Publishing in the Life Sciences

Grant Roy

Abstract

We consider the process of publishing as one of Knowledge Representation (KR). Traditionally, this knowledge is created textually, however we will argue that by approaching publishing more rigorously from the perspective of formal knowledge representation, there are significant gains to be had in overall knowledge production and understanding. At the heart of our proposal is a dual graphical/logical system for both specifying procedures (both experimental and computational) that result in knowledge, and a formal KR system for capturing the results of these procedures.

1 Introduction

Academic publishing, since the first journals began in 1665, have been created to facilitate the process of peer review (and thus establish scientifically credible results), and disseminate results. Traditionally, and contemporaneously, this has meant a textual representation, allowing for different and mixed modalities of natural language, mathematics, and images and or/diagrams. Ideally, this end product, the paper, is composed of verified (novel) understanding, and detailed methodological descriptions of how this understanding was gained. Thus the paper contains both a description of results, and also a prescription of how those results were obtained. From our post information age vantage point, we must critically assess this process of scientific knowledge production, asking: do we now have at our disposal tools to improve this process, and better align it with the goals stated at its outset. Further, if we restrict our domain to the life sciences, can this focus offer new perspectives and opportunities, particularly when we take into account advances in automation (both machine and intellectual), and increased access to technology.

Herein we will take the view that increasing robotic, and thus information science grounded experimental workflows will grow in importance, and in the near future dominate the life sciences, starting with biomedical research and applications. The key idea is that by understanding the control technology, which is specified as a program, we can understand the and sculpt scientific processes by understanding and sculpting programming languages. Thus we argue that focusing on programming languages, their representations, and critically their machine interpretability. The perspective we take, that of elevating programming languages/paradigms as an area focus, is grounded in an insightful analogy of *computational trinitarianism* [**?**] (and its extension to homotopical trinitarinism [8]). Over the last decade, an understanding has developed with respect to how logic, programming languages (homotopy type theory), and category theory (higher toposes) are related, in the sense that any improvements in one translate into the other. This is important because category theory is a very powerful language for knowledge representation and process description. By interpreting category theory in homotopy type theory, we can gain a very powerful type theory as a programming language. Our central idea is that as a conceptual framework, cateogory theory presents a unique opportunity to transform knowledge representation in science, and that as a uniquely powerful computational interpretation of category theory, homotopy type theory provides us with a uniquely powerful computational framework. Why this perspective? We will answer by looking at each of the three parts: categories, programming languages, and logic (which we will interpret from the perspective of reasoning).

1.1 Categories

Of central importance, we argue, to future of scientific knowledge, is in finding a middle ground of knowledge representation that is both legible to humans and machines, with the aim of removing ambiguity from both human and machine interpretation. From [9] we have

the following passage that illustrates our view that category theory can serve as a representation language for both scientific concepts and experimental processes:

> I intend to show that category theory is *incredibly efficient as a language for experimental design patterns*, introducing formality while remaining flexible.....Universal languages for science, such as calculus and differential equations, matrices, or simply graphs and pie charts, already exist, and they grant us a cultural cohesiveness that makes scientific research worthwhile. In this book I attempt to show that category theory can be similarly useful in describing complex scientific understanding.

Need to show olog here:

It has been shown that categories can improve on description logic as a basis for KR [6], however an even greater value will be realized as a foundation for functorial model management [**?**].

Knowledge Sheaves [3]

1.1.1 Scientific Circuits

Importantly, there is another categorical area of considerable potential use: the DiSCoCat framework [2, 5]. Compilation of text into circuits, or diagrams into circuits provides a very interesting representation for machines.

The transition to 'discovery science' (`https://en.wikipedia.org/wiki/Discovery_science`), and what that implies for the usefulness of computational methods (Patterson paper on learning data science programs, etc.)

1.2 Programming Languages and Type Theory

There a few different ways that programming languages, and more specifically those based on Homotopy Type Theory will serve science. First, it's instructive to know a little bit about this particular brand of type theory, and its importance as seen as a foundation of mathematics. Without recounting the full history of developments (see [] for a lengthier description of motivations), we give a brief overview in the form of the recollections of Vovoedsky, a Fields Medalist, and his concerns after finding errors in his published works. Vovoedsky had realized some years after publishing a paper that it had mistakes in it, and that those mistakes had gone unnoticed. Was this an isolated incident? If not, what was to be done? Vovoedsky decided that what was needed was a way to formalize mathematics using computers, so that proofs could be machine checked. What emerged was a reconceptualization of mathematics, based not on set theory (), but rather on an extension of constructive type theory using notions from homotopical algebra. The results of this synthesis, the first glimmers of which appeared in [**?**, **?**, **?**], via the univalence axiom resulted in what we now call Homotopy Type Theory. It is meant to serve as a new, fully computational foundation for mathematics, so that proofs may be easily constructed and checked by computers.

If we are concerned in life sciences with correctness and reproducibility, and if it follows from our reasoning that much of the process of life science research will be automated–and hence controlled by programs specified in programming languages, it then follows that a logical/computational framework like HoTT serves as an ideal foundation for a few reasons:

- 1) *Categories and Higher Categories*-If we wish to formally conceptualize scientific processes within the framework of category theory, as we advocated in the previous section, then HoTT is the ideal computational framework in which to realize them.
- 2) *Proof of Correctness*–One of the most explicit goals directing HoTT's creation was/is formal verification. If we are to specify our scientific theories and processes in a computational language, then HoTT is the ideal.
- 3) *Interpretable* If the front-end user facing diagrammatic language can be translated into a specification in a typed language, then we dramatically improve machine legibility
- 5) *Machine Interpretable* We want a representation that facilitates machine legibility

1.3 Logic/Reasoning

The third part of our trinity tells us that what we have done also has an interpretation in logic. We would like to understand the implications of our representation and framework for the process of reasoning over the representations we accumulate.

Indeed, the most powerful affect of offering a new perspective in publishing is that of the interaction that occurs, and the knowledge it is possible to gain, as we gain distributed representations that are formulated cohesively.

Neural Graph Reasoning, Complex Logical Query Answering Meets Graph Databases [] Neural Symbolic Programming [] Neural Symbolic Programming for Science []

We can conceptualize processes from neuro-symbolic learning, and yet, what is the logic we are using exactly?

Mostly this is a first order logic, but what advantages doo we get with HoTT substituted as the logic? In other words we learn neuro-symbolic representations over instances of homotopical types. This gives us access to higher categories and higher toposes.

Is quantum logic expressable in neuro-symbolic programming? If not, is this advantageous to do so, given recent work in the quantum structure of decision processes.

quote: "A neurosymbolic learning algorithm is a mechanism for program synthesis that uses deep representations and gradient-based optimization as well as symbolic methods such as search and automated deduction."

If we discover or synthesize a program in HoTT, and it is valid, then we have a proof of correctness. In neuro symbolic programming, is it possible to pre generate all of the data? I.E. just run trillions of programs and collect metat data, then use this as the neural network? I.e. using a simulator like in DeepRL, and then learning over this. We would need a scientific process simulator? this is the distillation perspective in neuro symbolic learning (pg. 19)

2 Specificity of Process

3 The maximum scientific contribution

There is a simple question we can ask: *what is the maximum scientific contribution that an act of publishing can make?* We'll outline how formalization can get us there.

- 1) *Peer review*-Peer review is more or less a methods check, can this be automated .
- 2) *Knowledge Representation*–Is publishable knowledge as actionable as possible?
- 3) *Reproducible* For each individual model, we compute and record the TreeSHAP feature importance score for the full set of predictors.
- 4) *Verifiable* This is distinct from reproducible, as the focus is on formal guarantees of correctness.
- 5) *Machine Interpretable* We want a representation that facilitates machine legibility

4 Metadata Encoding/Reasoning

The crucial insight is that we should not expect to figure out how to material reason well over all instances of experimental data, or even all experimental data of a certain type or class. What we can seek to do is specify attributes of metadata, and this in itself may be sufficient. Particularly within the context of automated experimentation we can expect metadata to be critical: time of day, time of sample processing, etc. All of this should be encoded as boolean attributes, which can enable a formal reasoning model based on formal concept analysis.

Important idea: If we have the metadata, can we infer and reconstruct the full experimental process....thus giving us the capability to construct process graphs of the experiments without the input description? Then reason over these?

5 Formally specifying knowledge

Critical to the functioning of the system is the formal specification of a learned concept. What conclusion was learned? And how do we encode this knowledge in a form that is machine readable/checkable? Is it a causal graph? An ontology or olog? Whatever it is, this is the key to the process, along with metadata encoding.

References

[1] Spencer Breiner, Blake Pollard, Eswaran Subrahmanian, *Functorial Model Management*

[2] , *A Compositional Distributional Model of Meaning*, (2008)

[3] Thomas Gebhart, J. Hansen, Paul Schrater, *Knowledge Sheaves: A Sheaf-Theoretic Framework for Knowledge Graph Embedding*, (2021) International Conference on Artificial Intelligence and Statistics

[4] David A. Kronick, *Original Publication: The Substantive Journal". A history of scientific and technical periodicals:the origins and development of the scientific and technological press, 1665-1790*, The Scarecrow Press, New York (1965)

[5] Vincent Wang-Masclanica, Jonathon Liu, Bob Coecke, *Distilling Text into Circuits*, (2023)

[6] Evan Patterson, [**?**], (2017) arXiv:1706.00526

[7] Urs Schreiber, Michael Schulman, John Baez, David Corfield, *computational trilogy*, nCatLab `https://ncatlab.org/nlab/show/computational+trilogy`

[8] Michael Schulman, *Homotopical trinitarinism: A perspective on homotopy type theory*,(2018) https://ncatlab.org/nlab/files/ShulmanHomotopicalTrinitarianism.pdf

[9] David I. Spivak, *Category Theory For The Sciences*, (2013) MIT Press.