notes in Computer Science*, pages 483–484. Springer-Verlag, 2001.

14 Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

15 Kimon Fountoulakis, David F. Gleich, and Michael W. Mahoney. A short introduction to local graph clustering methods and software. In *Book of Abstracts for 7th International Conference on Complex Networks and Their Applications*, pages 56–59, 2018.

16 Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

17 Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

18 Kenneth M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

19 Pan Li and Olgica Milenkovic. Submodular hypergraphs: p-laplacians, Cheeger inequalities and spectral clustering. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3014–3023, Stockholm Sweden, 10–15 Jul 2018. PMLR.

20 Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

21 Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

22 Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014.

23 Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA, 2014. ACM.

24 Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.

25 Martin Rosvall, Alcides V. Esquivel, Andrea Lancichinetti, Jevin D. West, and Renaud Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature Communications*, 5(4630), 2014.

26 Jürgen Hackl, Ingo Scholtes, Luka V Petrović, Vincenzo Perri, Luca Verginer, Christoph Gote. Analysis and visualisation of time series data on networks with pathpy In *Proceedings of the 11th Temporal Web Analytics Workshop (TempWeb 2021)* held in conjunction with The Web Conference 2021, Ljubljana, Slovenia, April 2021

27 Ingo Scholtes. When is a network a network? multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1037–1046, New York, NY, USA, 2017. Association for Computing Machinery.

28 Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

29 Justin Sybrandt, Ruslan Shaydulin, and Ilya Safro. Hypergraph partitioning with embeddings. *IEEE Transactions on Knowledge and Data Engineering*, page 1–1, 2020.

30 Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

31 Nate Veldt, Austin R. Benson, and Jon Kleinberg. Hypergraph cuts with general splitting functions, 2020.

32 Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016.

5 Panel discussions

5.1 What are Higher-Order Graph Models?

Ingo Scholtes (Julius-Maximilians-Universität Würzburg, DE & Universität Zürich, CH)

Joint work of all seminar participants who participated in the panel discussion

Following the sessions with brief introductory statements and a first meeting of the working groups, participants spontaneously engaged in an open-end evening discussion on what they identified as an important open issue in the community: we lack a commonly agreed-upon definition of higher-order graphs and networks. In particular, different researchers use the term "higher-order" to refer to different characteristics of either networked systems, network models, or data.

The discussion revealed that the seminar participants agree that, as a community, we must more clearly distinguish between (i) complex networked systems that consist of many interacting elements, (ii) high-dimensional data that capture those interactions between system elements, and (iii) graph or network *models* of those systems. Commonly used graph models with a single type of dyadic, static links as the simplest possible – but neither the only nor necessarily optimal – graphical representation of data on element-element interactions that can be used to generate insights into complex systems. The analysis of such *first-order* graph model can nevertheless be reasonable if (a) we know that the system exclusively features a single type of interaction between pairs of elements, (b) we only have access to relational data capturing pair-wise interactions even though we know interactions in the system are more complex, or (c) we seek to understand which of the system's characteristics can already be explained by first-order interaction.

A clear distinction whether the term *network* refers to the system to be modelled, the structure of the available data, or the mathematical model used to analyse the data is often missing. This complicates the rigorous definition of higher-order graphs and networks and has – at times – fostered misunderstandings between different communities regarding whether a given type of model should be considered *higher-order* or not. Summarising the results of the panel discussion, in the following we take two perspectives that focus on the characteristics of the *model* and the *data* on the system to be modelled.

Model perspective

A first approach to define *higher-order graph models* considers the mathematical representation used to study the topology, i.e. who can influence whom and how, of a complex system. First-order graph models assume that the topology (and the resulting behaviour) of a complex system can be reduced to a set of dyadic edges, which can be mathematically represented in terms of adjacency, transition, or Laplacian matrices with $\mathcal{O}(n^2)$ entries, where n is the number of elements or vertices in the system. Despite major differences in terms of modelling assumptions, a common feature of all higher-order graph models – be it hypergraphs, simplicial complexes, memory networks, or high-dimensional De Bruijn graphs – is that they require mathematical notations with higher dimensionality than common matrix representations. This characteristic of different higher-order models translates to similar ideas, e.g. the use of tensors and flattened representations of high-dimensional linear operators, as well as common challenges, e.g. computational challenges and dimensionality issues in higher-order graph learning methods.

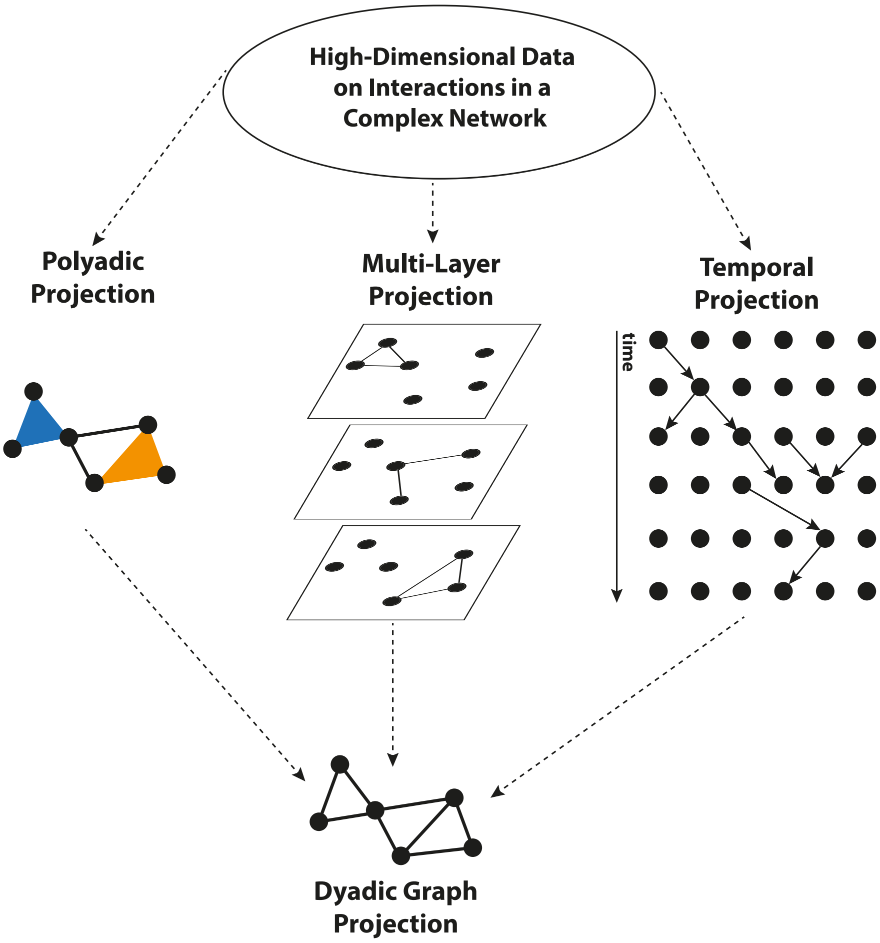


Figure 3 Different higher-order graph models can be viewed as different projections of high-dimensional data on interactions in complex systems along different dimensions, where single-typed dyadic graphs can be viewed as a maximally simple projection of those different higher-order models.

While it may seem intuitive that the use of higher-order graph models either requires networked systems with non-dyadic interactions or data with higher-order characteristics, this is not necessarily the case. Several graph learning techniques make use of higher-order primitives, which – however – are not used to model higher-order structure in the underlying systems or data. A prominent example is node2vec[16], which can be viewed as a random walk in a second-order graph model, but which is usually applied to data on graphs with simple dyadic interactions. Here, the higher-order model is rather used to encode non-local features of the graph topology into a model for a dynamical process on the graph. The question whether we consider such a model as *higher-order* graph model or not highlights that we may need to look beyond the characteristics of the data.

Data perspective

An interesting point raised during the discussion was that it may actually be easier to reach consensus on a definition of higher-order characteristics in data, rather than higher-order characteristics in *graph models*. From the perspective of "first-order" graph theory or network science, higher-order characteristics in data can be defined as any information that goes beyond the specification of dyadic edges, i.e. any data that gives rise to more than a subset of the Cartesian product of vertices. Examples for such data with higher-order characteristics include but are not limited to:

- multiple sets of edges capturing interactions with different properties (such as multi-typed or signed interactions that invalidate a simple transitive treatment of edges)
- data capturing polyadic interactions, e.g. tuples or sets with a cardinality higher than two
- ordered or time-stamped sequences of dyadic or polyadic interactions

In network science, such higher-order characteristics in data is often reduced to dyads because we want to apply standard graph algorithms or network analysis techniques. In contrast, as higher-order graph models we can define any model that seeks to more faithfully represent (one or more) of the higher-order characteristics present in data on complex systems that influence how nodes can directly or indirectly influence each other. Notably, different types of higher-order graph models can destroy different higher-order characteristics in the data: A hypergraph model of time-stamped polyadic interactions destroys higher-order patterns that are due to the timing and ordering of interactions, while higher-order De Bruijn graph models for temporally-ordered dyadic links destroy patterns that are due to the polyadic nature of the interactions. The combination or unification of different higher-order modelling frameworks to capture multiple higher-order characteristics of data is an important open challenge that must be addressed by the community.

We finally noted that the large popularity of graph models with dyadic links or edges often leads to the unfortunate development that the data collection and engineering process is informed by the features of simple graph models rather than the modelling process being informed by the higher-order characteristics of the system to be modelled. As an example, data on co-authorship networks are often provided in the form of dyadic relationships between authors even though the underlying interactions are fundamentally non-dyadic. Similarly, data is often tailored to the application of time-slice snapshot network models, discarding information that would be important to infer higher-order patterns in the temporal ordering of interactions. This leads to what could be called a “data bottleneck” that hinders the application of higher-order graph structures to model the higher-order characteristics present in many real complex systems.

Participants

- Unai Alvarez-Rodriguez
Universität Zürich, CH
- Luca Gallo
University of Catania, IT
- David F. Gleich
Purdue University –
West Lafayette, US
- Christoph Gote
ETH Zürich, CH
- Jürgen Hackl
University of Liverpool, GB
- Andreas Hotho
Universität Würzburg, DE
- Kang-Ju Lee
Seoul National University, KR
- Leonie Neuhäuser
RWTH Aachen, DE
- Vincenzo Perri
Universität Zürich, CH
- Giovanni Petri
ISI Foundation – Torino, IT
- Luka Petrovic
Universität Zürich, CH
- Martin Rosvall
University of Umeå, SE
- Michael Schaub
RWTH Aachen, DE
- Maximilian Schich
Tallinn University, EE
- Ingo Scholtes
Universität Würzburg, DE &
Universität Zürich, CH
- Frank Schweitzer
ETH Zürich, CH
- Markus Strohmaier
RWTH Aachen, DE

Remote Participants

- Federico Battiston
Central European University –
Vienna, AT
- Ginestra Bianconi
Queen Mary University of
London, GB
- Rebekka Burkholz
Harvard School of Public Health –
Boston, US
- Giulia Cencetti
Bruno Kessler Foundation –
Trento, IT
- Gabriele DiBona
Queen Mary University of
London, GB
- Daniel Edler
University of Umeå, SE
- Tina Eliassi-Rad
Northeastern University –
Boston, US
- Anton Eriksson
University of Umeå, SE
- Stephan Günnemann
TU München, DE
- Heather Harrington
University of Oxford, GB
- Desmond J. Higham
University of Edinburgh, GB
- Fariba Karimi
Complexity Science Hub –
Wien, AT
- Danai Koutra
University of Michigan –
Ann Arbor, US
- Renaud Lambiotte
University of Oxford, GB
- Timothy LaRock
Northeastern University –
Boston, US
- Vito Latora
Queen Mary University of
London, GB

- Yamir Moreno
University of Zaragoza, ES

- Natasa Przulj
Barcelona Supercomputing Center, ES

- Lisi Qarkaxhija
Koper, SI

- Alice Schwarze
University of Washington – Seattle, US

- Jelena Smiljanic
University of Umeå, SE

- Michele Starnini
ISI Foundation, IT

- Chester Tan
National University of Singapore, SG

- Leo Torres
Northeastern University – Boston, US

- Anatol Wegner
University College London, GB