Ben Goertzel · Matt Iklé ·
Alexey Potapov · Denis Ponomaryov (Eds.)

LNAI 13539

Artificial General Intelligence

15th International Conference, AGI 2022
Seattle, WA, USA, August 19–22, 2022
Proceedings

Springer

Lecture Notes in Artificial Intelligence 13539

Subseries of Lecture Notes in Computer Science

More information about this subseries at https://link.springer.com/bookseries/1244

Ben Goertzel · Matt Iklé · Alexey Potapov ·
Denis Ponomaryov (Eds.)

Artificial General Intelligence

15th International Conference, AGI 2022
Seattle, WA, USA, August 19–22, 2022
Proceedings

Springer

Editors

Ben Goertzel
SingularityNET Foundation
Amsterdam, The Netherlands

Matt Iklé
SingularityNET Foundation
Amsterdam, The Netherlands

Alexey Potapov
SingularityNET Foundation
Amsterdam, The Netherlands

Denis Ponomaryov
Russian Academy of Sciences
Novosibirsk, Russia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-19906-6 ISBN 978-3-031-19907-3 (eBook)
https://doi.org/10.1007/978-3-031-19907-3

LNCS Sublibrary: SL7 – Artificial Intelligence

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the papers presented at AGI 2022, the 15th Conference on Artificial General Intelligence. The conference took place at The Crocodile in Seattle during August 19–22, 2022, and was broadcast live via Zoom and co-sponsored by SingularityNET's YouTube channel for online participants and viewers.

Spread over four days, the conference included two full-day and two half-day workshops, eight keynote lectures, two fireside chats featuring AGI luminaries, two panel sessions, a code demonstration, eight sessions of in-person and online contributed papers and talks, two poster video sessions, and a full day of general audience sessions with leading AGI researchers and leaders.

Learning from our first attempt at a hybrid conference at AGI 2021 in Palo Alto, California, the AGI 2022 hybrid conference process went smoothly and included additional time for online paper presentations. In addition to the full-day deep dives into the NARS and OpenCog/Hyperon AGI systems that have become somewhat traditional in the AGI conference series, this year's conference also included half-day workshops on Interpretable Natural Language Processing and AGI in FinTech, as well as a full-day demonstration of FutureAI's Brain Simulator II. The theme for this year's general audience day was Ethical Machine Creativity, which culminated in an evening concert with a spectacular performance by the Jam Galaxy band, featuring the Desdemona robot singing AI-generated lyrics (albeit with some human curation).

Researchers from at least 19 countries attended AGI 2022, either in-person or online. As always, conference activities spawned numerous deep discussions on incredibly diverse topics outside of the venue itself.

This volume contains the contributed papers presented at AGI 2022. There were 61 submissions. Each submission was reviewed by at least two (on average 2.54) Program Committee members. The committee decided to accept 31 long papers (51% acceptance) for oral presentation, and an additional 13 submissions for poster presentations. As usual, conference presentations and discussions spanned a wide and ever-evolving array of topics. New this year were discussions on AGI hardware as well as the release of a major open-source software tool, the DR-Learner.

Speakers and panelists for the general audience day included Sophia Robot (AGI Ambassador), Joscha Bach (Principal AI Engineer, Intel Labs), Ben Goertzel (CEO, OpenCog Foundation and SingularityNET and Chair, AGI Society), Janet Adams (COO, SingularityNET), Kyrtin Atreides (COO, AGI Laboratory), Gabriel Axel Montes (Music Co-director, Jam Galaxy), Charles Simon (Founder and CEO, Future AI), Chris Poulin (TrueAGI Advisor), Ed Keller (SingularityNET), Dianne Krouse (Jam Galaxy), Matt Iklé (CAIO, SingularityNET, and Treasurer, AGI Society), and Douglas Miles (Logicmoo and SingularityNET).

Three additional keynote speeches were presented by researchers from both academia and industry. This year's speakers and topics were as follows:

- Nelson Niu (Ph.D. student, University of Washington), "Polynomial Functors: Natural Formal Models of Interaction"
- Rachel St. Clair (Founder and CEO, Simuli), "Resource Management in AGI Systems"
- Chris Poulin and Phil Tabor (Co-leader, DR-Learner), "Open Source Deep Reinforcement Learning: Deep Dive"

Also included this year were two fireside chats

- Ben Goertzel and Rachel St. Clair, "Novel Hardware for Enabling AGI and Machine Creativity"
- Ben Goertzel and Gary Marcus (Founder and CEO, Robust.AI), "Overcoming the Obstacles Between Here and AGI"

Finally, there was one code demo:

- Dzvinka Yarish (Co-author, DR-Learner), "Demo of Open Source DR-Learner Tool"

We end by thanking everyone who made this successful conference event happen. This includes all of our sponsors (SingularityNET for their incredible and hard-working team that helped organize the event and handled the logistics, moderation, and live-streaming of the conference; the AGI Society for organizing the conference series since 2008; Future AI for their generous financial support as a "Main AGI Session" sponsor; TrueAGI for their in-kind support; and Springer for their contribution for the best paper prize as well as their proceedings publishing support). It also includes every one of our Program Committee members for their dedicated service to the review process, our contributors, participants, and tutorial, workshop, and panel session organizers, without whom the conference would not exist.

August 2022

Ben Goertzel
Matt Iklé
Denis Ponomaryov
Alexey Potapov

Organization

General Chair

Ben Goertzel	AGI Society, USA, and SingularityNET, The Netherlands

Program Committee Vice Chairs

Matt Iklé	AGI Society, USA, and SingularityNET, The Netherlands
Alexey Potapov	SingularityNET, The Netherlands
Denis Ponomaryov	Novosibirsk State University, Russia

Organizing Committee

Janet Adams	SingularityNET, The Netherlands
Ibby Benali	SingularityNET, The Netherlands
Ben Goertzel	AGI Society, USA, and SingularityNET, The Netherlands
Lowy Haley	SingularityNET, The Netherlands
Matt Iklé	AGI Society, USA, and SingularityNET, The Netherlands,
Sarah Isakson	SingularityNET, The Netherlands
Lisa Rein	Jam Galaxy, USA
Ryan Sternlicht	Jam Galaxy, USA
Iain Wentworth	SingularityNET, The Netherlands

Program Committee

Cristiano Castelfranchi	Institute of Cognitive Sciences and Technologies, Italy
Antonio Chella	Università di Palermo, Italy
Deborah Duong	SingularityNET, The Netherlands
Aaron Eberhart	Kansas State University, USA
Nil Geisweiller	OpenCog Foundation and SingularityNET, The Netherlands
Ben Goertzel	SingularityNET, The Netherlands
Thomas Gärtner	TU Wien, Austria

Patrick Hammer	Temple University, USA
Jose Hernandez-Orallo	Universitat Politècnica de València, Spain
Marcus Hutter	Australian National University, Australia
Matt Iklé	SingularityNET, The Netherlands
Cliff Joslyn	Pacific Northwest National Laboratory, USA
Garrett Katz	Syracuse University, USA
Anton Kolonin	SingularityDAO, St. Lucia
Xiang Li	Liaoning University of Technology, China
John Licato	Indiana University-Purdue University Fort Wayne, USA
Frank Loebe	University of Leipzig, Germany
Tony Lofthouse	Stockholm University, Sweden
Douglas Miles	SingularityNET, The Netherlands
Till Mossakowski	University of Magdeburg, Germany
Amedeo Napoli	Loria Nancy (CNRS, Inria, and Université de Lorraine), France
Laurent Orseau	Google, USA
Guenther Palm	University of Ulm, Germany
Aleksandr I. Panov	Federal Research Center Computer Science and Control, RAS, Russia
Maxim Peterson	ITMO University, Russia
Denis Ponomaryov	Novosibirsk State University, Russia
Alexey Potapov	SingularityNET, The Netherlands
Rafal Rzepka	Hokkaido University, Japan
Ricardo Sanz	Universidad Politécnica de Madrid, Spain
Oleg Scherbakov	ITMO University, Russia
Ute Schmid	University of Bamberg, Germany
Abdulrahman Semrie	SingularityNET, The Netherlands
Barry Smith	SUNY Buffalo, USA
Bas Steunebrink	NNAISENSE, Switzerland
Kristinn R. Thorisson	Reykjavik University, Iceland
Adam Vandervorst	SingularityNET, The Netherlands
Mario Verdicchio	Università degli Studi di Bergamo, Italy
Pei Wang	Temple University, USA
Jonathan Warrell	SingularityNET, The Netherlands
Robert Wünsche	TU Dresden, Germany
Roman Yampolskiy	University of Louisville, USA
Liudmila Zhilyakova	V.A. Trapeznikov Institute of Control Sciences, RAS, Russia

Additional Reviewers

Catt, Elliot
Eberding, Leonard
Landgrebe, Jobst
Livenstev, Vadim
Penz, David
Sheikhlar, Arash

Contents

Accepted for Poster Presentation

Accepted for Full Oral Presentation

Toward Generating Natural-Language Explanations of Modal-Logic Proofs

Mike Giancola(✉), Selmer Bringsjord, and Naveen Sundar Govindarajulu

Rensselaer AI & Reasoning (RAIR) Lab, Department of Computer Science,
Department of Cognitive Science, Rensselaer Polytechnic Institute (RPI),
Troy, NY, USA
mike.j.giancola@gmail.com, naveen.sundar.g@gmail.com

Abstract. As we work toward artificial general intelligence, it is clear that we must try to imbue agents with faculties which ensure they are trustworthy. We firmly believe that an AGI agent must be able to explain its decision-making in order for it to be considered trustworthy. More specifically, agents must be able to explain themselves in a way that is both logically correct and understandable to humans. We take a first step toward a system that can generate explanations which satisfy this pair of conditions. We created the first model that can produce summaries of modal-logic proofs using a transformer language model. We qualitatively evaluated the model's outputs on a held-out test set and found that the logical content of the model's explanations precisely matched the input proofs in 60% of cases.

Keywords: Hybrid AI · Transformer language models · Modal logic

1 Introduction

As AI agents continue to play a larger role in our everyday lives, the issue of trust of AI systems is becoming more apparent. Moreover, as we work toward artificial general intelligence (AGI), it is clear that we must try to imbue agents with faculties which ensure they are trustworthy. While there is no one sufficient condition for trust of an AGI, we firmly believe that the ability to explain its decision-making is a necessary condition for trust in an AGI. More specifically, agents must be able to explain themselves in a way that is both logically correct and understandable to humans.

Many approaches to explainable AI secure one or the other of these two conditions. DARPA's Explainable AI Program has been focused primarily on the latter. The goal is to produce systems that can explain machine-learning models in a human-understandable way. But since the core technology is machine-learning-based, there is no guarantee that the decisions nor the explanations will be formally correct with respect to some relevant formal system. Alternatively, in much of our prior work we have taken a logic-based approach to AI which enables agents to explain their decisions in a formally correct (and verifiable)

B. Goertzel et al. (Eds.): AGI 2022, LNAI 13539, pp. 220–230, 2023.
https://doi.org/10.1007/978-3-031-19907-3_21

way [3,6,7]. Our AI agents do this by producing a formal proof; unfortunately, proofs are not easily understood by humans who don't have training in formal methods.

In this paper, we take a first step toward a system that can generate explanations which are both logically correct and understandable by humans. Specifically, we created the first model which can produce summaries[1] of modal-logic proofs.

The rest of the paper is as follows. We first introduce the modal logic used (Sect. 2), then the approach we took to generating natural-language explanations of proofs from that logic (Sect. 3). Next, we provide and analyze sample outputs of the system (Sect. 4). Finally, we discuss related work (Sect. 5) and conclude (Sect. 6).

2 Cognitive Calculi

A *cognitive calculus* is a multi-operator intensional logic with modal operators that capture propositional attitudes of human cognition (e.g. **K** for "knows", **B** for "believes"). For the purposes of this paper, a cognitive calculus consists essentially of two components:[2] (1) multi-sorted n^{th}-order logic[3] with intensional/modal operators for modeling cognitive attitudes (e.g. **K**, **B**) and (2) inference schemata that—in the tradition of proof-theoretic semantics [5]—fully express the semantics of the modal operators. We note that the title is slightly inaccurate, as a cognitive calculus is not exactly the same as a modal logic. Specifically, because of this last point: Whereas modal logics all have (typically model-theoretic) semantics, cognitive calculi have no model-based semantics. The meaning of formulae within a cognitive calculus is defined exclusively by the ways they can be used in proofs and arguments, which is accomplished formally by the calculus' inference schemata.

2.1 A Micro Calculus: $\mu\mathcal{C}$

In the present paper, we utilize a micro cognitive calculus we refer to as '$\mu\mathcal{C}$.' We use a micro calculus, as opposed to a full-fledged cognitive calculus,[4] in order

[1] As we discuss in Sect. 5, there are systems that can create *explanations* of modal logic proofs [4], but not *summaries*. That is, they can produce explanations which have a one-to-one correspondence with the input proof, but cannot synthesize a summary that highlights only the major components of the proof.

[2] For a full exposition of exactly what a cognitive calculus is and isn't, we point the interested reader to Appendix A of [3].

[3] Most cognitive calculi subsume first-order logic; some others include also second-, third-, and higher-order logics. For reasons that will be explained later in the paper, the cognitive calculus we utilize herein includes, of extensional logics, only zero-order logic.

[4] Such as the Deontic Cognitive Event Calculus ($\mathcal{DCEC}$) and its inductive counterpart ($\mathcal{IDCEC}$). The interested reader is referred to [3], which utilizes both of these calculi.

to simplify the challenging task at hand of generating explanations of modal-logic proofs. That is to say, the only difference between a standard cognitive calculus and a micro cognitive calculus is the relative size in terms of syntactic forms and inference schemata. We have elsewhere, previously, made use of other micro cognitive calculi, in order to simplify meta-proofs and explanation of our automated-reasoning algorithms; see e.g. [8].

In general, a cognitive calculus consists of two main pieces: a signature and a set of inference schemata. The signature of a cognitive calculus has four components: (1) a set of sorts, (2) a set of function signatures, (3) a grammar for terms, and (4) a grammar for formulae. Note that each of these components builds upon a pre-existing core.[5] The sorts and function signatures build upon the standard, extensional event calculus[6] [9]. While the terms and syntactic forms generally build upon first-order logic, in the case of $\mu\mathcal{C}$, they build upon zero-order logic; that is, propositional logic with predicates and function symbols, but *no quantifiers.*

Signature. The signature contains three sorts: Agent, for specifying human/artificial cognizers within modal formulae; Moment, for specifying time points; and Formula, for specifying any well-formed formula in the calculus. Next, there are three types of terms: variables, constants, and functions. Finally, the syntactic forms cover the standard relations of propositional logic, and modal operators for **P**erception, **B**elief, and **D**esire.

$\mu\mathcal{C}$ Signature

$$S ::= \mathsf{Agent} \mid \mathsf{Moment} \mid \mathsf{Formula}$$

$$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$$

$$\phi ::= \{\neg\phi \mid \phi \land \psi \mid \phi \lor \psi \mid \phi \rightarrow \psi \mid \mathbf{P}(a, t, \phi) \mid \mathbf{B}(a, t, \phi) \mid \mathbf{D}(a, t, \phi)$$

Inference Schemata. The calculus contains four inference schemata in the natural-deduction tradition: I_1 enables an agent to infer a belief in any formula ϕ which it perceives. I_2 enables an agent to perform *And Elimination* on any conjunction it believes holds. In the same way, I_3 and I_4 enable an agent to use *Or Introduction* and *Implication Elimination* within beliefs.

$\mu\mathcal{C}$ Inference Schemata

$$\frac{\mathbf{P}(a, t, \phi)}{\mathbf{B}(a, t, \phi)}\ [I_1] \quad \frac{\mathbf{B}(a, t, \phi \land \psi)}{\mathbf{B}(a, t, \phi)}\ [I_2] \quad \frac{\mathbf{B}(a, t, \phi)}{\mathbf{B}(a, t, \phi \lor \psi)}\ [I_3] \quad \frac{\mathbf{B}(a, t, \phi \rightarrow \psi) \quad \mathbf{B}(a, t, \phi)}{\mathbf{B}(a, t, \psi)}\ [I_4]$$

[5] For brevity, the pre-existing core of the function signatures is excluded as we will not need them for the problems presented herein.

[6] Other calculi (e.g. the *situation calculus*) for modeling commonsense and physical reasoning can be easily switched out in-place of the event calculus.

3 NLG via Transformer Language Models

While we do not assert that transformer language models are unproblematic,[7] their impressive ability to quickly generate reasonable-looking natural-language text is, at the time of this writing, unmatched by any other technology. Hence our model to convert formal proofs into natural-language explanations utilizes a transformer language model.[8]

Specifically, we fine-tuned Pegasus [12] on a dataset of $\mu\mathcal{C}$ proofs and corresponding explanations, and evaluated it on a held-out dataset. Next, we discuss the reasoning behind the choices of Pegasus and $\mu\mathcal{C}$ for this work.

3.1 Pegasus

We selected the Pegasus transformer model as it was designed to perform well at *abstractive summarization*. Briefly, whereas *extractive* summarization simply extracts a proper subset of the input verbatim to synthesize a summary, *abstractive* summarization attempts to create a coherent summary that contains words/phrases that did not appear in the source text. This approach to summarization was necessary for our task, since we did not want to simply pick out pieces of the proofs for the summaries, but rather summarize the key points *in English*. Our task would be more accurately categorized as “summarization and translation.” But since there are no translation models pre-trained on this type of data, we determined that an abstractive summarization model was the best available option.

3.2 $\mu\mathcal{C}$ and the Proof Domain

We selected $\mu\mathcal{C}$ as the cognitive calculus within which proofs would be created for this experiment largely for its simplicity. Whereas some cognitive calculi contain many more complex inference schemata, including e.g. meta-logical statements about provability, $\mu\mathcal{C}$ contains only four inference schemata, each of which can be easily explained in English. For example, I_1 allows an agent a to infer a belief in some formula ϕ which it perceives.[9] This enabled the creation of proofs involving several inferences that could be succinctly summarized in a few sentences.

Similarly, we selected the proof domain—the weather—in order to enable quick, manual generation of proofs which are logically correct and corresponding explanations which are sensible. We included predicates for simple types of

[7] The interested reader is referred to [2] for a thorough analysis of the environmental, financial, and societal concerns surrounding transformers.

[8] We certainly do not believe transformers are the only method by which this generation of proof explanations can be achieved. In fact, we expect that methods of natural-language generation which incorporate symbolic reasoning would almost certainly provide better assurances that the resulting explanations match the logical content of the proof. We discuss this more in Sect. 6.

[9] See Sect. 2.1.

weather (e.g. `Raining`, `Foggy`), road conditions which could be caused by the weather (e.g. `Slippery`, `ReducedVisibility`), and items one may want for certain weather (e.g. `rainboots`, `umbrella`). Example proofs and explanations are given in Sect. 4.

3.3 Model Fine-Tuning

We fine-tuned the Pegasus transformer on 20 proof-explanation pairs, holding out a set of 10 for evaluation. We note that the dataset is relatively small for a transformer training task for several reasons, but primarily because the pairs had to be engineered by hand, which was labor-intensive. The evaluation was also performed by hand, which will be discussed further in the following section.

Fortunately, because transformers are pre-trained on large datasets, we expected that we could be successful fine-tuning with a relatively small dataset. For details on the implementation of the fine-tuning process, see Appendix A.

4 Evaluation

We took a qualitative approach to evaluating the results of our fine-tuned model. Statistical metrics for measuring the similarity of the model's output to the ground truth aren't very meaningful in this case, as they fail to capture whether the logical reasoning content of the outputs are similar. For example, "You should bring an umbrella today because it is raining" and "You should bring an umbrella today because it is not raining" are very close syntactically, but the latter sentence doesn't exhibit valid reasoning (assuming common-sense understanding of the involved properties). Hence we evaluated the output qualitatively by partitioning it into the following trichotomy:

1. The output *logically corresponds* to the input proof.
2. The output doesn't logically correspond, but is still *logically correct.*
3. The output *neither* logically corresponds to the input proof nor is logically correct.

By "logical correspondence," we mean that the logical content of the summary is consistent with that in the proof. Since our goal is to generate summaries (as opposed to complete explanations), the model's output need not explain each individual step of the proof. However, it may not include language that does not have a formal correlate in the proof.

By "logical correctness," we mean that each step within the summary corresponds to an inference schema (or chain of inference schemata) in the cognitive calculus. Again, since our model is summarizing proofs, the explanation need not include a natural-language correlate of each step in the proof, but it cannot include language that is not sanctioned by some subset of the inference schemata.

The present page limit precludes inclusion of the full set of output on the test set. Here we show only a few noteworthy results, both positive and negative.[10]

[10] Newlines and indentation have been added for readability.

4.1 Example #1

```
Input:
PROOF: (Believes! a now (Slippery now road))
  VIA MODUS PONENS IN BELIEF:
   (Believes! a now (implies (or (Raining now) (Snowing now))
                             (Slippery now road))),
   (Believes! a now (or (Raining now) (Snowing now)))
     VIA GIVEN:
      (Believes! a now (implies (or (Raining now) (Snowing now))
                                (Slippery now road)))
     AND:
      (Believes! a now (or (Raining now) (Snowing now)))
        VIA DISJUNCTION INTRO IN BELIEF:
         (Believes! a now (or (Raining now) (Snowing now)))
           VIA GIVEN: (Believes! a now (Raining now))

Ground Truth (Human-Generated):
The agent believes the road is slippery. This is because the agent
believes if it is raining or snowing then the road is slippery, and
the agent believes it is raining.

Model Output:
The agent believes it is slippery on the road. This is because the agent
believes it is raining or snowing on the road.
```

We consider the model's output to be excellent for two reasons: (1) it logically corresponds to the input proof; and (2) it is more concise than the human-generated/ground-truth explanation, without excluding any content necessary for a person to understand it.

4.2 Example #2

```
Input:
PROOF: (Believes! a now (Slippery now road))
  VIA MODUS PONENS IN BELIEF:
   (Believes! a now (implies (or (Raining now) (Snowing now))
                             (Slippery now road))),
   (Believes! a now (or (Raining now) (Snowing now)))
     VIA GIVEN:
      (Believes! a now (implies (or (Raining now) (Snowing now))
                                (Slippery now road)))
     AND:
      (Believes! a now (or (Raining now) (Snowing now)))
        VIA DISJUNCTION INTRO IN BELIEF:
         (Believes! a now (Snowing now))
           VIA GIVEN: (Believes! a now (Snowing now))

Ground Truth (Human-Generated):
The agent believes the road is slippery. This is because the agent
```

```
believes if it is raining or snowing then the road is slippery, and
the agent believes it is snowing.

Model Output:
The agent believes it is slippery on the road. This is because the agent
believes if it is raining or snowing then it is slippery on the road, and
the agent believes it is raining or snowing.
```

In this case the machine's output is slightly longer than the ground truth but still logically corresponds to the input proof.

4.3 Example #3

```
Input:
PROOF: (Desires! a now (or (Umbrella now) (RainJacket now)))
  VIA MODUS PONENS IN BELIEF:
   (Believes! a now (implies (and (Raining now) (not (Windy now)))
                             (Desires! a now (or (Umbrella now)
                                                 (RainJacket now))))),
   (Believes! a now (and (Raining now) (not (Windy now))))
     VIA GIVEN:
      (Believes! a now (implies (and (Raining now) (not (Windy now)))
                                (Desires! a now (or (Umbrella now)
                                                    (RainJacket now)))))
     AND:
      (Believes! a now (and (Raining now) (not (Windy now))))
        VIA CONJUNCTION INTRO IN BELIEF:
         (Believes! a now (Raining now)),
         (Believes! a now (not (Windy now)))
           VIA GIVEN: (Believes! a now (Raining now))
           AND GIVEN: (Believes! a now (not (Windy now)))

Ground Truth (Human-Generated):
The agent desires an umbrella or a rainjacket. This is because the agent
believes if it is raining and not windy then it should hold this desire,
and the agent believes it is rainy and not windy.

Model Output:
The agent desires umbrellas and rain jackets. This is because the agent
believes if it is raining and not windy then it should hold this desire,
and the agent believes it is raining and not windy.
```

The machine's output is almost identical to the ground truth, but there is one critical difference: it should be "umbrellas *or* rain jackets", not "umbrellas *and* rain jackets". Hence while this output looks reasonable, it does not logically correspond to the input proof nor is it logically correct.

4.4 Example #4

```
Input:
PROOF: (Believes! a now (Rainbow now))
  VIA MODUS PONENS IN BELIEF:
   (Believes! a now (implies (and (Raining before) (Sunny now))
                             (Rainbow now))),
   (Believes! a now (and (Raining before) (Sunny now)))
     VIA GIVEN:
      (Believes! a now (implies (and (Raining before) (Sunny now))
                                (Rainbow now)))
     AND:
      (Believes! a now (and (Raining before) (Sunny now)))
        VIA CONJUNCTION INTRO IN BELIEF:
         (Believes! a now (Raining before)),
         (Believes! a now (Sunny now))
           VIA GIVEN: (Believes! a now (Raining before))
           AND GIVEN: (Believes! a now (Sunny now)))
```

```
Ground Truth (Human-Generated):
The agent believes there is a rainbow now. This is because the agent
believes if it rains first and is sunny second then there will be a
rainbow, and the agent believes the premise.
```

```
Model Output:
The agent believes it is sunny and raining. This is because the agent
believes if it is sunny and raining then it should hold this belief,
and the agent believes it is sunny and raining.
```

Here the machine's output is logically correct (albeit tautologically), but does not at all logically correspond to the input proof.

4.5 Overall

Out of the 10 held-out test examples, six logically corresponded to the input proof (60%), two did not logically correspond to the input but were logically correct (20%), and two neither logically corresponded nor were logically correct (20%). While we do not deny that there is plenty of room for improvement, our model is the first (to the authors' knowledge) able to generate summaries of modal-logic proofs. Next, we discuss relevant prior work in this space.

5 Related Work

To the author's knowledge, there is no prior work using transformer language models to generate explanations of proofs in *any* logic, let alone a logic as highly-expressive as $\mu\mathcal{C}$. There is, however, some prior work on generating explanations of proofs using other methods.

Felty and Hager [4] presented a method for generating natural-language explanations of modal-logic proofs. They essentially hard-code natural-language templates for every inference rule in their logic. Thus the technique cannot be generalized to new inference rules or logics without hard-coding new templates. Additionally, this creates a one-to-one correspondence between the proof and explanation. While this may be desired in some cases, this method is incapable of generating *summaries* of proofs (= explanations that leave out minor details in an effort to demonstrate "big picture" understanding).

Alexoudi et al. [1] developed a method for producing summaries of mathematical proofs. It used a submodule to extract only the mathematically "interesting" proof steps in order to create a higher-quality summary. However, again, the natural-language translation boils down to a hard-coded transformation. For example, the term "`primitive_ind`" is translated to the phrase "one-step structural induction on" [1]. Also, as the focus in this work was on generating summaries of simple mathematical proofs,[11] they use standard first-order logic, and hence their method doesn't address generating summaries of proofs which contain modal operators.

While our use of transformers introduces the possibility that the resulting explanations may not precisely logically correspond to the input proof, the linguistic content is of much higher quality than that seen in prior work. Alexoudi et al. specifically mention this drawback in their work, noting that "In certain cases the template mapping produces minor grammatical errors" [1]. We note that all of the explanations generated by our model were grammatically correct. Of course, the ultimate goal of our research is a model which guarantees logical correspondence *and* grammatical correctness. We discuss future work in this direction in the following section.

6 Conclusion

We firmly believe that for AGI agents to be considered trustworthy by most people, these agents will need the capability to explain their decision-making in a way that is both logically correct and understandable by humans. In this paper, we have taken a first step in that direction. We created the first model which can generate natural-language summaries of modal-logic proofs. Of the summaries generated from proofs in the test set, 60% logically corresponded to the input proof, and all summaries were grammatically correct and overall linguistically coherent. Nonetheless, clearly there are many needed, subsequent steps; we mention two general directions now.

[11] They state that their goal was to produce summaries of proofs similar to what would be seen in "mathematical textbooks". While they didn't specify any subfields of mathematics, nor the level of rigor (e.g. high-school, undergraduate, or graduate textbooks) they intended their method for, the examples in the paper all involve mathematical-induction proofs of arithmetic properties, e.g. commutativity of addition. It is now known that some classical mathematics beyond merely the textbooks involves third-order logic, and proofs couched therein.

First, our model was trained and tested on a single proof domain, with relatively simple proofs, within a relatively simple cognitive calculus. An AGI agent should be able to generate explanations of proofs in a wide variety of domains which it may not have encountered before. New methods may be needed to achieve this, as well as to enable such an agent to generate explanations of more complex proofs in cognitive calculi with more and deeper modalities and inference schemata.

Second, while using a transformer enabled our model to generate text that was linguistically coherent, one significant drawback is that there is no guarantee the logical content of the explanation matches that of the proof. That is, the explanation may be syntactically correct English, but not match the meaning of the proof.[12] We see two possible directions in this space. First, one could imbue the transformer with some type of rule-based system that ensures that the text it produces corresponds with the logical content of the proof. Second, one could take a different approach to language generation entirely. Specifically, a knowledge-based approach to language generation (e.g. [10,11]) could ensure that outputs are both logically and linguistically sensible.

Acknowledgements. This research is partially enabled by support from ONR and AFOSR (Award # FA9550-17-1-0191).

A Fine-Tuning and Evaluation Implementation

We used the Hugging Face Transformers library to access Pegasus, fine-tune the model, and evaluate it. Our fork of the model (https://github.com/mjgiancola/transformers) includes scripts for fine-tuning with the parameters we set to enable reproduction of our results (https://github.com/mjgiancola/transformers/tree/main/examples/research_projects/proof-nlg).

The script `fine_tune_pegasus.sh` runs the fine-tuning process. It is configured to generate the fine-tuned model's predictions on the test set after fine-tuning is completed. Additionally, the script `get_predictions_from_fine_tuned.py` loads the fine-tuned model and outputs pretty-printed results, including the input (a proof), the ground-truth output (a human-generated explanation), and the model's output.

[12] We note that, while this is a significant drawback, which we shortly address as pressing future work, AI agents which utilize the type of technology presented herein (i.e. a cognitive calculus for reasoning and a transformer for NLG) would still enact logically correct decision-making, even if its explanation wasn't correct. That is, the agent's behavior would still be bound by what it could prove via the calculus' inference schemata.

References

1. Alexoudi, M., Zinn, C., Bundy, A.: English summaries of mathematical proofs. In: Second International Joint Conference on Automated Reasoning-Workshop on Computer-Supported Mathematical Theory Development, pp. 49–60. Citeseer (2004)
2. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 610–623 (2021)
3. Bringsjord, S., Govindarajulu, N.S., Giancola, M.: Automated argument adjudication to solve ethical problems in multi-agent environments. Paladyn J. Behav. Robot. **12**, 310–335 (2021). https://doi.org/10.1515/pjbr-2021-0009
4. Felty, A., Hager, G.: Explaining modal logic proofs. In: Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, pp. 177–180. IEEE (1988)
5. Francez, N.: Proof-Theoretic Semantics. College Publications, London (2015)
6. Giancola, M., Bringsjord, S., Govindarajulu, N.S.: Novel intensional defeasible reasoning for AI: is it cognitively adequate? In: Proceedings of the IJCAI Workshop on "Cognitive Aspects of Knowledge Representation" (CAKR 2022). CEUR-WS (2022). http://ceur-ws.org/Vol-3251/paper9.pdf
7. Giancola, M., Bringsjord, S., Govindarajulu, N.S., Varela, C.: Making maximally ethical decisions via cognitive likelihood & formal planning. In: Ferreira, M.I.A., Tokhi, O. (eds.) Towards Trustworthy Artificial Intelligent Systems, Intelligent Systems, Control and Automation: Science and Engineering, vol. 102. Springer (2022). https://link.springer.com/chapter/10.1007/978-3-031-09823-9_10
8. Govindarajulu, N.S., Bringsjord , S., Peveler, M.: On quantified modal theorem proving for modeling ethics. Electron. Proc. Theor. Comput. Sci. **311**, 43–49 (2019)
9. Kowalski, R., Sergot, M.: A logic-based calculus of events. N. Gener. Comput. **4**(1), 67–95 (1986)
10. Leon, I.E.: OntoGen: a knowledge-based approach to natural language generation. Master's thesis, Rensselaer Polytechnic Institute (2020)
11. McShane, M., Leon, I.: Language generation for broad-coverage, explainable cognitive systems. arXiv preprint arXiv:2201.10422 (2022)
12. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)