Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

GPT-BASED MODELS MEET SIMULATION: HOW TO EFFICIENTLY USE LARGE-SCALE PRE-TRAINED LANGUAGE MODELS ACROSS SIMULATION TASKS

Philippe J. Giabbanelli

Department of Computer Science & Software Engineering
Miami University
501 East High Street
Oxford, OH 45056, USA

ABSTRACT

The disruptive technology provided by large-scale pre-trained language models (LLMs) such as ChatGPT or GPT-4 has received significant attention in several application domains, often with an emphasis on high-level opportunities and concerns. This paper is the first examination regarding the use of LLMs for scientific simulations. We focus on four modeling and simulation tasks, each time assessing the expected benefits and limitations of LLMs while providing practical guidance for modelers regarding the steps involved. The first task is devoted to explaining the structure of a conceptual model to promote the engagement of participants in the modeling process. The second task focuses on summarizing simulation outputs, so that model users can identify a preferred scenario. The third task seeks to broaden accessibility to simulation platforms by conveying the insights of simulation visualizations via text. Finally, the last task evokes the possibility of explaining simulation errors and providing guidance to resolve them.

1 INTRODUCTION

Natural Language Generation (NLG) has been in the limelight recently, following the release of ChatGPT and its wide potential application areas from writing (academic) papers to assignments and software code. Much in the same way as 'Google' colloquially refers to using a search engine, ChatGPT has served as a proxy to discuss the opportunities and concerns raised by Large Language Models (LLMs) (van Dis et al. 2023; Zhou et al. 2023). These large-scale pre-trained models are based on transformer architectures (Tay et al. 2022; Wang et al. 2022) and include several versions of GPT (e.g., GPT4 with one trillion parameters) alongside Google's Pathways Language Model (PaLM, whose 540 billion parameters support the Bard chatbot), LLaMA from Meta (available at several sizes), or Megatron-Turing (530 billion parameters) created by Microsoft and NVIDIA (Chowdhery et al. 2022; Smith et al. 2022). Beyond the sensational headlines, there is a growing realization that these models are complex tools that require technical attention before being adequately deployed. As summarized by the editor-in-chief of *Science*, ChatGPT provides "endless entertainment" but ultimately, like other machines, it serves as a tool "for the people posing the hypotheses, designing the experiments, and making sense of the results" (Thorp 2023). For example, researchers illustrated that ChatGPT was not going to perform a literature review by itself, as two thirds of the scientific studies that it discussed did not exist (Haman and Školník 2023); this phenomenon known as *hallucination* is one of the many errors or 'unpredictable qualities' occurring in LLMs (Ganguli et al. 2022; Borji 2023). It is thus important to complement the nascent literature on high-level opportunities and concerns with an emphasis on *practical tasks* and how they may be facilitated with LLMs under the right human intervention, which may include fine-tuning (Ding et al. 2023), asking the right questions (i.e., prompt engineering as discussed in White et al. 2023), and identifying where to correct the generated text.

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

In this paper, we are interested in shifting from using LLMs such as GPT as assistants in high-level science tasks (e.g., summarizing papers) to becoming central actors in specific tasks for Modeling & Simulation (M&S). This shift finds several parallels with the advent of Machine Learning and its impact on M&S (Giabbanelli 2019; Elbattah 2019). Neither NLG nor machine learning are brand new, as their concepts and early systems were operational decades ago (Gatt and Krahmer 2018; Dong et al. 2022). However, their rise is based on the ability of new tools to operate at an unprecedented *scale* while being easily *accessible*. Plethora of online courses can equip practitioners with machine learning skills and a model can be quickly trained thanks to library such as `scikit-learn` or drag-and-drop software. Tools such as GPT have been in existence for several years already, as we presented a prototype using GPT-3 for simulation at the Winter Conference 2022 (Shrestha et al. 2022). But the recent availability of products such as ChatGPT now makes these tools accessible, as there is no need for programming via an API. We can thus expect that NLG will potentially permeate most stages of the M&S process, just as machine learning has become commonplace at the Winter Simulation through a multitude of innovative hybrid simulations (Müller et al. 2022; Ghasemi et al. 2022; Onggo et al. 2018). In this context, this paper contributes to *preparing our research community for this shift by examining which M&S tasks can benefit from NLG and how it would be achieved*. We note that such inventories of candidate tasks for NLG are now abundant in healthcare and medical education (Sallam 2023), business and marketing (Rivas and Zhao 2023), or environmental science (Zhu et al. 2023), but such guidance had yet to be issued for M&S.

This paper proceeds in the order of simulation tasks summarized in Figure 1. This summary only illustrates the key steps of this paper, as we acknowledge that M&S involves several other steps such as the transition from a conceptual model to a *mathematical* specification and its *implementation* as a computational model. As a popular phrase goes, "if all you have is a hammer, everything looks like a nail"; each section thus begins by establishing the *rationale* for using NLG. Then, we detail the *methods* involved, while noting that their maturity decreases as we progress along simulation tasks.

2 EXPLAINING THE STRUCTURE OF A SIMULATION MODEL

2.1 Rationale for Using Natural Language Generation

Modelers work within interdisciplinary teams, where members have (potentially overlapping) roles such as model commissioners who set the purpose of the model or participants who inform its content (Calder et al. 2018). Empirical studies have repeatedly highlighted the importance of *communication skills* in such teams. As discussed by Ahrweiler et al. (2019), "the first and most important [demand] is that the clients want to understand the model", which means that the *structure* of the model and the logic of its decision should be clear, rather than treated as blackbox that only allows to view and discuss outputs. Clearly conveying a model's structure is challenging, since team members may not be expert in modeling techniques hence they delving into code is not a viable solution. Although modeling languages (e.g., UML, SysML) can be familiar tools for model development, their steep learning curve for participants also presents an obstacle (Padilla et al. 2019). Our experiments confirmed that even a graph that only shows concepts and whether they are connected can be a significant learning curve for participants (senior executives), who struggled to provide confident and timely answers for basic questions about the model (Giabbanelli and Vesuvala 2023). Natural Language Generation thus opens up the opportunity to explain a model in a format that is potentially accessible to all parties: textual narratives.

Explaining model as narratives has limitations. For example, modelers should still be involved to assist participants. As argued by Gilbert et al. (2018), "it is impossible to capture in a report all the nuances of the model simplifications, data weaknesses etc. in a way that [participants] can use reliably". In addition, a report automatically generated from a schema may not be able to cover all topics that ought to be communicated, such as the purpose of the model (Grimm et al. 2020). However, we argue that an automatically generated report can at least convey the *structure* of a model so that participants understand which variables are involved and how they interact. Providing these expectations can address the existing

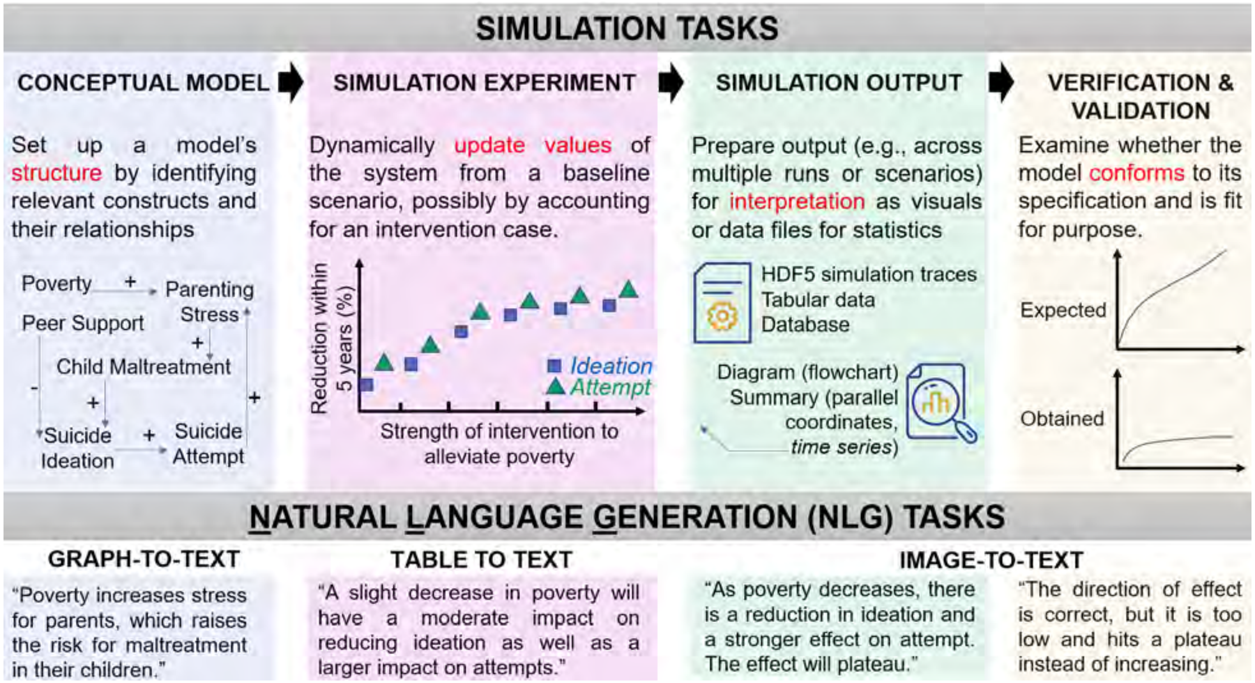

Figure 1: After creating a conceptual model, it is implemented and utilized to perform experiments whose outputs support decision-making activities. Outputs are also used to ensure the correctness of the model vis-à-vis its specification (i.e., verification) or the adequacy of the model with regards to its real-world counterpart (i.e., validation). In this paper, we map each of these steps to a task in NLG.

disconnect between the transparent and often informal process to *elicit* information from participants, and the opacity of the resulting model (Figure 2). Since transparency and trust in a model often come together (Falconi and Palmer 2017), we posit that a careful use of NLG to turn models into reports may ultimately increase the engagement of participants with the modeling process and their support of the decisions suggested by the simulations.

2.2 Core Methodological Components

Although the relation between textual descriptions and conceptual models has been discussed at the Winter Simulation conference, it has primarily been from the viewpoint of extracting model elements from text (Shuttleworth and Padilla 2022). This involves Natural Language *Processing*, with a focus on identifying entities such as agents and their properties. In contrast, turning a model into text is a matter of Natural Language *Generation* and it involves vastly different steps, particularly when operating via LLMs. Although LLMs such as GPT-4 have made headways with the use of images as inputs (Section 4), modelers cannot generally expect to just 'drop' a schema of their conceptual model and have it turned into a report. The schema first needs to be converted into a textual form.

Since schema (e.g., UML, causal maps) often depict concepts and their relations, the corresponding NLG task is known as *graph-to-text*. Graphs can have cycles: for example, the conceptual model in Figure 1 has a cycle of parenting stress increasing the risk of suicide attempt in their children, thus causing more stress. In contrast, sentences must be linear. The conversion thus starts by turning the model's schema into linearized sentences, but it cannot simply be achieved by removing parts of the schema to break existing cycles (this is known as a 'loss of structural information'). One strategy is to break the schema into parts that *collectively* can recreate the full schema, thus resulting in some nodes being duplicated among the decomposed parts. These parts should be kept small (Figure 2) as experiments show that the quality of the

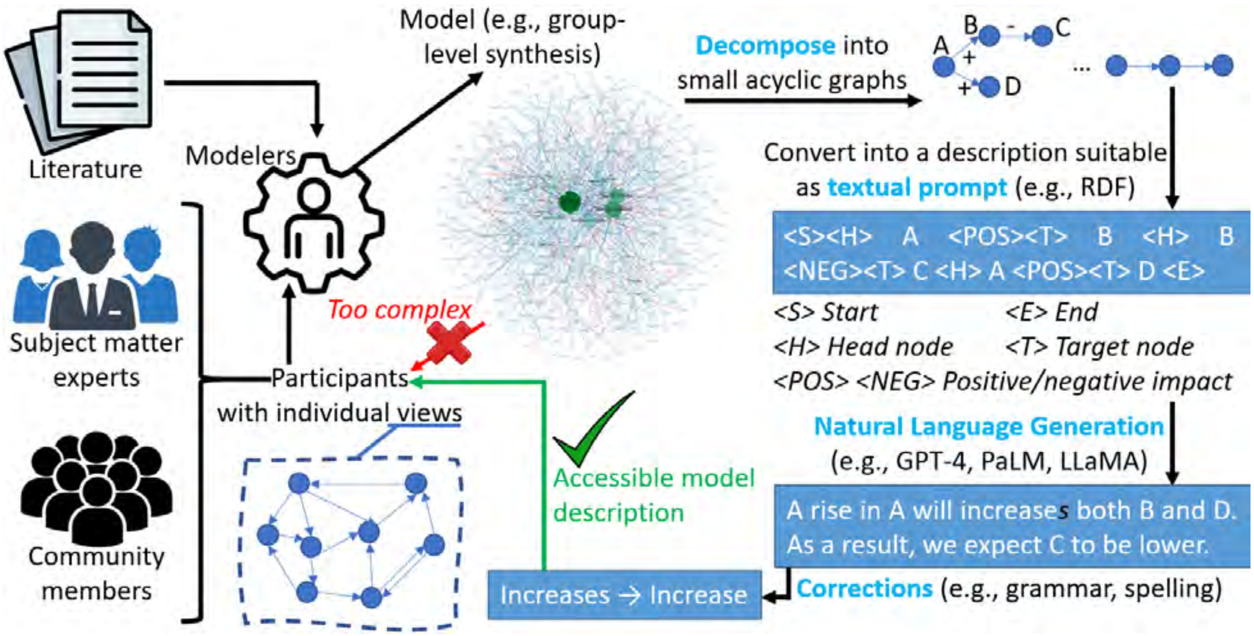

Figure 2: Modelers can work with the literature and/or participants to create a conceptual model. Each article or participant may provide a *small* model on a facet of the problem, but the overall model may be larger and harder too explain due to the sheer number of constructs and relationships or complex dynamics (e.g., loops). NLG can explain the model in textual form and it requires several steps: decomposing the model into smaller inputs, converting them to textual prompts, and correcting the output (if necessary).

generated text deteriorates with larger inputs (Shrestha et al. 2022). Linearization is a problem in itself, as it cannot simply be achieved by removing parts of the schema to break existing cycles (this is known as a 'loss of structural information'). Alternatively, a new schema could be defined and introduce additional nodes to encapsulate the meaning of cycles (Rodrigues Ribeiro 2022). So far, linearization has primarily been studied for knowledge graphs rather than simulation schema, hence modelers would need to write custom linearization algorithms.

Once the graph is linearized, it needs to be expressed as a textual input. It does not suffice to just write out the graph as words, as the edges would be ambiguous. For example, the list A,B,C,D could be interpreted as the edges (A,B) and (C,D), or (A,B), (B,C), (C,D). An unambiguous representation thus uses flags to separate the origin node of an edge from the target. The Resource Description Framework (RDF) is widely used for this purpose (Yang et al. 2020; Zhang et al. 2023). Depending on its sophistication, the LLM may need to be *fine-tuned* by being presented with numerous RDF inputs and the expected sentence to generate. Once the (fine-tuned) LLM creates sentences, they may still need to be corrected. This can involve correcting typos (e.g., in earlier versions such as GPT-2), avoiding the redundancy that readers quickly associated with machine-generated text (Liu et al. 2023), or mitigating various forms of biases. The latter has received abundant attention, as authors have discussed the presence (and sometimes the inevitability) of bias (Ferrara 2023) on sex, race, religion, or disability – all of which are protected classes in US law on discrimination. However, studies on bias have not yet been conducted in the case of text generated from model schema, hence additional work is needed to assess the problem and whether some application fields are more at risk (e.g., models of social systems vs. physical systems).

Accomplishing the above steps results in generating *sentences*, but it does not yet make a report. As an analogy, consider teaching: an instructor cannot deliver content material in random order, or teach a second year elective class in the same way as a graduate seminar. Generating a report also needs to account for the *audience* and orchestrate sentences into meaningful *paragraphs* with an appropriate flow. Satisfying either of these requirements is currently an open topic when applying NLG to explaining simulation models. Because modeling is conducted in an interdisciplinary setting, insufficiently accounting for "the languages of the different research traditions can lead to misunderstanding and resentment" (Smaldino 2020). We posit that forming paragraphs may be an easier problem (and hence a prime research target) because the readability of paragraphs can be automatically measured for a language in general (e.g., using the Flesch–Kincaid readability test), whereas the appropriateness of terms and style with regards to a scientific discipline does not have an algorithmic scoring method.

3 HANDLING DYNAMICITY: COMPARING OUTCOMES FROM PREDICTIVE SIMULATIONS

3.1 Rationale for Using Natural Language Generation

In the previous section, we discussed the benefits of explaining the *structure* of a model by converting its schema to text. At a high-level, explaining the *simulation outcomes* would yield similar benefits, such as greater transparency and engagement. To further estimate benefits and limitations, it is necessary to precisely define the task of 'explaining outcomes'. A simulation model may be presented with a number of cases, also known as *scenarios* or what-if questions. For instance, in a model for suicide prevention (Figure 3), such scenarios could include educating parents to avoid harsh discipline, improving coping mechanisms in children, or providing treatment for substance misuse. One of the cases may be marked as baseline, thus reflecting the current state of the world in the absence of hypothetical interventions. The goal is to inform model users about the difference in simulation outcomes across cases, such that they can *choose* the best candidate interventions. By summarizing simulation outcomes across cases to focus on key differences, NLG can reduce the cognitive efforts involved in decision-making.

There are potential limitations when attempting to automatically convey the main differences across cases. First, a *textual format may not always be most efficient*. For example, if a simulation has only one critical output, then users may prefer a bar chart visualization (Figure 4) that shows the output of interest (y-axis) across simulation cases (x-axis). Since a bar chart relies on preattentive visual properties such as line length (Wolfe and Utochkin 2019) to bring attention to data points that stand out (e.g., lowest, highest), users would quickly notice which case yields the minimal/maximal outcome and hence would be preferred. A well-constructed visualization would thus be more effective than text. Second, a simulation may have multiple objectives that cannot all be optimized. For example, interventions for obesity could be characterized by indicators related to physical health (e.g., type-2 diabetes, musculoskeletal disorders, hypertension) as well as mental health (e.g., self-esteem and body image). Different users may give more importance to some of these indicators, and these preferences may be implicit. The incorporation of implicit preferences in multi-objective optimization (Cruz-Reyes et al. 2017) is a complex problem and has yet to be studied in the context of summaries generated by NLG. This section thus focuses on cases that have multiple outputs of interest and assumes that their importance is either equal or can be explicitly quantified.

3.2 Core Methodological Components

The input is a set of simulation cases, each containing the value of the model's constructs from the initial to the final iteration. Consider without loss of generality that the output seeks to summarize the baseline scenario, the design of each intervention and how it leads to changes compared to the baseline (Figure 3); the remainder of this section would be similar if there was no 'baseline' case. As explained in section 2.2, a LLM does not directly go from the input to the desired output: modelers need to perform at least two additional steps. The first step is to gather the results into a single table that contains the characteristics of each intervention and the final value of each construct (Figure 3, bottom right); characteristics and final values can be expressed as a difference with respect to the baseline (if applicable). By transforming simulation outputs into one table, we frame the problem as a *table-to-text task*, which has been well studied.

The second step is to transform the table into textual input for a LLM. Early methods follow a 'pipeline paradigm' in which a *table transformation module* applies a *template* to turn a table into text (Gong et al.

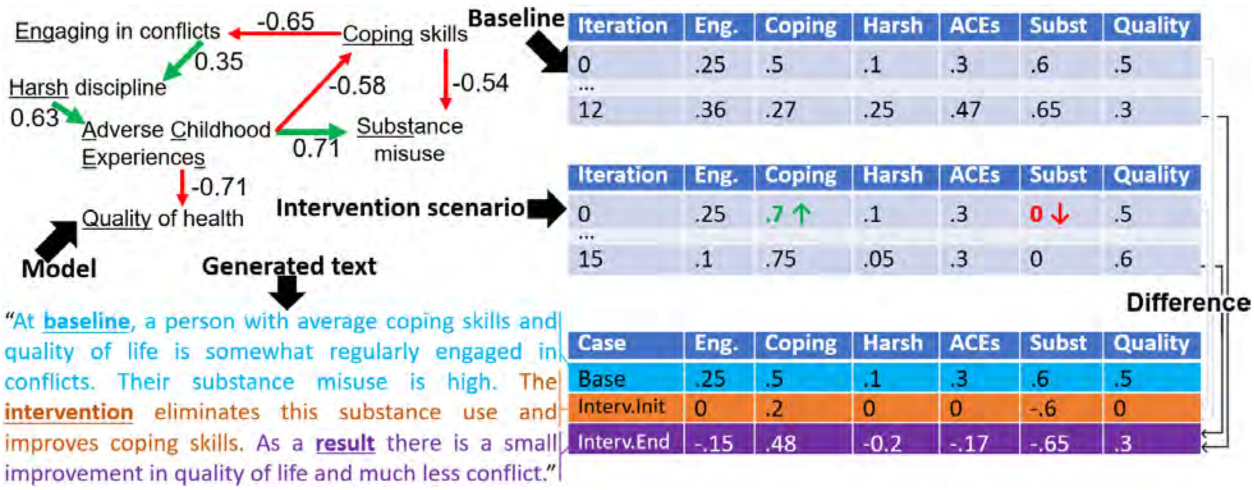

Figure 3: The illustrative model of Adverse Childhood Experiences (ACEs) is a part of a larger model on youth suicide, detailed in Giabbanelli et al. (2022). The implications of the baseline or 'status quo' scenario can be organized as a table, showing how the 6 constructs of the ACEs model change over discrete iterations until a final step. An intervention also consists of a table, whereby the initial values of some constructs change (due to the intervention) hence the final values may also change.

2020). Although this approach has the advantage of being conceptually simple, it requires users to design a template. Furthemore, the output is limited (hence repetitive) due to a reliance on set templates and rules. Newer approaches use end-to-end methods based on neural networks (Yang et al. 2021), but they depend on a large training set and may not be readily applicable to the specific context of a simulation model. We refer the reader to Guo et al. (2023) for an overview of current options, and to Table 4 by Sharma et al. (2022) for a summary of methods based on the application dataset. While many prior works are concerned with tables that contain words, this is not directly applicable to simulation traces since they consist of *numerical outputs*. We thus recommend methods specifically designed for tables with numerical content, such as Suadaa et al. (2021).

Accomplishing the above steps results in a *complete* transformation of a table into text. If the table is short (e.g., few simulation cases and/or constructs), then model users may be able to read the generated text and identify the best simulation case. However, additional steps would be necessary to *selectively* transform larger tables and avoid overly verbose reports (Figure 4). As shown in Figure 3, it is not necessary to generate text regarding the initial state of every construct for every case: we can simply state which constructs had a different value than in the baseline case. We also do not need to specify the impact on every construct at the end of the simulation: applying a user-defined filter (e.g., ignore changes of less than 10%, only include changes on three specific constructs) can trim the list. Beyond these simple means to keep the text short, we note that text summarization algorithms may offer additional solutions (Gupta and Gupta 2019; Raffel et al. 2020). These algorithms operate either by compiling the most important existing sentences (*extractive* summarization e.g., Textrank, BERT-ext, Longformer-Ext) or by generating sentences (*abstractive* summarization e.g., BART, T5), which tend to have higher readability and are more concise but may not exactly reflect the meaning of the original text (Alomari et al. 2022). However, new summarization algorithms would need to be developed since the existing ones are not readily applicable to simulation data, which consist entirely of numbers. Indeed, general purpose summarization algorithms either ignore numbers or treat sentences with numerical data as more important (Sindhu and Seshadri 2022); neither option would help to identify the main characteristics of simulation scenarios or the key changes that they produce.

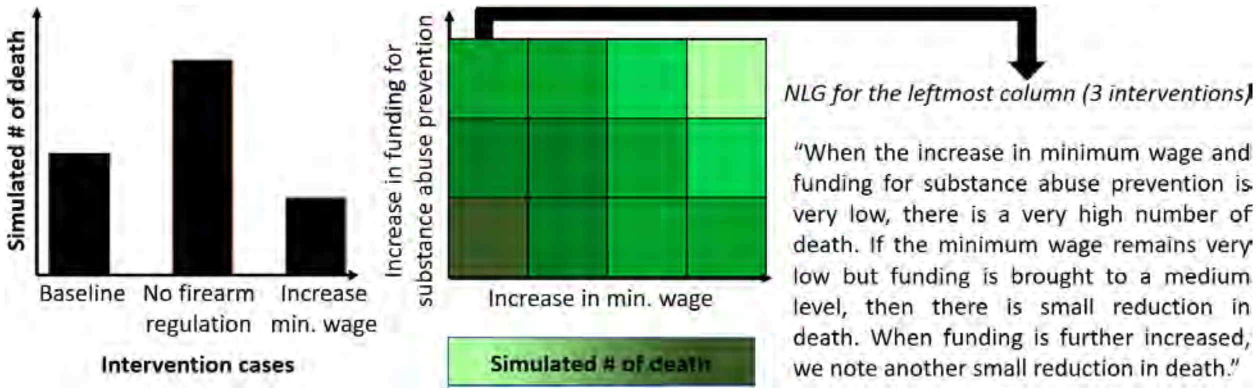

Figure 4: A simple bar chart (left) can quickly reveal the preferred scenario for a model user, as scanning the image to find the lowest/highest bar relies on preattentive visual features. Even when scenarios are not defined by intervention category (left) but rather than different levels of change in numerical parameters (center), a visualization can rely on other preattentive visual features such as hue to guide decision makers. In contrast, a complete text report can be overly verbose (right) and less effective for decision-making. There are thus situations in which visualizations can be used instead of, or in complement to, NLG.

4 EMERGING CAPABILITIES: SIMULATION VISUALIZATION AS TEXT

4.1 Rationale for Using Natural Language Generation

Since about 70% of all human sensory receptors are in the eyes, it is no surprise that scientific visualizations are commonplace to derive insight from simulation results. New packages are regularly developed to support visualizations of simulation outputs in diverse application areas such as the large simulation datasets used in Earth system science (Li et al. 2019; Wang et al. 2019) or molecular dynamics simulation (Hildebrand et al. 2019). However, such visualizations create obstacles for individuals with visual impairments, which applies to 253 million individuals worldwide (Ackland et al. 2017). An academic group developing a simulation software for its own purpose or research can decide to rely extensively on visualizations to interpret results. However, when a simulation software is developed for government agencies, laws on information technology can require that the simulation be accessible for people with disabilities (including visual impairments). This applies even if the software resides solely on the intranet and is intended to be used by a team that does not currently have members with visual impairments. Section 508 of the Rehabilitation Act in the United States enacts such requirements, which are echoed in the Barrier-Free Information Technology Ordinance from Germany, and various other disability discrimination acts. An approach to ensure compliance with regulations is to provide a data table for every simulation visualization (Figure 5), as screen readers can read tables one cell at a time. This *technically* supports accessibility, to the same extent as a wheelchair ramp circling around a building at a steep angle would *technically* provide access. Reading every simulation data point is not only cumbersome, but it also prevents users from identifying patterns, just as it would be challenging to make sense of a picture if it was read one pixel at a time. Turning visualizations into written reports focusing on the main patterns would thus broaden accessibility to simulations and ensure compliance with legal requirements.

4.2 Core Methodological Components

Neural networks have been used in browser extensions to automatically decode charts (Choi et al. 2019) and newer LLMs (e.g., GPT-4) now include the ability of transforming images to text. However, this is still only an *emerging* capability, hence we note that significant research efforts are still needed. Recent studies can guide modelers who seek to operate LLMs in the near future to generate textual summaries of their visualizations. First, evaluations with visually impaired individuals concluded that the preferred natural

	Population		Hispanic		Non-Hispanic Black		Non-Hispanic White		Other	
Time	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male
Q1	32227.2727 27272728	24028.3018 86792455	32227.2727 27272728	24028.3018 86792455	32227.2727 27272728	24028.3018 86792455	32227.2727 27272728	24028.3018 86792455	32227.2727 27272728	24028.3018 86792455
Q3	33227.2727 2727273	25028.3018 86792455	33227.2727 2727273	25028.3018 86792455	33227.2727 2727273	25028.3018 86792455	33227.2727 2727273	25028.3018 86792455	33227.2727 2727273	25028.3018 86792455
Min	31727.2727 27272728	23528.3018 86792455	31727.2727 27272728	23528.3018 86792455	31727.2727 27272728	23528.3018 86792455	31727.2727 27272728	23528.3018 86792455	31727.2727 27272728	23528.3018 86792455
Max	33727.2727 2727273	25528.3018 86792455	33727.2727 2727273	25528.3018 86792455	33727.2727 2727273	25528.3018 86792455	33727.2727 2727273	25528.3018 86792455	33727.2727 2727273	25528.3018 86792455
Median	32727.2727 27272728	24528.3018 86792455	32727.2727 27272728	24528.3018 86792455	32727.2727 27272728	24528.3018 86792455	32727.2727 27272728	24528.3018 86792455	32727.2727 27272728	24528.3018 86792455

Figure 5: An Agent-Based Model platform for a U.S. federal client was discussed at the Winter Simulation Conference with regards to accessibility requirements (Huddleston et al. 2022). Providing a table for each visualization contributes to meeting these requirements, as the HTML code can be used by screen readers.

language descriptions for visualizations were *very reader-specific* (Lundgard and Satyanarayan 2021). Modelers thus need to be cognizant of their target audience, ideally by eliciting individual preferences. Second, the usefulness of the description *depends on the task* that the reader seeks to accomplish. For instance, the heatmap in Figure 4 can be examined to find the best intervention (top-right corner) or to know how the outcome depends on the two control parameters (strictly decreases as a function of both parameters). Third, users may expect different *verbosity levels*. As exemplified by Zong et al. (2022), "at higher verbosity the screen reader announces more structural, wayfinding content (e.g. the start and end of regions)." More verbose summaries are not necessarily more effective, and a few bullet point statements can be a better starting point (Brath and Hagerman 2021). Modelers may thus consider starting their NLG prompts by requiring a few short statements instead of a comprehensive summary.

5 THE NEXT FRONTIER? EXPLAINING AND FIXING SIMULATION ERRORS

5.1 Rationale for Using Natural Language Generation

Once a simulation model can perform experiments, we need to ensure the correctness of the implementation with regards to the specification (verification) and with respect to the expected approximation of a real-world phenomenon (validation). At first, this may resemble the task of detecting and explaining implementation errors, which is increasingly studied in the nascent literature on LLMs for debugging. However, the literature on automated debugging has mostly examined the behavior of *small functions*, for example by using prompts to state that a function 'obviously has a bug' because it returned $Output_{prompt}$ instead of $Output_{expected}$ for a given *Input* (Liventsev et al. 2023). While locating and fixing bugs within functions is undoubtedly beneficial, verification and validation are concerned with errors that may happen at the level of the *whole model*. Two sub-tasks are involved: explaining why there are errors by comparing simulation outputs with expectations, and providing guidance to address these errors. We posit that LLMs may be best employed to explain errors (e.g., ‘without vaccines and social distancing your population saw a reduction in COVID-19 cases but we believe that either of these interventions would have been needed to yield such results’) than to identify them, which may be achieved through established statistical techniques. We also note that the guidance offered would primarily consist of generating hypotheses (Kang et al. 2023) such as ‘your agents may need an exposed stage before infection’. This would already be tremendously supportive for *modelers in training*, by automating part of the feedback that is otherwise provided by instructors. We caution against expectations that a LLM may directly write the code logic for a model, as current LLMs write text that looks like code but does not always run and they are best suited to automate mundane tasks by writing functions that have already been encountered in their training data (Merow et al. 2023).

5.2 Core Methodological Components

We view the use of LLMs to explain and address simulation errors as the next frontier, as the tools are further from practical use than in the previous sections. Two components can play an important role in allowing an LLM to identify errors. First, the LLM needs to relate the behavior of this specific model to its general knowledge base, which can leverage an LLM’s ability as a causal learner. Modelers would thus need to write a prompt that states the goal of the model and summarizes its causal pathways. For example: “A model for COVID-19 assumes that most people have not been exposed to the virus. There is a chance of getting sick upon exposure. Infected people either recover or die”. Second, the LLM would benefit from contextual information on the error (e.g., which outputs are lower/higher than expected and by how much), which can be provided by existing statistical packages that compare model outputs with expected outputs. The result can complete the prompt as follows: “After one year, without vaccines or social distancing, nobody is infected by COVID-19 anymore. This is obviously wrong. Why?” Note that keywords such as ‘obvious’ can trigger LLMs such as GPT to give particular considerations to some statements (Liventsev et al. 2023). Providing guidance on the existence of the error is an arduous task. A LLM does not understand the structure of a simulation model, and it may also struggle to also understand how the code is related to the model’s output since that can be an emerging behavior. It is likely that a LLM would need to be taught (by prompts) about the behavior of related models and then rely on transfer learning to investigate abnormalities in the proposed model.

A potential challenge arises when the model is correct, but its outputs are different from a modeler’s expectation. A prompt stating that a correct outcome (albeit unexpected) is ‘obviously wrong’ would be biased. In addition, a modeler may use such prompts repeatedly until the LLM finally provides the desired response that validates the modeler’s mental schema. In this case, the LLM would be forced to hallucinate. Until safeguards are in place, a LLM may thus be better suited for a discussion about the methods and expectations rather than directly for fixing errors.

6 CONCLUSION

We focused on tasks that are enabled by the emerging technology of LLMs. There are cases in which tasks that used to be performed by other technology (e.g., question-answering systems based on information retrieval rather than machine learning) are now also using LLMs, in the manner of an *oracle*. For instance, Q&A systems have previously served to check whether the concepts and relationships of a conceptual model built by a modeling team are supported by the literature (Sandhu et al. 2019). The same ‘yes’ or ‘no’ questions could be asked to a GPT-based model (e.g., ‘given the following documents [...], can infection from COVID-19 follow exposure to COVID-19 particles?’), but this would be a relatively minor update of a technology rather than a breakthrough for M&S. Recently, researchers have shifted from *checking* a proposed conceptual model to *building* it automatically. This was first done by retrieving a corpus and repeatedly running a Q&A system in the same manner as a facilitator would develop a model by talking with a subject matter expert (Davis et al. 2022), but researchers are now examining the *feasibility* of relying entirely on LLMs to build causal graphs (Long et al. 2023; Zhang et al. 2023). We believe that this automation is an exciting step for M&S, particularly if the conceptual model built by LLMs can then be mapped onto simulation building blocks to automatically create a working prototype (Schroeder et al. 2022). The guidance provided in this paper can thus be updated as progress is realized by the NLG community, to ensure that it ultimately benefits modeling and simulation.

ACKNOWLEDGEMENTS

Several of the reflections in this paper are the product of fruitful discussions with numerous individuals. In particular, the author wishes to thank Mr. Anish Shrestha and Mr. Tyler Gandee, whose theses help to realize the potential but also the technical challenges in using GPT-based models. The author has also benefited from stimulating exchanges of ideas with various participants (including Dr. Ameeta Agrawal and Dr. Jose Padilla) at a research seminar co-organized at Miami University with Dr Vijay Mago.

REFERENCES

Ackland, P., S. Resnikoff, and R. Bourne. 2017. "World Blindness and Visual Impairment: Despite Many Successes, the Problem is Growing". *Community Eye Health* 30(100):71–73.

Ahrweiler, P., D. Frank, and N. Gilbert. 2019. "Co-Designing Social Simulation Models for Policy Advise: Lessons Learned from the INFSO-SKIN Study". In *Proc. 2019 Spring Simulation Conference (SpringSim), April 29–May 2 2019. Tucson, AZ, USA*, 1–12. IEEE.

Alomari, A., N. Idris, A. Q. M. Sabri, and I. Alsmadi. 2022. "Deep Reinforcement and Transfer Learning for Abstractive Text Summarization: A Review". *Computer Speech & Language* 71:101276.

Borji, A. 2023. "A Categorical Archive of ChatGPT Failures". *arXiv preprint arXiv:2302.03494. https://arxiv.org/abs/2302.03494*.

Brath, R., and C. Hagerman. 2021. "Automated Insights on Visualizations with Natural Language Generation". In *2021 25th International Conference Information Visualisation (IV), 5-9 July 2021, Sydney, Australia*, 278–284. IEEE.

Calder, M., C. Craig, D. Culley, R. De Cani, C. A. Donnelly, R. Douglas, and B. Edmonds. 2018. "Computational Modelling for Decision-Making: Where, Why, What, Who and How". *Royal Society open science* 5(6):172096.

Choi, J., S. Jung, D. G. Park, J. Choo, and N. Elmqvist. 2019. "Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization". In *Computer Graphics Forum*, Volume 38, 249–260. Wiley Online Library.

Chowdhery, A., S. Narang, J. Devlin, M. Bosma, G. Mishra, and A. Roberts. 2022. "Palm: Scaling Language Modeling With Pathways". *arXiv preprint arXiv:2204.02311; https://arxiv.org/abs/2204.02311*.

Cruz-Reyes, L., E. Fernandez, P. Sanchez, C. A. C. Coello, and C. Gomez. 2017. "Incorporation of Implicit Decision-Maker Preferences in Multi-Objective Evolutionary Optimization Using a Multi-Criteria Classification Method". *Applied Soft Computing* 50:48–57.

Davis, C. W., A. J. Jetter, and P. J. Giabbanelli. 2022. "Automatically Generating Scenarios from a Text Corpus: A Case Study on Electric Vehicles". *Sustainability* 14(13):7938.

Ding, N., Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen et al. 2023. "Parameter-efficient Fine-tuning of Large-scale Pre-trained Language Models". *Nature Machine Intelligence*:1–16.

Dong, C., Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, and M. Yang. 2022. "A Survey of Natural Language Generation". *ACM Computing Surveys* 55(8):1–38.

Elbattah, M. 2019. "How can Machine Learning Support the Practice of Modeling and Simulation?–A Review and Directions for Future Research". In *Proc. 2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), 7-9 Oct. 2019, Cosenza, Italy*, 1–7. IEEE.

Falconi, S. M., and R. N. Palmer. 2017. "An Interdisciplinary Framework for Participatory Modeling Design and Evaluation–What Makes Models Effective Participatory Decision Tools?". *Water Resources Research* 53(2):1625–1645.

Ferrara, E. 2023. "Should ChatGPT be Biased? Challenges and Risks of Bias in Large Language Models". *arXiv preprint arXiv:2304.03738; https://arxiv.org/abs/2304.03738*.

Ganguli, D., D. Hernandez, L. Lovitt, A. Askell, Y. Bai, A. Chen, T. Conerly, N. Dassarma, D. Drain, N. Elhage et al. 2022. "Predictability and Surprise in Large Generative Models". In *Proc. ACM Conference on Fairness, Accountability, and Transparency, June 21 - 24 2002, Seoul, Korea*, 1747–1764.

Gatt, A., and E. Krahmer. 2018. "Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation". *Journal of Artificial Intelligence Research* 61:65–170.

Ghasemi, A., K. E. Kabak, and C. Heavey. 2022. "Demonstration of the Feasibility of Real Time Application of Machine Learning to Production Scheduling". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 3406–3417. IEEE.

Giabbanelli, P. J. 2019. "Solving Challenges at the Interface of Simulation and Big Data Using Machine Learning". In *Proc. Winter Simulation Conference (WSC), 8-11 Dec. 2019, National Harbor, Maryland, USA*, 572–583. IEEE.

Giabbanelli, P. J., K. L. Rice, M. C. Galgoczy, N. Nataraj, M. M. Brown, C. R. Harper, M. D. Nguyen, and R. Foy. 2022. "Pathways to Suicide or Collections of Vicious Cycles? Understanding the Complexity of Suicide Through Causal Mapping". *Social network analysis and mining* 12(1):60.

Giabbanelli, P. J., and C. X. Vesuvala. 2023. "Human Factors in Leveraging Systems Science to Shape Public Policy for Obesity: A Usability Study". *Information* 14(3):196.

Gilbert, N., P. Ahrweiler, P. Barbrook-Johnson, K. P. Narasimhan, and H. Wilkinson. 2018. "Computational Modelling of Public Policy: Reflections on Practice". *Journal of Artificial Societies and Social Simulation* 21(1).

Gong, H., Y. Sun, X. Feng, B. Qin, W. Bi, X. Liu, and T. Liu. 2020, December. "TableGPT: Few-shot Table-to-Text Generation with Table Structure Reconstruction and Content Matching". In *Proc. 28th International Conference on Computational Linguistics*, edited by D. Scott, N. Bel, and C. Zong, 1978–1988. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Grimm, V., A. S. Johnston, H.-H. Thulke, V. Forbes, and P. Thorbek. 2020. "Three Questions to Ask Before Using Model Outputs for Decision Support". *Nature Communications* 11(1):4959.

Guo, Z., M. Yan, J. Qi, J. Zhou, Z. He, Z. Lin, G. Zheng, and X. Wang. 2023. "Few-Shot Table-to-Text Generation with Prompt-based Adapter". *arXiv preprint arXiv:2302.12468; https://arxiv.org/abs/2302.12468*.

Gupta, S., and S. K. Gupta. 2019. "Abstractive Summarization: An Overview of the State of the Art". *Expert Systems with Applications* 121:49–65.

Haman, M., and M. Školník. 2023. "Using ChatGPT to Conduct a Literature Review". *Accountability in Research*:1–3.

Hildebrand, P. W., A. S. Rose, and J. K. Tiemann. 2019. "Bringing Molecular Dynamics Simulation Data Into View". *Trends in Biochemical Sciences* 44(11):902–913.

Huddleston, J., M. C. Galgoczy, K. A. Ghumrawi, P. J. Giabbanelli, K. L. Rice, N. Nataraj, and M. M. Brown. 2022. "Design and Deployment of a Simulation Platform: Case Study of an Agent-Based Model for Youth Suicide Prevention". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 2582–2593. IEEE.

Kang, S., B. Chen, S. Yoo, and J.-G. Lou. 2023. "Explainable Automated Debugging via Large Language Model-driven Scientific Debugging". *arXiv preprint arXiv:2304.02195; https://arxiv.org/abs/2304.02195*.

Li, S., S. Jaroszynski, S. Pearse, L. Orf, and J. Clyne. 2019. "Vapor: A Visualization Package Tailored to Analyze Simulation Data in Earth System Science". *Atmosphere* 10(9):488.

Liu, Y., Z. Zhang, W. Zhang, S. Yue, X. Zhao, X. Cheng, Y. Zhang, and H. Hu. 2023. "ArguGPT: Evaluating, Understanding and Identifying Argumentative Essays Generated by GPT Models". *arXiv preprint arXiv:2304.07666; https://arxiv.org/abs/2304.07666*.

Liventsev, V., A. Grishina, A. Härmä, and L. Moonen. 2023. "Fully Autonomous Programming with Large Language Models". *arXiv preprint arXiv:2304.10423; https://arxiv.org/abs/2304.10423*.

Long, S., T. Schuster, and A. Piché. 2023. "Can Large Language Models Build Causal Graphs?". *arXiv preprint arXiv:2303.05279; https://arxiv.org/abs/2303.05279*.

Lundgard, A., and A. Satyanarayan. 2021. "Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content". *IEEE transactions on visualization and computer graphics* 28(1):1073–1083.

Merow, C., J. M. Serra-Diaz, B. J. Enquist, and A. M. Wilson. 2023. "AI Chatbots Can Boost Scientific Coding". *Nature Ecology & Evolution*:1–3.

Müller, M., T. Reggelin, I. Kutsenko, H. Zadek, and L. S. Reyes-Rubiano. 2022. "Towards Deadlock Handling with Machine Learning in a Simulation-Based Learning Environment". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 1485–1496. IEEE.

Onggo, B. S., N. Mustafee, A. Smart, A. A. Juan, and O. Molloy. 2018. "Symbiotic Simulation System: Hybrid Systems Model Meets Big Data Analytics". In *Proc. Winter Simulation Conference (WSC), 9-12 Dec. 2018, Gothenburg, Sweden*, 1358–1369. IEEE.

Padilla, J. J., D. Shuttleworth, and K. O'Brien. 2019. "Agent-Based Model Characterization Using Natural Language Processing". In *Proc. Winter Simulation Conference (WSC), 8-11 Dec. 2019, National Harbor, Maryland, USA*, 560–571. IEEE.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. "Exploring the Limits of Transfer Learning With a Unified Text-to-Text Transformer". *The Journal of Machine Learning Research* 21(1):5485–5551.

Rivas, P., and L. Zhao. 2023. "Marketing with ChatGPT: Navigating the Ethical Terrain of GPT-Based Chatbot Technology". *AI* 4(2):375–384.

Rodrigues Ribeiro, L. F. 2022. *Graph-based Approaches to Text Generation*. Ph. D. thesis, Technische Universität Darmstadt.

Sallam, M. 2023. "ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns". *Healthcare* 11(6):887.

Sandhu, M., P. J. Giabbanelli, and V. K. Mago. 2019. "From Social Media to Expert Reports: The Impact of Source Selection on Automatically Validating Complex Conceptual Models of Obesity". In *Social Computing and Social Media. Design, Human Behavior and Analytics: 11th International Conference, SCSM 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26-31, 2019, Proceedings, Part I 21*, 434–452. Springer.

Schroeder, S. A., C. Vendome, P. J. Giabbanelli, and A. M. Montfort. 2022. "Towards Reusable Building Blocks to Develop COVID-19 Simulation Models". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 569–580. IEEE.

Sharma, M., A. Gogineni, and N. Ramakrishnan. 2022. "Innovations in Neural Data-to-Text Generation". *arXiv preprint arXiv:2207.12571; https://arxiv.org/abs/2207.12571*.

Shrestha, A., K. Mielke, T. A. Nguyen, and P. J. Giabbanelli. 2022. "Automatically Explaining a Model: Using Deep Neural Networks to Generate Text From Causal Maps". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 2629–2640. IEEE.

Shuttleworth, D., and J. Padilla. 2022. "From Narratives to Conceptual Models via Natural Language Processing". In *Proc. Winter Simulation Conference (WSC), 11-14 Dec. 2022, Singapore*, 2222–2233. IEEE.

Sindhu, K., and K. Seshadri. 2022. *Text Summarization: A Technical Overview and Research Perspectives*, Chapter 13, 261–286. John Wiley & Sons, Ltd.

Smaldino, P. E. 2020. "How to Translate a Verbal Theory into a Formal Model". *Social Psychology* (51):207–218.

Smith, S., M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti et al. 2022. "Using Deepspeed and Megatron to Train Megatron-turing NLG 530b, a Large-scale Generative Language Model". *arXiv preprint arXiv:2201.11990; https://arxiv.org/abs/2201.11990*.

Suadaa, L. H., H. Kamigaito, K. Funakoshi, M. Okumura, and H. Takamura. 2021, August. "Towards Table-to-Text Generation with Numerical Reasoning". In *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, edited by C. Zong, F. Xia, W. Li, and R. Navigli, 1451–1465. Online: Association for Computational Linguistics.

Tay, Y., M. Dehghani, D. Bahri, and D. Metzler. 2022. "Efficient Transformers: A Survey". *ACM Computing Surveys* 55(6):1–28.

Thorp, H. H. 2023. "ChatGPT is Fun, but not an Author". *Science* 379(6630):313–313.

van Dis, E. A., J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting. 2023. "ChatGPT: Five Priorities for Research". *Nature* 614(7947):224–226.

Wang, C., J. Hou, D. Miller, I. Brown, and Y. Jiang. 2019. "Flood Risk Management in Sponge Cities: The Role of Integrated Simulation and 3D Visualization". *International Journal of Disaster Risk Reduction* 39:101139.

Wang, H., J. Li, H. Wu, E. Hovy, and Y. Sun. 2022. "Pre-Trained Language Models and Their Applications". *Engineering*.

White, J., Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt. 2023. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT". *arXiv preprint arXiv:2302.11382; https://arxiv.org/abs/2302.11382*.

Wolfe, J. M., and I. S. Utochkin. 2019. "What is a Preattentive Feature?". *Current opinion in psychology* 29:19–26.

Yang, Y., J. Cao, Y. Wen, and P. Zhang. 2021. "Table to Text Generation With Accurate Content Copying". *Scientific reports* 11(1):22750.

Yang, Z., A. Einolghozati, H. Inan, K. Diedrick, A. Fan, P. Donmez, and S. Gupta. 2020, 12. "Improving Text-to-Text Pre-trained Models for the Graph-to-Text Task". In *Proc. 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, edited by T. C. Ferreira et al., 107–116. Dublin, Ireland (Virtual): Association for Computational Linguistics.

Zhang, C., S. Bauer, P. Bennett, J. Gao, W. Gong, A. Hilmkil, J. Jennings, C. Ma, T. Minka, N. Pawlowski et al. 2023. "Understanding Causality with Large Language Models: Feasibility and Opportunities". *arXiv preprint arXiv:2304.05524; https://arxiv.org/abs/2304.05524*.

Zhang, F., M. Zhang, S. Liu, Y. Sun, and N. Duan. 2023. "Enhancing RDF Verbalization with Descriptive and Relational Knowledge". *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Zhou, J., P. Ke, X. Qiu, M. Huang, and J. Zhang. 2023. "ChatGPT: Potential, prospects, and limitations". *Frontiers of Information Technology & Electronic Engineering*:1–6.

Zhu, J.-J., J. Jiang, M. Yang, and Z. J. Ren. 2023. "ChatGPT and Environmental Research". *Environmental Science & Technology*.

Zong, J., C. Lee, A. Lundgard, J. Jang, D. Hajas, and A. Satyanarayan. 2022. "Rich Screen Reader Experiences for Accessible Data Visualization". In *Computer Graphics Forum*, Volume 41, 15–27. Wiley Online Library.

AUTHOR BIOGRAPHIES

PHILIPPE J. GIABBANELLI is an Associate Professor in the Department of Computer Science & Software Engineering at Miami University. His research interests include network science, machine learning, and simulation applied to human health behaviors, as reflected across 10 articles at the Winter Simulation conference. His research group has used Natural Language Generation tools for simulations starting with GPT-2 in 2021 and is now actively engaged in creating new solutions via GPT-4. He serves as general chair for the 2024 Annual Modeling and Simulation Conference (ANNSIM) and is an editor for five journals, including SIMULATION. His email address is giabbapj@miamioh.edu.