Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

AUTOMATICALLY EXPLAINING A MODEL: USING DEEP NEURAL NETWORKS TO GENERATE TEXT FROM CAUSAL MAPS

Anish Shrestha
Kyle Mielke
Tuong Anh Nguyen
Philippe J. Giabbanelli

Department of Computer Science & Software Engineering
Miami University
205W Benton Hall, 510 E. High St.
Oxford, OH 45056, USA

ABSTRACT

Simulation models start as conceptual models, which list relevant factors and their relationships. In complex socio-environmental problems, these conceptual models are routinely created with participants, via a 'participatory modeling' approach. Transparency is a tenet of participatory modeling: participants should easily provide their input into the model-building process and see how that input is utilized. Although several elicitation methods are transparent, the resulting conceptual model can become too large and difficult to interpret. Usability studies have shown that participants struggle to interact with such large conceptual models, even if they contributed to creating parts of it. In this paper, we propose to automatically transform these large conceptual models into a more familiar format for participants: textual reports. We designed and implemented a process combining Natural Language Generation (via the deep learning GPT-3 model) and Network Science. Two case studies demonstrate that our prototype generates sentences that perform satisfactorily on several metrics.

1 INTRODUCTION

The process of developing a simulation model starts with a *conceptual model*, which identifies relevant factors and relationships for the problem domain. These conceptual models can be small and well-known, for example as SIR or SEIR models in the context of infectious diseases. However, for complex socio-environmental problems, models can encompass dozens to hundreds of factors, and many more relationships. For instance, suicide (Chung 2016; Giabbanelli et al. 2021) or obesity conceptual models (McPherson et al. 2007; Drasic and Giabbanelli 2015) involve a very large number of protective or risk factors, across several domains (e.g., mental health, physical health, financial) and at several levels (e.g., individual, community, society). As they cross domain boundaries, such problems often involve teams of experts who work with modelers to arrive at a conceptual model. For example, participatory modeling approaches (Voinov et al. 2018) can elicit the knowledge of each expert in the form of a causal map (Bryson et al. 2004). This semi-quantitative approach externalizes the experts' perspectives as directed and signed graphs (Figure 1) and constitutes the starting point of expert-driven models (Kininmonth et al. 2021). Each causal map can be transparently elicited from an expert, either in person via a modeler trained as a facilitator (Kiekens et al. 2022), or online via software such as MentalModeler or automated procedures.

A problem happens as the individual perspectives of experts are combined into one conceptual model for the domain. These models tend to become very large and/or present complicated topologies (Giabbanelli and Baniukiewicz 2018), manifested by an abundance of loops and alternative paths (Figure 1). Although

individual causal maps are transparently elicited from experts, the aggregate model of the system is *no longer transparent* due to its sheer size and structure. Individuals often lack the systems thinking skills to work with the structure of these aggregate models (Gray et al. 2019; Giabbanelli and Tawfik 2020). Studies have shown that even custom network visualizations for node-link diagrams did not suffice in efficiently helping domain experts to navigate such unwieldy models (Giabbanelli et al. 2016). These large conceptual models are portrayed by domain experts (Siokou et al. 2014) as overly complex (emphasis added):

> "With 100 or so causal factors, and 300 or more connections linking each cause to one or more of the others, the [model] is a complicated, *almost incomprehensible web* of interconnectedness that depicts the drivers of obesity prevalence and the ways in which they depend on each other. The diagram is *brilliantly useful in demonstrating the complexity* of factors driving the current obesity trend, but the scale and number of interactions in the diagram make it *difficult to see how one might use it in any practical way* [...]"

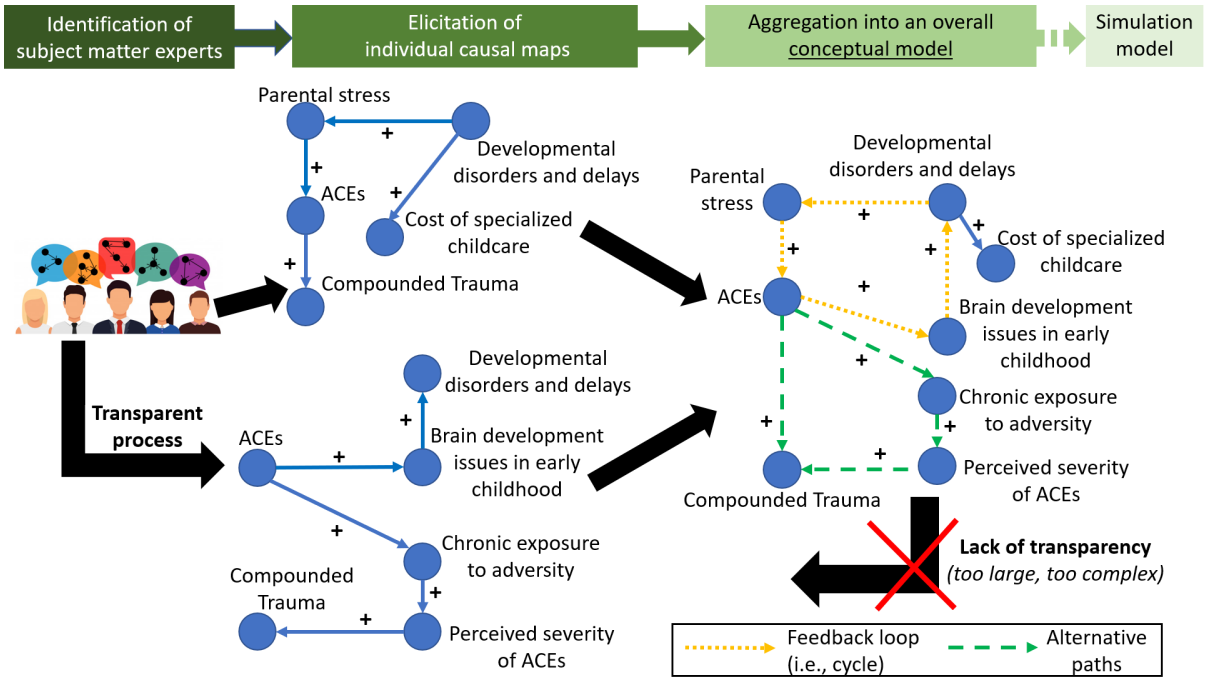

Figure 1: Two suicide experts provide a causal map on Adverse Childhood Experiences (ACEs). Each map is small and transparently obtained from an elicitation process. However, the final map aggregated the perspectives of all experts is larger and much more complicated (e.g., in feedback loops or alternative paths) than the sum of its individual maps, which prevents experts from navigating the model.

This is a significant communication obstacle, as it means that experts can *contribute to* a modeling project but are then unable to get *information back* from it. To promote a participatory approach to modeling complex socio-environmental problems and ensure a high level of involvement *throughout* the process (Hedelin et al. 2021), it is thus necessary to remove the obstacles that prevent domain experts from accessing the information encoded in the conceptual model. In this paper, we seek to make large conceptual maps transparent and accessible once again to experts. Specifically, we transform these maps into a more familiar and universal representation: text. At a high level, this objective resembles the model-to-text transformations (Rose et al. 2012) originating from Model-Driven Engineering. These text generation frameworks often use template-based code generators to produce application code from high level specifications. In contrast, our objective is to transform large causal maps into textual reports, since

domain experts are familiar with this format as the primary means of explaining a large system (Freund and Giabbanelli 2021). We thus use *Natural Language Generation* (NLG) to create text from causal maps, which involves both network analysis (to decompose the map) and deep learning (to generate text).

Our main contribution lies in the design, implementation, and evaluation of a system that effectively generates natural language from causal maps. Two case studies are provided on fields that have regularly yielded large maps, obesity and suicide. We briefly cover methods from text generation in Section 2, then build on them to explain the design and implementation of our system in Section 3. Results from our two case studies are presented in Section 4, while concluding remarks are provided in Section 5. For full transparency, our implementation and case studies are openly accessible at https://osf.io/98kmy/.

2 BACKGROUND: NATURAL LANGUAGE GENERATION (NLG) FROM DATA

2.1 From Traditional Methods to Deep Neural Networks

Data-To-Text refers to generating text from non-linguistic input data (Reiter 2007), such as tables, knowledge bases, or simulations. Significant methodological developments have happened in data-to-text generation (Puduppully and Lapata 2021). Traditional methods from the early 80's to early 2010's followed a rule-based system consisting of modules for content planning, sentence planning, and surface realization (Kukich 1983; Reiter and Dale 1997). Deep neural networks resulted in a paradigm shift in the mid 2010's. For example, Recurrent Neural Networks (RNNs) do not have separate modules on how to generate text that is faithful to the source material. RNNs have shown remarkable performances to generate short and detailed texts given a small data set (Wiseman et al. 2017). However, when given larger and more complicated inputs (such as the causal maps of interest in this work), neural network systems experience 'hallucinations' (i.e., reference errors), struggle to capture the relationship between sentences, and lack faithfulness to the source input (Wiseman et al. 2017).

In a simple *sequence-to-sequence architecture* for neural networks, an encoder transforms input sequences into a fixed-sized vector (the *context vector*), which is passed onto a decoder to generate the target sequence. However, the context vector in this architecture is a bottleneck when dealing with long sentences. The technique of *attention* was introduced so that the decoder can consider all hidden states of the encoder by learning to give them an attention score. The *transformer* architecture replaced the RNNs by attention and feed-forward layers for encoders and decoders. Transformers introduce a concept called *self-attention*, which allows to process each word in a sequence and attend to other words for better processing of the current word (Alammar 2018). These *transformers* rapidly became the dominant architecture in NLG.

2.2 Generative Pretrained Transformer (GPT)

Pretrained language models (PLMs) support a variety of tasks in Natural Language Processing, for example to classify sentiments via `BERT`. The purpose of PLMs is to learn language knowledge from a large corpus. This computationally intensive task is performed in a dedicated environment, then a PLM is transferred to another NLP task with task-specific datasets that are less computationally costly. PLMs based transformer architectures such as `GPT` (Alex Radford and Karthik Narasimhan and Tim Salimans and Ilya Sutskever 2018) or `T5` (Raffel et al. 2020) have achieved state-of-the-art performances because they benefit from the vast amount of factual language knowledge in their parameters (Harkous et al. 2020; Ribeiro et al. 2019). GPT-2 was released in November 2019 with 1.5 billion parameters, followed by the release of GPT-3 (used in this paper) in 2020 with 175 billion parameters.

PLMs do have limitations: they may generate texts that are not supported by the source input (Wang et al. 2021), or reproduce some of the typos on which they have been trained. It is thus necessary to thoroughly evaluate the performances of the system. Three metrics are commonly used to evaluate natural language generation outputs: BLEU (Papineni et al. 2002), METEOR (Denkowski and Lavie 2014), and chrF (Popović 2015). BLEU uses the notion of *n-gram*, that is, a contiguous subsequence of *n* words (e.g., for *n*=2 we have a 'bigram' consisting of a pair of words occurring next to each other). The BLEU

score is calculated by comparing the n-grams of the generated sentence with the n-grams of the reference sentence and counting the number of matches. METEOR scores generated sentences by aligning them with a reference sentence on the basis of exact, stem, synonym, and paraphrase matches (Denkowski and Lavie 2014). chrF, or character n-gram F-score, is calculated by comparing the character-level n-grams of the generated sentence with the character-level n-grams of the reference sentence. Each of these three scores is on a scale of 0 to 1, where 1 is a perfect match and 0 is a perfect mismatch. Automatic metrics are often criticized for only weakly reflecting human judgments (e.g., "BLEU–human correlations are poor for NLG" per Reiter 2018) hence they are commonly supplemented by a human evaluation of some generated texts. Three criteria are commonly applied: *faithfulness* (the extent to which the text preserve the *facts* in the input), *fluency* (*quality* of written text, including lexical skills, syntactic skills, and grammar skills), and *coverage* (the degree to which the *knowledge* in the input has been preserved); these measures are detailed in (Gatt and Krahmer 2018; Li et al. 2022).

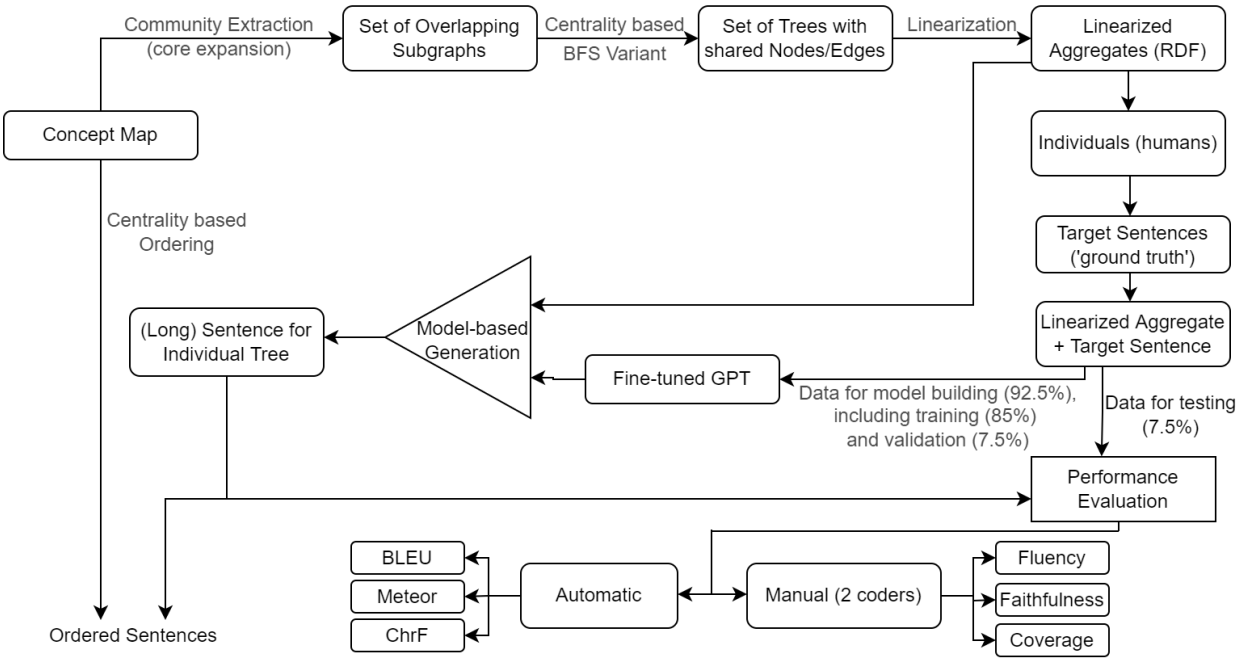

Figure 2: Our overall architecture to transform conceptual models (causal maps) into text.

3 METHODS

3.1 Overview

Our approach is summarized in Figure 2. In short, we start with a causal map that can contain a large number of factors on various topics, as well as complex structures such as loops. The ultimate goal of the system is to better understand the map it represents through text. If we convert the entire map into text by going through its parts in any order, there is a high chance that the theme of a sentence will be unrelated to the themes of adjacent sentences. Through community detection, we can take a large map and break it apart into smaller subsets that can be studied separately (Newman 2018, p. 551), each representing a single theme. In addition, smoothly connecting paragraphs requires that they have some elements in common, hence we focused on *overlapping community detection algorithms*. Each community is further decomposed into smaller fragments and *linearized* (Section 3.2). That is, there may be loops in a causal map but a text is only read left-to-right so any loop needs to be decomposed into fragments which can individually be

converted to text and, together, capture the content of the loop. Linearized representations are commonplace in recent works on graph-to-text representations (Ribeiro et al. 2020; Mager et al. 2020). Based on these representations, we prepare a corpus of well-formed and relevant sentences which serve to train and evaluate GPT-3 (Section 3.3). Once GPT-3 is fine-tuned, linearized representations are provided as inputs and the output is evaluated (Section 2.2) both automatically and manually, resulting in six performance indicators.

3.2 Map Transformation: Communities, Sub-parts, and Linearization

Algorithm 1 starts by using the Core Expansion method (Choumane et al. 2020) provided by CD-LIB (Rossetti et al. 2019) to create overlapping communities. Each community is then decomposed into a set of acyclic directed graphs, through a process akin to an iterated parametrized Breadth-First Search (BFS), which starts at a node and then explores its children in successive 'layers'. We start a modified BFS at the node with the most outgoing edges (i.e., in decreasing order of out-degree centrality), if it has not been visited too many times already. The modified BFS (Algorithm 2) has three parameters: a depth limit d, a width limit w, and a reuse threshold τ. The depth and width limits impact the length and eventually the complexity of the sentence. The depth limit prevents the search from going too many layers away, hence avoiding sentences that may appear to go on a tangent. The width limit prevents too many children of a node from being included, thus forbidding sentences formed of long lists (e.g., "an increase in A will increase B, C, ..., X"). Due to these two limits, a single run of the BFS can miss nodes in the input graph. Since Algorithm 1 repeatedly runs a BFS, nodes that were missing in one search would be included in another. Using the same nodes a few times is tolerable, for example to support the flow between sentences. However, an excessive level of reuse would result in an abnormal level of repetition in the sentences, hence a reuse threshold prevents a BFS from exceedingly re-visiting nodes. When the BFS algorithm adds a node to the output tree it increments a counter that is stored as an attribute of the node in the input graph. This is preserved between iterations of the search. If the search finds a node that has a counter exceeding the threshold, it does not include it in the output tree. Through experimentation, we set all three parameters to 3 as it mostly produced short, simple sentences, with adequate flow.

Algorithm 1 Decomposition of Causal Maps

Input: Causal graph (labeled nodes, directed typed edges); τ: maximum number of times that a node can be reused; d and w: depth and width of the breadth-first search

Output: Set of communities decomposed into smaller acyclic graphs

```
Find community subgraphs by the Core Expansion method
Determine and mark the out-degree centrality of nodes
Create an empty set S for results
for each community subgraph
    Create an empty set C for the decomposition of this community
    Set every nodes use counter to 0
    for each node i in subgraph in order of centrality
        if the node's use count < τ then
            Run BFS(d,w,i) and increment the counter of each visited node
            Add the graph from the BFS to C
    Add results C for the community to the overall results S
Return S
```

3.3 Evaluating and Fine-Tuning GPT-3

In order to evaluate performances (i.e., *testing*), it is common practice to compare the output of an NLG system with *human-authored narratives* (Ahn et al. 2016). In addition, a *pretrained language model*

Algorithm 2 Parameterized Breadth First Search

Input: Same as Algorithm 1 (causal graph, τ, d and w)
Output: One acyclic graph

```
start_node                                  ▷ Pick a start node chosen according to order from centrality index
start_node.depth = -1
search_queue.push(start_node)               ▷ Initialize search queue with the start node
start_node.parent = start_node              ▷ Initialize itself as the parent
start_node.use_counter += 1
while search_queue is not empty
    node = search_queue.pop()
    node.depth = node.parent.depth + 1      ▷ Set the node's depth to one more than its parent
    if node.depth > d then break
    for the first w neighbors of the node
        if neighbor.parent is null and neighbor.use_counter < τ then
            neighbor.parent = node          ▷ Mark the current node as its parent
            neighbor.use_counter += 1       ▷ Increment the neighbor's use counter
            outgraph.add_edge(node, neighbor)
            search_queue.push(neighbor)
```

(Section 2.2) such as GPT-3 can be fine-tuned with such narratives to capture domain specific knowledge. Note that *fine-tuning* differs from *re-training* in two ways: creating enough human-authored narratives for training can be extremely time consuming and the training itself is computationally intensive; in contrast, fine-tuning requires fewer instances and much less computations. In the case of GPT-3, a few hundred examples suffice, as the model is efficient on low-shot learning (Chintagunta et al. 2021). Our approach thus includes the creation of a few hundred human-authored narratives for the dual purpose of evaluating and fine-tuning the generator. This process is performed in several steps, starting from the linearized aggregates (Figure 2 top-right) obtained in the previous section.

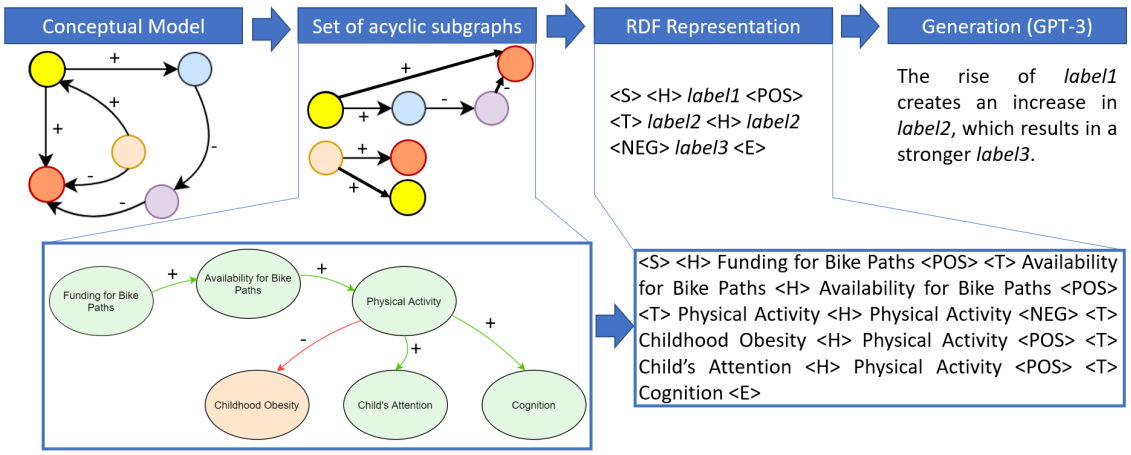

Figure 3: Each acyclic graph is coded by stringing RDF (Resource Description Framework) triples, which are used in datasets for natural language generation such as WebNLG (Gardent et al. 2017). <H>, <T>, <NEG>, and <POS> tags encode the source (head), destination (tail), negative, and positive causality.

For a given conceptual model, we first transform it into a set of linearized aggregates, represented as RDF triples (Figure 3). Human annotators write a sentence for each of the linearized aggregates, such that every aggregate is handled by at least two independent annotators. This produces pairs of linearized representations

and target sentences, such as {"prompt":"<S> <H> Mental well-being <POS> <T> Physical well-being <E>","completion":" Mental well-being promotes physical well-being.<end>"}. OpenAI recommends using special tokens to start and end prompts (linearized representations) and completions (target sentences), hence our sentences always start with a space and end with *<end>*.

As mentioned above, *some* of this data will fine-tune the model and the rest serves to evaluate the output. In machine learning, most of the data goes toward building the model and a lesser amount is devoted to evaluating it. The data for model building is further subdivided into 'training' and 'validating' sets; a 'validation' set serves to adjust hyper-parameter values and should not be misinterpreted as evaluation data for the final model, which is accomplished by the 'testing' set. This approach is known as a 'training-validation-test split' and constitutes standard practice in machine learning (Duan et al. 2022). In our work, we use a ratio of 80:12:8 for training, validating, and testing respectively.

OpenAI offers four models of different sizes and capabilities for training GPT-3, namely Davinci, Curie, Babbage, and Ada (OpenAI 2021). To fine-tune, we compared performances between the largest model (Davinci) and the second largest (Curie); due to space limitation, our comparisons (3 pages) are provided as a supplementary online table on the Open Science Framework at https://osf.io/98kmy/. Curie offered similar performances when training on the same dataset at an approximately 8 fold reduced price, hence all our case studies employ the Curie model. In deep learning, training is specified in *epochs*, which are the number of times that the entire dataset has been used. To determine a sufficient number of epochs, we identify when the training performance outcome (i.e., training loss) reaches a steady state. In our case studies, Curie was trained for five epochs, which is when the training loss starts plateauing.

3.4 Evaluation Metric

We used `NLTK` 3.6.5 to perform the three automatic comparisons (BLEU, chrF, METEOR) between generated sentences and the testing. Human evaluation was performed independently by two annotators in two steps. First, they independently evaluated each sentence with respect to each metric (faithfulness, fluency, and coverage) on a 5-point Likert scale, from (1) poor to (5) excellent. Second, the scores were compared and all discrepancies were discussed to arrive at a consensual score. This final score is reported in our case studies below and, for full disclosure, our repository at https://osf.io/98kmy/ offers the scores of each annotators together with their notes on identifying and addressing discrepancies.

4 CASE STUDIES

We evaluated our system on two studies that created an open-access conceptual model: an obesity conceptual model of 98 nodes and 177 edges (Drasic and Giabbanelli 2015), and a conceptual model of youth suicide with 361 nodes and 948 edges (Giabbanelli et al. 2021). Both studies had several subject-matter experts, each producing a map that is small and interpretable, but the aggregate map is large and complex. This situation suffers from a lack of transparency and motivates our work in transforming models to text.

The conceptual models are first decomposed into linearized aggregates (Section 3.2), then we wrote target sentences for each aggregate and performed the training-validation-test split (Section 3.3). For obesity, this created a total of 625 sentences, of which 521 sentences were for training, 58 sentences on validation, and 46 sentences on testing. For suicide, we had 494 sentences for training, 48 sentences for validation, 46 sentences for testing, hence a total of 588 sentences. We then evaluated our system (Section 3.4) with respect to automatic and manual metrics, yielding the results in Table 1 and Table 2 respectively. The $P45$ metric is the ratio of human scores that are 4 or 5 (on a scale of $1-5$) out of the total number of scores (Mager et al. 2020). For instance, if three evaluators scored the fluency, coverage, and faithfulness as $\{3,4,4\}$, $\{3,3,2\}$, and $\{4,5,3\}$ then the $P45$ would be 4 out of 9.

Scores for BLEU are similar to previous studies where few ground-truth sentences are available for each sample (Qader et al. 2018), since BLEU is driven by n-gram matches between the candidate and reference sentences. In contrast, METEOR is more tolerant of variation in language, as it aligns words

between the candidate and reference sentences by allowing for exact matching, synonymy, and stemming. Scores for METEOR match or exceed previous studies on neural text generation (Qader et al. 2019). Similarly, the chrF score also indicates a decent performance.

Since there is evidence that the size of the input graph (in number of nodes) impacts the quality and length of the sentence generated (Ribeiro et al. 2020), we detailed the results of the human evaluation based on three sizes: small (2–3 nodes), medium (4–5 nodes), or large (6 nodes and above). We note that faithfulness and coverage decrease as the size of the input increases, which means that GPT-3 is best able at capturing the whole content (coverage) and preserving its meaning (faithfulness) when the conceptual model has been decomposed into *small* subsets. Fluency is only affected in larger graphs, as the longer sentences generated are more at risk of accumulating errors. Overall, our approach can create texts that are as comprehensive and factual as the initial conceptual model, if it is finely divided into small parts.

Table 1: Average automatic evaluation scores (BLEU, METEOR, chrF) for text generation two conceptual models. Each score ranges from 0 (mismatch) to 1 (match).

Case study	BLEU	METEOR	chrF
Obesity	0.028	0.601	0.633
Suicide	0.023	0.662	0.683

Table 2: Average human evaluation scores and ratio of sentences evaluated 4 and 5 (P45) for text generation from two conceptual models. Each score ranges from 1 (bad) to 5 (excellent).

Case study	Nodes	Faithful	Fluency	Coverage	P45
Obesity	(2,3)	4.84	4.72	4.92	0.94
	(4,5)	4.57	4.76	4.78	
	6+	3.67	4.00	3.67	
Suicide	(2,3)	4.78	4.47	5.00	0.93
	(4,5)	4.42	4.50	5.00	
	6+	3.86	4.57	3.71	

To better characterize the sentences generated by GPT-3 in both case studies, Table 3 provides several examples of correct and incorrect sentences, for each of the two case studies. On the positive side, we observe that the fine-tuned GPT-3 model can aptly rename concepts (e.g., 1st example: breast feeding), learn commas in sentences (e.g., 2nd example), creates a variety of causal connectors (e.g., increase, improve, higher, more), and even simplifies causality (e.g., two negatives make a positive) to create fluid sentences. On the negative side, the data used for pre-training of GPT-3 contained typos which resurface in generated sentences (e.g., corrisponds, suicidial), subtlety can be lost (e.g., 6th example), causal branches can be an issue (Figure 4), and some modeling features cannot currently be translated. That is, some modeling features can be binary such as whether an agent has a given ethnicity, or whether the agent's social norm is the main one. Since most of the map deals with an *increase* in source concepts rather than the *presence* of certain concepts, binary and other categorical features are currently a limitation (e.g., last example). Although the problem of 'hallucination' (i.e., outputting ideas that are nowhere in the input) is often discussed in natural language generation (Wang et al. 2021; Wiseman et al. 2017), we only observed this issue in one out of 84 cases. It happened in the obesity case study, where GPT-3 added a clause (emphasis added): "Physical activity is an important factor for good health, *however it can not give anyone superhuman powers.*" Although such 'silly mistakes' can receive more than their fair share of attention, this rare hallucination is a lesser concern than limitations on modeling features at present.

The strength of our results partly relies on the latest version of GPT, called GPT-3. We performed complementary analyses by contrasting our current results with those obtained from GPT-2; the difference

Table 3: Examples of correct and incorrect sentences. All sentences are provided on our repository.

Case study	**Input** (part of the conceptual model)	**Text Generated**
Obesity: Correct	<S><H>Quality & Quantity of breast feeding <POS><T>Appropriateness of child growth <H>Quality & Quantity of breast feeding <POS><T>Appropriateness of nutrient partitioning <E>	Breast feeding of higher quality and more quantity leads to more appropriate child growth and nutrient partitioning.
	<S><H>calories <POS><T>obesity <E>	Higher the calorie intake, the more obese a person gets.
	<S><H>promote safe routes to encourage walking <NEG><T>unsafe neighborhoods <E>	Increasing safe routes to encourage walking reduces unsafe neighborhoods.
Obesity: Incorrect	<S><H>weight <POS><T>BMI <E>	Increase in weight corrisponds to increase in BMI.
	<S><H>Unhealthy weight control practices <NEG><T>Obesity <H>Obesity <POS> <T>Weight bias <H>Obesity <NEG><T> Access to health professionals <H>Weight bias <NEG><T>Positive experience of sports <E>	Increasing unhealthy weight control practices reducings obesity, thereby increasing weight bias and reduces access to health professionals. This increase to weight bias reduces the positive experience of sports.
	<S><H>Effort to increase efficiency of consumption <POS><T>Demand for convenience <E>	Effort to increase efficiency of consumption increases the convenience.
Suicide: Correct	<S><H>Low socio-economic status of the neighbourhood <NEG><T>Access to healthcare <H>Low socio-economic status of the neighbourhood <POS><T> Involvement in violence <E>	Socio-economic status of the neighbourhood can have a direct impact on access to healthcare and also involvement in violence.
	<S><H>Policies on responsible suicide awareness <NEG><T>Imitating a suicidal behavior <E>	Improving policies on responsible suicide awareness decreases the tendency to imitate a suicidal behavior.
	<S><H>Engaging in externalizing behaviors <POS><T>Getting labeled at school <E>	Engaging in externalizing behaviors gets one labeled at school.
Suicide: Incorrect	<S><H>Suicide ideation <POS><T>Visit of ER <H>Visit of ER <POS><T>Receive care at home <H>Visit of ER <POS><T> Hospitalized <H>Hospitalized <POS><T> Treatment success <E>	Increase in suicidial ideation increases the visit of ER which further increases the chance of receiving care at home and hargens hospitalized. This increase to hospitalized increases treatment success.
	<S><H>Identity is outside the socially accepted norm in the community <POS> <T>Bullying <H>Bullying <POS><T> Abusive relationships with peers <H> Bullying <POS><T>ACEs <H>Bullying <POS><T>Emotional breakup and rejection <H>ACEs <POS><T>Shame <E>	Increasing the identity of outsiders increases bullying, thereby increasing abusive relationships with peers, ACEs, emotional breakup and rejection, and shame.

is available on our repository at https://osf.io/98kmy/. GPT-2 was about two points lower on faithfulness and coverage than GPT-3, while automatic scores were about half of those obtained on GPT-3.

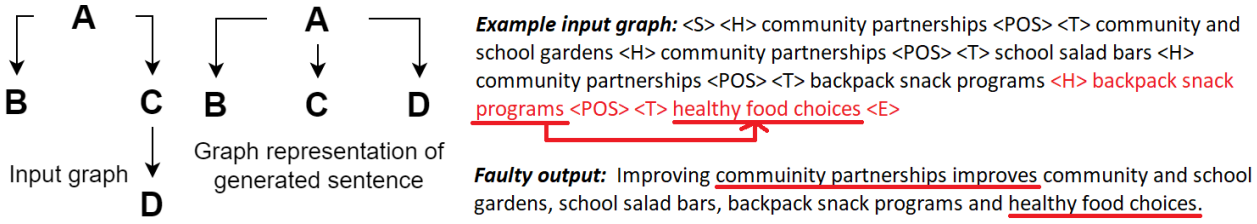

Figure 4: Some oversimplifications are associated with specific input structures.

5 DISCUSSION AND CONCLUSION

We proposed a process to transform a large conceptual model (in the form of a causal map) into sentences, by decomposing it into smaller parts and then performing Natural Language Generation (NLG) via a fine-tuned GPT-3 (Figure 2). Automatic metrics that tolerate variations in language show encouraging performances on two case studies (Table 1). Manual evaluation emphasizes the strong relationship between the decomposition algorithm and the ensuing NLG step, as the best sentences are obtained when the conceptual model is decomposed into small parts. By examining the sentences (Table 3), we noted several strengths. For example, the variety of causal connectors helps to avoid boring, repetitive structures. This follows the recommendations of Ahn, Morbini, and Gordon (2016): "when serializing and realizing a causal graph structure as natural language text, some care must be taken to avoid the generation of cumbersome sentences with repetitive syntactic structure, e.g. as a long chain of 'because' clauses." We also noted several shortcomings. Some of them can be addressed by avoiding certain input structures in the decomposition algorithm (Figure 4), and some may be handled by an additional autocorrect filter on the output.

Two avenues are of particular interest for future work. First, it was necessary to create a ground-truth dataset for evaluation by handcrafting sentences, which also serve to fine-tune the model. However, when we focus on deployment, the process should be entirely automatized. Natural Language Processing will thus be needed to automatically creating sentences for fine-tuning. Second, by deploying the prototype, participants can provide feedback on the text. This feedback can identify errors, which can be used to improve the model. In particular, information can be injected by editing the prompt for each input graph, thus helping the model to make corrections without needing either re-training or fine-tuning. This notion of 'prompt engineering' (Liu et al. 2021) was successfully used with GPT-3 in (Madaan et al. 2022) and is particularly promising to improve products once they are deployed.

REFERENCES

Ahn, E., F. Morbini, and A. Gordon. 2016. "Improving Fluency in Narrative Text Generation With Grammatical Transformations and Probabilistic Parsing". In *Proceedings of the 9th International Natural Language Generation Conference*, 70–73. Association for Computational Linguistics.

Jay Alammar 2018. "The Illustrated Transformer". https://jalammar.github.io/illustrated-transformer/, accessed 21st September 2022.

Bryson, J., F. Ackermann, C. Eden, and C. Finn. 2004. *Visible Thinking: Unlocking Causal Mapping for Practical Business Results*. Chichester, UK: Wiley.

Chintagunta, B., N. Katariya, X. Amatriain, and A. Kannan. 2021. "Medically Aware GPT-3 as a Data Generator for Medical Dialogue Summarization". In *Proceedings of the 6th Machine Learning for Healthcare Conference*, 354–372. Proceedings of Machine Learning Research.

Choumane, A., A. Awada, and A. Harkous. 2020. "Core Expansion: A New Community Detection Algorithm Based on Neighborhood Overlap". *Social Network Analysis and Mining* 10:1–11.

Chung, S. Y. 2016. *Suicide Attempts from Adolescence into Young Adulthood: A System Dynamics Perspective for Intervention and Prevention*. Ph.D. thesis, Washington University in St. Louis, St. Louis, MO, USA. https://openscholarship.wustl.edu/art_sci_etds/722/, accessed 21st September 2022.

Denkowski, M., and A. Lavie. 2014. "Meteor Universal: Language Specific Translation Evaluation for Any Target Language". In *Proceedings of the 2014 Workshop on Statistical Machine Translation*. Association for Computational Linguistics.

Drasic, L., and P. J. Giabbanelli. 2015. "Exploring the Interactions Between Physical Well-Being, and Obesity". *Canadian Journal of Diabetes* 39:S12–S13.

Duan, J., S. Yu, H. L. Tan, H. Zhu, and C. Tan. 2022. "A Survey of Embodied AI: From Simulators to Research Tasks". *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Freund, A. J., and P. J. Giabbanelli. 2021. "Are We Modeling the Evidence or Our Own Biases? A Comparison of Conceptual Models Created from Reports". In *Proceedings of the 2021 Annual Modeling and Simulation Conference*, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gardent, C., A. Shimorina, S. Narayan, and L. Perez-Beltrachini. 2017. "The WebNLG Challenge: Generating Text from RDF Data". In *Proceedings of the 10th International Conference on Natural Language Generation*, 124–133. Association for Computational Linguistics.

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.

Giabbanelli, P., R. Flarsheim, C. Vesuvala, and L. Drasic. 2016. "Developing Technology to Support Policymakers in Taking a Systems Science Approach to Obesity and Well-being". *Obesity Reviews* 17:194–195.

Giabbanelli, P. J., and M. Baniukiewicz. 2018. "Navigating Complex Systems for Policymaking Using Simple Software Tools". In *Advanced Data Analytics in Health*, edited by P. J. Giabbanelli, V. K. Mago, and E. I. Papageorgiou, 21–40. Cham: Springer International Publishing.

Giabbanelli, P. J., M. C. Galgoczy, D. M. Nguyen et al. 2021. "Mapping the complexity of suicide by combining participatory modeling and network science". In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, edited by M. Coscia, A. Cuzzocrea, K. Shu, R. Klamma, S. O'Halloran, and J. Rokne, 339–342. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Giabbanelli, P. J., and A. A. Tawfik. 2020. "Reducing the Gap Between the Conceptual Models of Students and Experts Using Graph-Based Adaptive Instructional Systems". In *HCI International 2020 – Late Breaking Papers: Cognition, Learning and Games*, edited by C. Stephanidis, D. Harris, W.-C. Li, et al., 538–556. Cham: Springer International Publishing.

Gray, S., E. J. Sterling, P. Aminpour, L. Goralnik, A. Singer, C. Wei, S. Akabas, R. C. Jordan, P. J. Giabbanelli, J. Hodbod et al. 2019. "Assessing (Social-Ecological) Systems Thinking by Evaluating Cognitive Maps". *Sustainability* 11(20):5753.

Harkous, H., I. Groves, and A. Saffari. 2020, December. "Have Your Text and Use It Too! End-to-End Neural Data-to-Text Generation with Semantic Fidelity". In *Proceedings of the 28th International Conference on Computational Linguistics*, 2410–2424. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Hedelin, B., S. Gray, S. Woehlke et al. 2021. "What's Left Before Participatory Modeling Can Fully Support Real-world Environmental Planning Processes: A Case Study Review". *Environmental Modelling & Software* 143:105073.

Kiekens, A., B. Dierckx de Casterlé, and A.-M. Vandamme. 2022. "Qualitative Systems Mapping for Complex Public Health Problems: A Practical Guide". *PloS One* 17(2):e0264463.

Kininmonth, S., S. Gray, and K. Kok. 2021. *Expert Modelling*, 231–240. Oxon UK & New York NY USA: Taylor and Francis.

Kukich, K. 1983. "Design of a Knowledge-Based Report Generator". In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 145–150. Association for Computational Linguistics.

Li, W., W. Wu, M. Chen et al. 2022. "Faithfulness in Natural Language Generation: A Systematic Survey of Analysis, Evaluation and Optimization Methods". https://arxiv.org/abs/2203.05227, accessed 21st September 2022.

Liu, P., W. Yuan, J. Fu et al. 2021. "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". *arXiv preprint arXiv:2107.13586*. https://arxiv.org/abs/2107.13586, accessed 21st September 2022.

Madaan, A., N. Tandon, P. Clark, and Y. Yang. 2022. "Memory-Assisted Prompt Editing to Improve GPT-3 After Deployment". *arXiv preprint arXiv:2201.06009*. https://arxiv.org/abs/2201.06009, accessed 21st September 2022.

Mager, M., R. F. Astudillo, T. Naseem, M. A. Sultan, Y.-S. Lee, R. Florian, and S. Roukos. 2020. "Gpt-too: A Language-Model-First Approach For AMR-To-Text Generation". *arXiv preprint arXiv:2005.09123*. https://arxiv.org/abs/2005.09123, accessed 21st September 2022.

McPherson, K., T. Marsh, and M. Brown. 2007. "Foresight Report on Obesity". *The Lancet* 370(9601):1755.

Newman, M. 2018. *Networks*. Oxford, UK: Oxford University Press.

OpenAI 2021. "Customizing GPT-3 for Your Application". https://openai.com/blog/customized-gpt-3/, accessed 26th February 2022.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. "Bleu: a Method for Automatic Evaluation of Machine Translation". In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Association for Computational Linguistics.

Popović, M. 2015, September. "chrF: Character N-Gram F-Score for Automatic MT Evaluation". In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392–395. Lisbon, Portugal: Association for Computational Linguistics.

Puduppully, R., and M. Lapata. 2021. "Data-to-text Generation with Macro Planning". *CoRR* abs/2102.02723. https://arxiv.org/abs/2102.02723, accessed 21st September 2022.

Qader, R., K. Jneid, F. Portet, and C. Labbé. 2018. "Generation of Company Descriptions Using Concept-to-Text and Text-to-Text Deep Models: Dataset Collection and Systems Evaluation". In *Proceedings of the 11th International Conference on Natural Language Generation*. Association for Computational Linguistics.

Qader, R., F. Portet, and C. Labbé. 2019. "Semi-Supervised Neural Text Generation by Joint Learning of Natural Language Generation and Natural Language Understanding Models". https://arxiv.org/abs/1910.03484, accessed 21st September 2022.

Alex Radford and Karthik Narasimhan and Tim Salimans and Ilya Sutskever 2018. "Improving Language Understanding by Generative Pre-Training". https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf, accessed 21st September 2022.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu et al. 2020. "Exploring the Limits of Transfer Learning With a Unified Text-To-Text Transformer". *Journal of Machine Learning Research* 21(140):1–67.

Reiter, E. 2007. "An Architecture for Data-to-Text Systems". In *Proceedings of the 11th International Conference on Natural Language Generation.*, 97–104. Association for Computational Linguistics.

Reiter, E. 2018. "A Structured Review of the Validity of BLEU". *Computational Linguistics* 44(3):393–401.

Reiter, E., and R. Dale. 1997. "Building Applied Natural Language Generation Systems". *Natural Language Engineering* 3(1):57–87.

Ribeiro, L. F., M. Schmitt, H. Schütze, and I. Gurevych. 2020. "Investigating Pretrained Language Models for Graph-to-Text Generation". *arXiv preprint arXiv:2007.08426*. https://arxiv.org/abs/2007.08426, accessed 21st September 2022.

Ribeiro, L. F., Y. Zhang, C. Gardent, and I. Gurevych. 2020. "Modeling Global and Local Node Contexts for Text Generation From Knowledge Graphs". *Transactions of the Association for Computational Linguistics* 8:589–604.

Ribeiro, L. F. R., C. Gardent, and I. Gurevych. 2019, November. "Enhancing AMR-to-Text Generation with Dual Graph Representations". In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 3183–3194. Hong Kong, China: Association for Computational Linguistics.

Rose, L. M., N. Matragkas, D. S. Kolovos, and R. F. Paige. 2012. "A Feature Model for Model-to-Text Transformation Languages". In *Proceedings of the 4th International Workshop on Modeling in Software Engineering*, edited by J. Atlee, R. Baillargeon, R. France, et al., 57–63. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rossetti, G., L. Milli, and R. Cazabet. 2019. "CDLIB: a Python Library to Extract, Compare and Evaluate Communities From Complex Networks". *Applied Network Science* 4(1):1–26.

Siokou, C., R. Morgan, and A. Shiell. 2014. "Group Model Building: a Participatory Approach to Understanding and Acting on Systems". *Public Health Research & Practice* 25(1):e2511404.

Voinov, A., K. Jenni, S. Gray, N. Kolagani, P. D. Glynn, P. Bommel, C. Prell et al. 2018. "Tools and Methods in Participatory Modeling: Selecting the Right Tool for the Job". *Environmental Modelling & Software* 109:232–255.

Wang, Q., S. Yavuz, V. Lin, H. Ji, and N. F. Rajani. 2021. "Stage-Wise Fine-tuning for Graph-to-Text Generation". *ArXiv* abs/2105.08021. https://arxiv.org/abs/2105.08021, accessed 21st September 2022.

Wiseman, S., S. Shieber, and A. Rush. 2017, September. "Challenges in Data-to-Document Generation". In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2253–2263. Copenhagen, Denmark: Association for Computational Linguistics.

AUTHOR BIOGRAPHIES

ANISH SHRESTHA is a graduate student in the department of Computer Science and Software Engineering at Miami University, where he is advised by Dr Giabbanelli on NLG and virtual reality. His email address is shresta3@miamioh.edu.

KYLE MIELKE is a graduate student in the department of Computer Science and Software Engineering at Miami University. His email address is mielkek@miamioh.edu.

TUONG ANH NGUYEN is an undergraduate student in the department of Computer Science and Software Engineering at Miami University. Her email address is nguyen47@miamioh.edu.

PHILIPPE J. GIABBANELLI is an Associate Professor of Computer Science & Software Engineering at Miami University. He holds a Ph.D. from Simon Fraser University. He has over 100 publications, primarily on Modeling & Simulation and Machine Learning. He is an associate editor for five journals, including SIMULATION. His email address is giabbapj@miamioh.edu.