NX-RESEARCH

White Paper

Introduction to Narrative Computation

A framework for designing AI systems that are accurate, robust, and explainable

by
Femi Fadeyi
Founder
NX-RESEARCH INC.

2023-Jul-08

EXECUTIVE SUMMARY

Narrative computation is a framework for designing artificial intelligence (AI) systems that leverages descriptions of the computational states and processes of the system to ensure that the system is accurate, robust and explainable. The purpose of this white paper is to provide an introduction to narrative computation concepts, functions, features and limitations; as well as to provide a brief overview of the research being conducted by NX-RESEARCH into developing and deploying narrative computation systems (NCS).

Narrative objects allow writing 'programs' within a narrative computation system such that narrative programs can leverage accurate behavioral descriptions to control the behavior of the overall system itself, in a reliable and robust way.

For example, narrative computation allows writing a 'program' of the following form:

- (pseudocode) <if> AgentSelf 'IS CONFUSED ABOUT REQUEST' <trigger> 'CONCEIVE PLAN TO RESOLVE CONFUSION' <then> 'EXECUTE PLAN';
- The structure of NCS's enables defining contextually-reasonable and specific measures of 'CONFUSION', as well as contextually-reasonable evaluations of 'CONCEIVE PLAN TO RESOLVE CONFUSION' and 'EXECUTE PLAN';
- The incorporation of such contextually-reasonable semantics into the underlying system itself are what make it possible to 'program' NCS's such that they can robustly evaluate themselves and exhibit reasonable-behaviors with consideration of prior goals, current circumstances, novel observations, and novel hypotheses.

At NX-RESEARCH, we have implemented aspects of the narrative computation framework into our NX™ systems, which include NX-Eval™, Curious-NX™ and others. Our current research objectives are to continue development of core features and utility evaluation systems, while we use our Curious-NX™ agent platform to test more complex narrative programs and to evaluate how our architecture behaves when provided with larger datasets and under multi-modal conditions. Our hope is that NCS's will demonstrate scalable efficiency and practical performance, while retaining the defining characteristics of our NX-Eval™ systems.

This white paper will provide an overview of key concepts, functions, features and limitations of the narrative computation framework, contrast narrative computation with other AI approaches, and briefly highlight some of NX-RESEARCH's systems that implement aspects of the narrative computation framework. With continued research, we hope to be able to demonstrate that narrative computation technology will add significant value for mission-critical AI applications that demand accuracy, robustness, and explainability.

1. Overview

1.1. Background and Motivation

1.1.1. There is increasing demand for deploying AI systems for many use cases and applications;

1.1.2. However, certain mission-critical applications, such as engineering, healthcare, autonomous driving, and many others, require a high degree of accuracy, robustness and explainability that is a challenge for AI systems;

1.1.3. Certain implementations of deep neural networks, generative-AI, and related approaches show significant promise for many AI applications, especially for text, image, and audio generation; the quality that is currently available indicates that these tools will soon become indispensable in many such applications;

1.1.4. However, for some applications, accuracy, robustness and explainability remain difficult challenges for even the most capable state-of-the-art systems like GPT-4, PaLM-2, SAM, BERT, etc;

1.1.5. Separately, there are some other approaches to AI that advocate for more 'symbolic' AI systems which have different operating principles from neural-network approaches (like generative-AI);

1.1.6. Good-Old-Fashioned-AI approaches (symbolic AI), such as expert systems, rule-based systems, and others, were originally promising but ran into issues, including scalability, computational complexity, and economic viability; which eventually resulted in profound setbacks for the field of AI research as a whole;

1.1.7. Despite setbacks, symbolic-AI is survived by an array of modern approaches such as *A Neuro-vector-symbolic Architecture for Solving Raven's Progressive Matrices (M. Hersche et al.);*

1.1.8. Narrative computation is another form of symbolic-AI that is based on developing systems of discrete rules but hopes to avoid some of the fatal limitations of traditional symbolic approaches - specifically scalability and computational complexity;

1.1.9. At NX-RESEARCH, we are investigating if narrative computation is a viable framework for improving the accuracy, robustness and explainability of AI systems while remaining efficient and performant;

1.1.10. NX-RESEARCH is a research startup focused on developing AI systems that are accurate, efficient and explainable; our vision is a smarter world where more humans have greater access to enjoy engaging, fulfilling and rich experiences, empowered by responsible and explainable AI systems.

1.2. Objectives of this White Paper

1.2.1. A key objective of this white paper is to provide an introduction to the conceptual underpinnings of narrative computation, situate the framework among other approaches to AI, and inform the opinions of the reader about the narrative computation framework, including potential possibilities and limitations of the approach.

1.2.2. This white paper also aims to inform the reader about some of the innovative narrative computation technologies that are under active research and development by NX-RESEARCH; specifically NX-Eval™ and Curious-NX™.

1.3. Target Audience

1.3.1. This white paper is written to be approachable for a general audience, but it may be helpful to have some background understanding of programming concepts. This white paper is especially tailored to inform an audience who have an interest in gaining an introductory understanding of how narrative computation proposes to work, especially in contrast to other approaches to AI.

1.4. Scope

1.4.1. This white paper will focus on explaining the key concepts and elements of narrative computation, specifically how it relates to and differs from other approaches to AI systems. This white paper will also highlight some of the ongoing research within NX-RESEARCH and discuss some of our systems and architectures;

1.4.2. Importantly, this white paper will NOT dive into the details of implementing NCS's or algorithms; and it will also NOT provide functional details, performance assessments or comparisons for NX-RESEARCH systems or tools;

1.4.3. And though this white paper acknowledges ethical and social implications associated with AI systems, it does NOT address the ethical considerations for AI research in general, nor for narrative computation in particular.

2. Key Concepts and Definitions

2.1. Narrative Computation Framework

2.1.1. The narrative computation framework can be explained as: utilizing narrative objects and narrative programs to create systems that can usefully describe and evaluate themselves;

2.1.2. Designing the underlying set of narrative objects within a narrative computing system is an art that requires full view of the overall objectives of the system, a strategy for organizing symbols, thoughtful planning, careful experimentation and evaluation, and judicious allocation of design effort;

2.1.3. Writing useful and robust narrative programs is also an art of itself;

2.1.4. The central observation of narrative computation is that systems implemented with useful and robust narrative objects and programs, can be designed to evaluate themselves usefully and robustly.

2.2. Narrative Objects

2.2.1. Narrative objects are semantically-grounded computational-state descriptions that are conceptually similar to causal models;

2.2.2. Semantically-grounded refers to the notion that the computational descriptions use 'symbols' which have specific meanings within the system that are consistent with our usual understanding of those symbols outside of the system;

2.2.3. Example would be that for a 'symbol' with a name 'CHECK_IF_SEMANTIC_OBJECT_UNDERSTOOD', the particular routing outcome of following such a 'symbol' should match the expectation described by its label in a consistent and robust way;

2.2.4. Specifically, consistent implies that the behaviors enforced by following the symbol should have the some consistent expected effect in all circumstances where it is invoked (and importantly it should be explicit in not allowing invocation in ambiguous or inappropriate circumstances);

2.2.5. And, robust implies that this design is thoroughly verified across a wide range of operational circumstances and practical edge cases that may occur even with low frequency;

2.2.6. And computational-state refers to the notion that the description (set of symbols) represents particular aspect or sequence of some computation ('computational state');

2.2.7. Example: a 'computation' which followed some sequential set of 2 'symbols' must have 'checked reasonableness of questions' then 'decided question was worthwhile', in that specific order;

2.2.8. Such a description could both (a) specify a pattern for behavior, e.g. for composing some plan for behavior, another e.g. for a designer to 'program' some specific behavioral sequence in advance;

2.2.9. Such a description (b) could also describe some observed behavior, e.g. for circumstances where complex set of influences drove some outcome behavior over some time period, the occupancy of particular states within that period is a specific and particular trace of which behaviors were executed;

2.2.10. If the 'symbols' utilized are generic (ex: XNOR), they will contain low descriptive value in comparison to more situationally-specific symbols (ex: CURIOSITY_WAS_SATISFIED_POSITIVELY).

2.3. Narrative Programs

2.3.1. Narrative programming is the process of writing 'programs' using some defined set of narrative objects / 'symbols':

2.3.2. A well-designed set of 'symbols' enables us to 'program' systems that can exhibit useful real world behaviors, such as systems that robustly evaluate themselves inclusive of their observations, hypotheses, and prior goals;

2.3.3. But importantly, components of narrative programs are actually implemented as particular 'symbols' themselves as conditional routes to other symbols and operations involving the memory system;

2.3.4. Thus, narrative programming consists of describing conditional branches among symbols and organizing sets of conditional branches that describe some particular patterns of behavior that are meaningful and useful both within and outside of the system;

2.3.5. An example of a 'program' might be `(pseudocode) if ('SEMANTIC_UNCLEAR') trigger ('_QUERY_TRUSTED_SOURCE') <then> trigger ('CHECK_SEMANTIC_CLARITY');`

2.3.6. Another example might be `(pseudocode) if ('ENERGY_ACCOUNT_LOW') trigger ('REDUCE_EXPLORATION') <and> ('INCREASE_EXPLOITATION');` external to the system (at multiple levels of abstraction, over multiple epochs, and for multiple scopes of concern).

3. Functional Architecture

3.1. Functional Overview

3.1.1. The functional architecture for an NCS specifies the nature and function of the core narrative programs within an NCS, and how these core narrative programs interact;

3.1.2. The specification of the functional architecture, inclusive of the selection and design of the core programs, is tailored specific to each application and significantly affects the performance characteristics of the NCS;

3.1.3. For a typical AI agent application, the architecture specification and core programs may focus on, for example, hypothesization, simulation, narration, attention, decision-making and belief-updating, etc;

3.1.4. These focus areas may be implemented as various combinations of narrative programs, and as part of the underlying controller, memory system, and interfaces of the system;

3.1.5. The next paragraphs will discuss these focus areas briefly to highlight how they fit with the other focus areas discussed here, as part of an NCS.

3.2. Hypothesization

3.2.1. In narrative computation, hypothesization is the process of generating novel narrative objects and programs;

3.2.2. Owing to the composable nature of narrative objects and hypotheses, generating novel objects and programs is relatively trivial;

3.2.3. Importantly, this does not imply that generated objects or programs are useful in any way, unique in usefulness, nor specifically relevant to any particular set of other concerns;

3.2.4. Hypothesization is merely the the process of composing novel objects within the system, but where the composed object is traceable to the motivating circumstances, objects and programs;

3.2.5. Can be used for ideation, explanation, contextual understanding, situational reasoning, etc

3.2.6. The specific mechanisms for generating hypotheses may be various in nature, and for varying concerns and strategies.

3.3. Simulation

3.3.1. Simulation is an essential aspect of planning and plan evaluation within an NCS;

3.3.2. Simulation involves forward-projection of likely subsequent states of the world given some initial state and a world model;

3.3.3. The objective of simulation is to identify potential decision points / nexi of uncertainty, then following branches forward;

3.3.4. The knowledge that is used to make assessments and predictions are encoded via combinations of narrative objects and programs as part of the world model;

3.3.5. The process of simulation is combinatorial in nature, and mandates periodic assessment of the simulation processes and resource allocations to reflect domains of concern in line with the contemporaneous priorities of the system;

3.3.6. Uncertainties within the system's beliefs and world model are what cause uncertain (conditional) branches within the simulations.

3.4. Narration

3.4.1. Narration is the process of using narrative objects to describe the sequences of observations of the world including the systems' own expectations and behaviors;

3.4.2. Narrations are a form of causal model that make it possible to assess the consistency and reasonableness of observations, expectations, and behaviors;

3.4.3. Importantly, violations of expectations are indicative that the system's beliefs and world model may need to be updated;

3.4.4. Also important, consistency with the behavioral parameters and areas of concern in the system design can be used to determine when specific sets of behaviors may be problematic;

3.4.5. Narrations include (a) a description of the current state of the system itself at different levels of abstraction and time scales, (b) enumerations of candidate deliberations and actions that may fulfill goals, and (c) concerns that might significantly affect how the system evolves.

3.5. Attention

3.5.1. In narrative computation, resource allocation and attention are an inherent and continuously important aspect of useful simulation and narration;

3.5.2. Computing simulations and narrations exhaustively can be resource-intensive and provides no guarantees of concluding with definitive, useful results in any particular time frames;

3.5.3. As such, it is a natural and essential aspect of the narrative computation framework to implement mechanisms that ensure alignment of resource allocation with contemporaneous circumstances and general preferences;

3.5.4. An important aspect of resource allocation is to avoid recursive traps and to provide guarantees and expectations of stability in certain contexts;

3.5.5. Another important aspect of resource allocation is to support shifting attention to handle interrupts and emergencies;

3.5.6. Another notable aspect of resource allocation is to highlight which particular details to prioritize and consider for some particular circumstance, inclusive of which hypotheses or observations may be acceptable to delete (forget).

3.6. Decision-Making

3.6.1. In narrative computation, decision-making is essential in the progress of the computation, at a high level, for the shifting attentional objects and priorities;

3.6.2. Specifically, in determining if some particular task is sufficiently well-defined or explored, narrative computation framework suggests the implementation of specific mechanisms to assess ‘completeness’ of the current task, and mechanisms to 'decide' when to move on to a next ‘task’ or behavioral phase;

3.6.3. Further, the prioritization and selection of which task set to pursue next is handled in a similar way by the attention systems;

3.6.4. The decision-making for completeness and priority enforces consistency with general / prior preferences as well as various objectives, as well as other contemporaneous details of circumstances.

3.7. Belief-updating

3.7.1. In narrative computation, it is possible to update the definitions of objects and programs

3.7.2. Generally, such modifications must be controlled to prevent improper changes in behaviors

3.7.3. But, if belief updates are controlled properly, a narrative computation system can exhibit the ability to change its behaviors to adapt to changing circumstances in ways that are reasonable; otherwise stated: this is how learning works in narrative computation;

3.7.4. An important aspect of ensuring that belief updates are reasonable is to establish a well-designed cost function (utility-evaluation system) that aligns with the specified objectives of the system; inclusive of handling external feedback, internal feedback, and resolving dissonance;

3.7.5. Given that belief updates involve behavioral parameters and structures of ‘programs’, and that the interplay of these factors can evolve in complex ways, the design of the reward, prioritization and decision-making systems is not trivial: This often requires experimentation in the design process and robust performance evaluation systems.

4. Features and Benefits

4.1. Accuracy

4.1.1. Accuracy refers to the tendency for an AI system to not provide incorrect information, inclusive of their priority and context;

4.1.2. Accuracy is important for critical applications where wrong answers may have serious consequences, like engineering, healthcare, autonomous driving, and many more;

4.1.3. Due to the rules-based design of NCS's, they are able to produce precise solutions that involve complex interactions of the mechanisms of a problem domain;

4.1.4. Narrative computation is different from generative-AI approaches which rely on judgements that are fundamentally based on token prediction from the training data and sometimes require external systems to factually-ground assessments;

4.1.5. Instead, NCS's can encode facts and rules, and 'reason' directly from these facts and rules such that they can evaluate the reasonableness of candidate solutions themselves;

4.1.6. And importantly, NCS's don’t have the risks with confabulation in the same ways that generative-AI systems might.

4.2. Robustness

4.2.1. Robustness refers to the tendency of an AI system to make best-effort attempts at solving problems, especially in new or unexpected circumstances;

4.2.2. Robustness is important because the world changes in complex ways that are often difficult to plan for in advance, meanwhile many AI applications require systems that can handle difficult edge cases in ways that are safe and reasonable;

4.2.3. NCS's are able to encode high level preferences and rules that they can utilize as general principles when approaching new or unexpected circumstances;

4.2.4. Well-founded designs of these priors within an NCS can enable them to handle many circumstances, such that in nearly all cases where a safe and reasonable solution cannot be found, NCS can fail gracefully by executing appropriate safety protocols;

4.2.5. The way that NCS robustness is designed is different from how generative-AI systems handle difficult edge cases, which often involves some moderation system with the same underlying technology that attempts to describe behaviors and evaluate alignment with some prior rule sets;

4.2.6. The generative-AI approach can result in systems which are extremely limited when approaching edge cases or that may be quite brittle in trying to enforce alignment under new circumstances;

4.2.7. NCS's implement controls that can be tailored for applications with much more nuance and flexibility even when approaching complex edge cases;

4.2.8. In some circumstances, robustness may be comparable with common sense in the common parlance.

4.3. Explainability

4.3.1. Explainability refers to the tendency for an AI system's behaviors to be usefully-explained in human terms; i.e. not a black box;

4.3.2. Explainability is important because understanding how AI systems behave is critical for fixing problems, for improving performance and for increasing confidence in the design of the systems;

4.3.3. NCS's are designed such that they are constantly narrating their motivations, observations, expectations, and behaviors;

4.3.4. This design makes it relatively straightforward, though not trivial, for NCS to produce useful summarization and explanations of their behaviors;

4.3.5. Importantly, this includes tracing back explicitly the circumstantial motivations, sources of information and sequences of decisions;

4.3.6. The way that narration is a fundamental aspect of NCS is quite different from the way that generative-AI systems are typically designed to speculate about how sequences of behaviors may connect in reasonable ways;

4.3.7. And generative-AI systems must either trace back to external factual sources or they must speculate as to where the information came from within the black box of the model;

4.3.8. Conversely, NCS's provide direct and reliable explanations that allow greater confidence in the behaviors of such systems.

4.4. Multimodality

4.4.1. Multimodality refers to the capacity of AI systems to handle multiple categories of information, like images, text, audio, etc;

4.4.2. Multimodality is important because it allows AI systems to interact with real world systems which often involve vision, sound and other senses;

4.4.3. NCS's are able to natively handle information of essentially any modality;

4.4.4. As long as the proper interfaces are implemented within the system, the underlying architecture of NCS will be able to combine these different modalities into one common sensory experience;

4.4.5. While this is a native feature of NCS's, it is instead an ongoing research area for generative-AI systems that so far indicates generative AI systems benefit from multimodality, though sometimes at a cost of greater complexity;

4.4.6. Separately however, NCS's allow implementing natively-multimodal systems that can thus interact with real world systems, which will be a useful feature for many applications, like engineering, healthcare, autonomous driving, and many more.

5. Challenges and Limitations

5.1. Designing Systems of Rules is Resource-Intensive

5.1.1. One of the central challenges of symbolic-AI systems is the need to develop complex systems of rules for essentially all circumstances;

5.1.2. The scale of this challenge is such that it has hampered deployment of symbolic approaches for many AI applications for decades;

5.1.3. Narrative computation resolves a central concern in this area by allowing the use of narrative objects and narrative programs;

5.1.4. However, NCS's still require designing a complex system of these narrative objects and programs, including evaluating, and optimizing them for particular performance characteristics;

5.1.5. And, it is noteworthy that this design process can be quite resource-intensive.

5.2. Reward and Cost Functions

5.2.1. The design of the reward / cost systems (utility evaluation) in an NCS are critical for the systems to behave in reasonable ways, especially when this relates to how the systems update beliefs and handle unexpected circumstances;

5.2.2. Narrative computation requires thoughtful design of these utility-evaluation systems from the ground up;

5.2.3. While it may be feasible to implement differentiable functions on large datasets, the distillation of these into useful modules within an NCS is a non-trivial engineering process; which can be quite resource intensive.

5.3. Scalability

5.3.1. Given the significant resource costs for programming and designing in narrative computation, this introduces a further challenge in scaling these design efforts up to large system sizes;

5.3.2. Specifically, complex applications may require a significantly large NCS at a scale that may prove challenging, especially to implement, manage and ensure deployment of effective and useful systems;

5.3.3. It remains to be seen if an NCS can be deployed at a scale which can deliver performance that competes with the state of the art generative-AI approaches that are available today.

6. Contrasting with Other Approaches

6.1. Expert Systems and Rule-Based Systems

6.1.1. Expert systems and rule-based systems are symbolic-AI approaches to create systems that reason similarly to humans, using systems of rules that in some cases rely on input from domain experts;

6.1.2. Such systems received significant attention and optimism initially, but later fell out of prominence for various reasons including the scalability, computational complexity, and a failure to deliver on the original optimism;

6.1.3. Narrative computation shares scalability concerns similar to expert systems and rule-based systems, but importantly narrative computation does not seek to manually develop domain-specific knowledge in the same way as expert systems or rule-based systems, and instead the focus is on engineering behavioral robustness - which is challenging, but is a different kind of challenge;

6.1.4. Narrative computation also shares some computational complexity concerns similar to expert systems and rule-based systems, given that inference on a combinatorially-large system of rules is fundamentally limited (these are well-established limits in computer science); but narrative computation proposes resource allocation systems that ensure useful behaviors can be deployed in a way that reflects behavioral characteristics defined by the design of the system;

6.1.5. As such, while narrative computation is similar to the symbolic-AI approaches of expert systems and rule-based systems, the focus of the system design as well as the particular implementation of resource-allocation systems are important differentiators between these symbolic AI approaches.

6.2. Hypothesis Search and Inductive Logic Programming

6.2.1. Hypothesis search approaches (like decision-tree learning, version-space learning and inductive logic programming) are symbolic-AI approaches that implement systems of hypotheses and relations as knowledge bases, and then attempt to identify propositions that are consistent with the knowledge base in some specific ways;

6.2.2. These approaches have the benefit that given some properly-defined set of input constraints, they are able to produce exact solutions to problems and detect some potential problems within the input constraints, such as being insufficiently-specified;

6.2.3. The limitations of these approaches is that developing the systems of constraints can be incredibly resource-intensive, and that searching for exact solutions results in concerns regarding computational complexity;

6.2.4. There are some contemporary approaches to deal with these limitations such as combining neural network approaches with symbolic approaches, such as *A Neuro-vector-symbolic Architecture for Solving Raven's Progressive Matrices by M. Hersche et al.* (NVSA) which show significant improvements on the Raven's progressive matrices (a widely-used assessment of fluid intelligence and abstract reasoning) when compared with state-of-the-art deep neural network and neuro-symbolic approaches;

6.2.5. These contemporary symbolic-AI approaches are incredibly promising and may soon play a vital role in deploying AI systems that are accurate, robust and explainable (to some extent);

6.2.6. Narrative computation shares some of the concerns of resource-intensiveness and computational complexity as some of the hypothesis search and inductive logic programming approaches, though importantly, narrative computation focuses on engineering behavioral robustness and reasonable resource allocation such that it doesn't have exactly the same concerns as those systems;

6.2.7. Separately, narrative computation can leverage neural networks for a hybridized neural-narrative approach in a somewhat similar way as the contemporary NVSA and similar approaches;

6.2.8. Ultimately, narrative computation is similar to the symbolic-AI approaches of hypothesis search and inductive logic programming, but the specific focus areas in the design of the behavioral and resource-allocation systems are an important differentiator for narrative computation.

6.3. Transformers and Generative AI Approaches

6.3.1. Transformers and generative-AI are powerful models that leverage attention mechanisms to capture contextual dependencies in sequential data and can generate new data samples that resemble a given training dataset;

6.3.2. These generative-AI approaches (like GPT-4, PaLM-2, SAM, BERT, etc), currently demonstrate state of the art capabilities on many tasks, in some cases surpassing human baselines, while demonstrating incredible robustness in a wide array of areas and benchmarks, including SuperGLUE (https://super.gluebenchmark.com/leaderboard), AGIEval (AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models by W. Zhong, et al.), and text-based matrix reasoning (*Emergent Analogical Reasoning in Large Language Models by T. Webb, et al.*);

6.3.3. However, these generative-AI approaches also tend to have limitations related to accuracy (*Do Language Models Know When They're Hallucinating References? By A Agrawal, et al. and Automatic Evaluation of Attribution by Large Language Models by X. Yue et al.*), robustness (https://github.com/jxzhangjhu/Awesome-LLM-Uncertainty-Reliability-Robustness) and explainability (Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback by B. Peng, et al.) - all of which are currently very active areas of research;

6.3.4. Narrative computation has not yet demonstrated capabilities anywhere near the SOTA capabilities of generative-AI approaches on SuperGLUE, AGIEval or Raven's Progressive Matrices;

6.3.5. However, narrative computation has features which directly address some of the noted limitations of generative-AI approaches, specifically accuracy, robustness, and explainability;

6.3.6. And it is possible to combine generative-AI approaches with narrative computation approaches, as a type of neural-narrative approach to AI;

6.3.7. For the time-being generative-AI approaches have real-world systems deployed which are far superior to many other approaches on the benchmarks of SuperGLUE, AGIEval and Raven's Progressive Matrices;

6.3.8. And it remains to be seen how narrative computation approaches will compete against generative-AI approaches on these benchmarks, especially considering open concerns regarding scalability of NCS's.

6.4. Reinforcement Learning

6.4.1. Reinforcement learning (RL), including deep reinforcement learning (DRL) and reinforcement learning from human feedback (RLHF) are important approaches that are related to the contemporary generative-AI approaches discussed in the prior section;

6.4.2. Specifically, RL is often used to fine-tune generative-AI models for specific considerations like alignment and robustness, and, in some cases, to address some of the key limitations of generative-AI approaches;

6.4.3. RL approaches are currently undergoing similar research and development efforts as with generative AI; whereas narrative computation approaches have features that specifically address some of these concerns;

6.4.4. Also, as with neural-narrative combined approach, RL approaches can be combined with narrative computation approaches to address certain areas of concern, perhaps as it relates to the scalability concerns about narrative computation systems.

7. NX-RESEARCH Systems

7.1. NX-Eval™ Systems

7.1.1. Performance assessment and alignment are central concerns for all AI systems, especially for systems that are expected to be accurate, robust and explainable;

7.1.2. Evaluating the performance of AI systems (utility evaluation) in an effective and robust way is an important and challenging task;

7.1.3. Published benchmarks such as SuperGUE, AGIEval and Raven's Performance Matrices offer examples on how such evaluation systems are carefully designed to provide effective and robust assessments of different aspects of AI systems;

7.1.4. NX-Eval™ by NX-RESEARCH are the utility-evaluation systems by which we implement performance assessments in our design processes, alignment mechanisms, and subsystems;

7.1.5. We expect that NX-Eval™ will continue to be an indispensable set of internal systems for our development efforts in producing AI systems that are accurate, robust and effective;

7.1.6. Using the narrative computation approach, we are able to ensure that the internal measures and evaluations of computational states, simulations, goals and plans in our NX™ systems are externally useful; which we also evaluate using our NX-Eval™ modules that focus on external-reasonableness, robustness, explainability, among others.

7.2. Curious-NX™ Agent Platform

7.2.1. Curious-NX™ by NX-RESEARCH is an agent development and testing platform that is specifically-designed to test small scale narrative computation systems;

7.2.2. Rapid prototyping of various designs, systems, and structures is an important aspect of our research process;

7.2.3. We expect Curious-NX™ to continue to be a valuable internal tool for us to evaluate scaling of narrative computation systems, especially on larger datasets, against more challenging benchmarks and under multimodal conditions.

7.3. Research Roadmap

7.3.1. NX-RESEARCH is not hereby publishing any specific roadmap for our research program, which is currently at an extremely-early stage;

7.3.2. But generally, the challenges and limitations described in this white paper are indicative of the focus areas that our research will be engaged with in the next months - specifically, scalability to larger datasets, performance assessments against more challenging benchmarks, and under multimodal conditions.

8. Discussion

8.1. Final Remarks

8.1.1. In the preceding sections, we provided an introduction to narrative computation, including key concepts, functions, features and limitations of this approach to artificial intelligent systems that are accurate, robust and efficient; we compared against other approaches to AI; and we discussed the state of research within NX-RESEARCH;

8.1.2. Our hope is that this white paper has informed the opinions of the reader about the narrative computation framework, including potential possibilities and limitations of the approach, and how narrative computation relates to other approaches to AI.