Counterfactual Token Generation in Large Language Models

Ivi Chatzi[*,§], **Nina Corvelo Benz**[*,§,†], **Eleni Straitouri**[*,§],
Stratis Tsirtsis[*,§] **and Manuel Gomez-Rodriguez**[§]
[§]MPI-SWS [†]ETH Zurich [*]Equal contribution
{ichatzi,ninacobe,estraitouri,stsirtsis,manuelgr}@mpi-sws.org

Abstract

"Sure, I am happy to generate a story for you: Captain Lyra stood at the helm of her trusty ship, the Maelstrom's Fury, gazing out at the endless sea. [...] Lyra's eyes welled up with tears as she realized the bitter truth – she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself." Although this story, generated by a large language model, is captivating, one may wonder—how would the story have unfolded if the model had chosen "Captain Maeve" as the protagonist instead? We cannot know. State-of-the-art large language models are stateless—they maintain no internal memory or state. Given a prompt, they generate a sequence of tokens as an output using an autoregressive process. As a consequence, they cannot reason about counterfactual alternatives to tokens they have generated in the past. In this work, our goal is to enhance them with this functionality. To this end, we develop a causal model of token generation that builds upon the Gumbel-Max structural causal model. Our model allows any large language model to perform counterfactual token generation at almost no cost in comparison with vanilla token generation, it is embarrassingly simple to implement, and it does not require any fine-tuning nor prompt engineering. We implement our model on Llama 3 8B-instruct and conduct qualitative and quantitative analyses of counterfactually generated text. We conclude with a demonstrative application of counterfactual token generation for bias detection, unveiling interesting insights about the model of the world constructed by large language models.

We encourage the reader to refer to the full version of the paper at https://arxiv.org/abs/2409.17027.

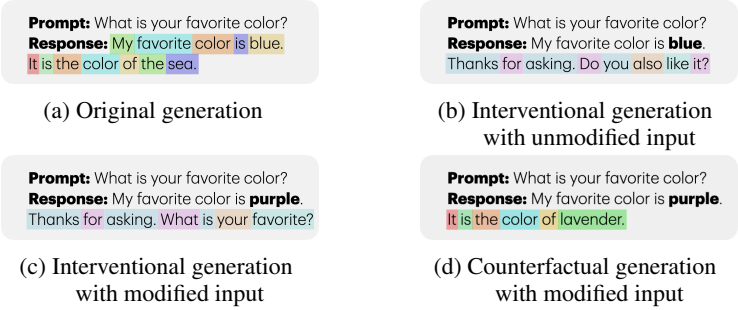
1 Introduction

Reasoning about "what might have been", about alternatives to our own past actions, is a landmark of human intelligence [1–3]. This type of reasoning, known as counterfactual reasoning, has been shown to play a significant role in the ability that humans have to learn from limited past experience and improve their decision making skills over time [4–6], it provides the basis for creativity and insight [7], and it is tightly connected to the way we attribute causality and responsibility [8–11]. Can currently available large language models (LLMs) conduct counterfactual reasoning about alternatives to their own outputs? In this work, we argue that they cannot, by design.

Currently available LLMs are stateless—they maintain no internal memory or state. Given an input prompt, they generate a sequence of tokens[1] as output using an autoregressive process [12, 13]. At each time step, they first use a neural network to map the prompt and the (partial) sequence of tokens generated so far to a token distribution. Then, they use a sampler to draw the next token at random from the token distribution.[2] Finally, they append the next token to the (partial) sequence of tokens,

[1]Tokens are the units that make up text, such as (sub-)words, symbols, and special end-of-sequence tokens.
[2]Evidence suggests that, if an LLM is forced to output tokens deterministically, its performance worsens [14].

NeurIPS 2024 Workshop on Causality and Large Models (CaLM).



(a) Original generation

(b) Interventional generation with unmodified input

(c) Interventional generation with modified input

(d) Counterfactual generation with modified input

Figure 1: **Illustrative examples of autoregressive token generation.** In all panels, plain text indicates the input provided to the LLM and highlighted text indicates the output generated by the model. Each token in the output sequence is highlighted in a different color to represent the (stochastic) state of the sampler. Panel (a) shows an LLM's output to a user's prompt using vanilla autoregressive token generation. Panels (b, c) show an LLM's output to an input comprising a user's prompt and an unmodified/modified part of the original output from Panel (a) using vanilla autoregressive token generation. Panel (d) shows an LLM's counterfactual output to an input comprising a user's prompt and a modified part of the output from Panel (a) using autoregressive token generation augmented with the Gumbel-Max SCM.

and continue until a special end-of-sequence token is sampled. To understand why this autoregressive process is insufficient to reason counterfactually about alternatives to a previously generated sequence of tokens, we will use an illustrative example.

Consider that we ask an LLM to share its favorite color, as shown in Figure 1a. Had the LLM chosen a different color (*i.e.*, purple instead of blue), what would the rest of its output have been? To answer such a counterfactual question, we need to implement two actions: (i) modify the (partial) sequence of tokens fed to the neural network used by the LLM and (ii) compel the sampler used by the LLM to *behave exactly as it did* in the original generation. Using currently available LLMs, we can readily implement the first action, which can be viewed as a causal intervention [15, 16], by replacing "blue" with "purple" in the (partial) sequence of tokens fed to the neural network. However, we cannot easily implement the second action, because the sampler does not specify how it *would have behaved* after taking the first action *while keeping everything else equal*. In fact, note that, if we provide the (modified) partial sequence up to and including the world "blue" ("purple") as input to the LLM, there is no way to ensure that the LLM will generate an output that matches (the structure of) the original output because the (stochastic) state of the sampler is different, as shown in Figures 1b and 1c.[3]

Our contributions. Our key idea is to augment the autoregressive process underpinning an LLM, particularly the sampler used in the process, using the Gumbel-Max structural causal model (SCM) [17]. Under this model, the sampler is defined through a causal mechanism which receives as an input the distribution of the next token and a set of Gumbel noise values. Importantly, this causal mechanism specifies how the sampler would have behaved under an intervention on the distribution of the next token and thus allow us to answer counterfactual questions about a previously generated sequence of tokens, as shown in Figure 1d. Along the way, we also introduce an efficient implementation of the augmented autoregressive process that can generate counterfactual tokens at almost no cost in comparison with vanilla token generation. As a proof of concept, we implement our model on Llama 3 8B-instruct, and conduct experiments to qualitatively and quantitatively analyze the similarity between an LLM's original output and the one generated via counterfactual token generation. Additionally, we demonstrate the use of our methodology for bias detection, unveiling interesting insights about the model of the world constructed by large language models. An open-source implementation of our model on Llama 3 8B-instruct is available at `https://github.com/Networks-Learning/counterfactual-llms`.

Further related work. Our work is most closely related to a line of work on counterfactual text generation [18–27]. In this line of work, given pairs of factual statements and interventions over these statements, the goal is to generate counterfactual statements that match those made by humans—counterfactual statements that are consistent with the underlying model of the world shared by

[3]Note that using the same random seed is not sufficient because the inputs in Figure 1a and Figures 1b and 1c differ in their number of tokens.

humans. To this end, existing methods typically fine-tune an LLM using a dataset comprising factual statements, interventions over these statements, and counterfactual statements made by humans. In contrast, in our work, our goal is to generate counterfactual statements that are consistent with the underlying model of the world constructed by a given LLM [28–31]. In this context, our work also relates to a rapidly increasing number of empirical studies assessing the ability of LLMs to answer questions that require counterfactual reasoning [32–43]. Here, the LLMs are typically evaluated using multiple choice questions about a given set of factual and counterfactual statements. However, similarly as in the line of work on counterfactual text generation discussed previously, the counterfactual statements are made by humans.

The Gumbel-max structural causal model has been previously used to enable counterfactual reasoning in MDPs [44], temporal point processes [45], and expert predictions [46]. However, to the best of our knowledge, it has not been previously used to enable counterfactual reasoning in LLMs.

2 A Causal Model of Token Generation

To formally express autoregressive token generation, we adopt (part of) the notation introduced by Duetting et al. [47] in a different (non-causal) context. Let V denote the vocabulary (set) of tokens available to the LLM, which includes an end-of-sequence token $\perp$. Then, we denote by $V^* = V \cup V^2 \cup \cdots \cup V^K$ the set of sequences of tokens up to maximum length K, and by $\varnothing$ the empty token. An LLM takes as input a prompt sequence $s_q \in V^*$ and responds with an output sequence $s \in V^*$. The output sequence is generated using an autoregressive process. At each time step $i \in [K]$, the LLM first takes as input the concatenation of the prompt sequence s_q and the (partial) output sequence s_{i-1} and generates a distribution over tokens $d_i \in \Delta(V)$. Then, it samples the next token $t_i \sim d_i$ from the distribution d_i and creates the output sequence $s_i = s_{i-1} \circ t_i$, where $\circ$ denotes the concatenation of a token or sequence with another sequence. Further, if $t_i = \perp$, it terminates and returns $s = s_i$, otherwise it continues to the next step $i + 1$ in the generation.

Given any prompt sequence, the above autoregressive process determines what (factual) output sequence the LLM generates as a response. However, given a generated output sequence, the above process does not determine what counterfactual output sequence the LLM would have generated if the prompt sequence, or some of the tokens in the output sequence, had been different. To address this limitation, we augment the autoregressive process using a structural causal model (SCM) [15, 16], which we denote as $\mathcal{M}$. Our SCM $\mathcal{M}$ is defined by the following assignments[4]:

$$S_0 = S_q, \quad D_i = \begin{cases} f_D(S_{i-1}) & \text{if } \texttt{last}(S_{i-1}) \neq \perp, \\ P_\varnothing & \text{otherwise} \end{cases}, \quad T_i = \begin{cases} f_T(D_i, U_i) & \text{if } D_i \neq P_\varnothing, \\ \varnothing & \text{otherwise} \end{cases}, \tag{1}$$

$$S_i = S_{i-1} \circ T_i \quad \text{and} \quad S = S_K,$$

where S_q and $\boldsymbol{U} = (U_i)_{i \in \{1,\ldots,K\}}$ are independent exogenous random variables, with $S_q \sim P_Q$ and $U_i \sim P_U$, respectively, f_D and f_T are given functions, $P_\varnothing$ is the point mass distribution on $\varnothing$, and $\texttt{last}(S_{i-1})$ denotes the last token of the sequence S_{i-1}. Here, the function f_D is defined by the transformer architecture of the LLM and the choice of function f_T and distribution P_U determines the exact mechanism that the LLM's sampler uses to (stochastically) select the next token T_i. Note that, there always exists a pair of f_T and P_U such that the distribution over tokens D_i matches the distribution $P^{\mathcal{M}}(T_i)$ entailed by $\mathcal{M}$ (see Buesing et al. [48], Lemma 2 for a technical argument). Moreover, note that, in the SCM $\mathcal{M}$, the output sequence S contains the prompt sequence to lighten the notation regarding interventions.

Under this augmented autoregressive process, given an output sequence $S = s$ and noise values $\boldsymbol{U} = \mathbf{u}$, we can generate the counterfactual output sequence the LLM would have generated if the prompt sequence, or some of the tokens in the output sequence had been different, deterministically. More formally, given an intervention $\text{do}[S_i = \tilde{s}]$, with $i \leq |s|$, the counterfactual output sequence $S = S_K$ can be computed recursively using the following expression:

$$S_j = \begin{cases} s_j & \text{if } j < i \\ \tilde{s} & \text{if } j = i \\ S_{j-1} \circ f_T(f_D(S_{j-1}), u_j) & \text{if } j > i \text{ and } \texttt{last}(S_{j-1}) \neq \perp \\ S_{j-1} & \text{otherwise.} \end{cases} \tag{2}$$

[4]We denote random variables with capital letters and realizations of random variables with lower case letters.

Note that the key element of this recursive expression for the counterfactual output sequence is the use of the same realized noise values u_j that were used to generate the factual output sequence s. However, without further assumptions, the counterfactual output sequence may be non-identifiable. This is because there may be multiple noise distributions P_U and functions f_T under which $P^{\mathcal{M}}(T_i) = D_i$, but each pair produces a different counterfactual output sequence—Oberst and Sontag [17] make a similar argument in the context of MDPs. In simpler terms, without explicitly modeling the stochastic mechanism by which the sampler selects the next token in the factual sequence, it is not possible to determine which tokens would have been selected in the counterfactual sequence. Next, we address this issue by focusing on the class of Gumbel-Max SCMs to implement an LLM's sampler.

3 Counterfactual Token Generation Using Gumbel-Max SCMs

Under the class of Gumbel-Max SCMs, the function f_T that implements the sampling of the next token in the SCM $\mathcal{M}$ adopts the following functional form [17]:

$$f_T(D_i, U_i) = \underset{t \in V}{\operatorname{argmax}}\{\log D_{i,t} + U_{i,t}\}, \tag{3}$$

where $U_{i,v} \sim \text{Gumbel}(0, 1)$ are independently distributed Gumbel variables. Importantly, this class of SCMs has been shown to satisfy a desirable counterfactual stability property that can be intuitively expressed as follows. Assume that, at time step i, the augmented autoregressive process sampled token t_i given $d_i = f_D(s_i)$. Then, in a counterfactual scenario where $D_i = d'$, it is *unlikely* that, at time step i, the augmented autoregressive process would have sampled a token t' other than t_i—the factual one—unless, under the token distribution d', the relative chance of generating t_i decreases compared to other tokens. Formally, for any token distribution $d' \in \Delta(V)$ with $d' \neq d_i$ such that

$$\frac{P^{\mathcal{M}}(T_i = t_i \mid D_i = d')}{P^{\mathcal{M}}(T_i = t_i \mid D_i = d_i)} \geq \frac{P^{\mathcal{M}}(T_i = t' \mid D_i = d')}{P^{\mathcal{M}}(T_i = t' \mid D_i = d_i)},$$

it holds that, in the counterfactual scenario where $D_i = d'$, the counterfactual token $T_i \neq t'$.

In addition to solving the non-identifiability issues discussed previously, the use of Gumbel-Max SCMs allows for an efficient procedure to sample a sequence of counterfactual tokens with minimal additional memory requirements compared to vanilla token generation. We summarize the procedure in Algorithm 1 in Appendix A. The algorithm performs the autoregressive computation of Eq. 2 without storing the values u_j for the noise variables that were used during the factual generation, which require a large amount of memory. Instead, it stores the state of the random number generator used at each time step $j \in [K]$ of the factual generation, and it regenerates the values u_j on the fly.

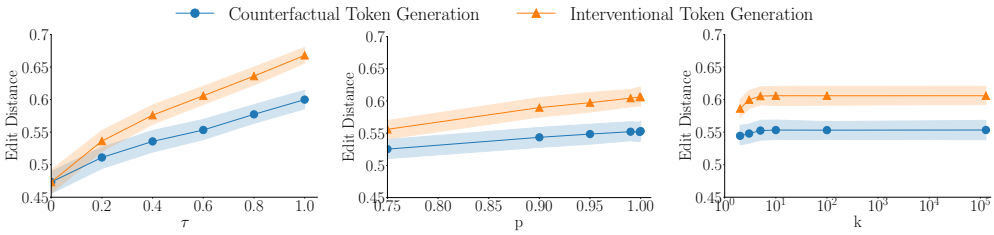
Remarks on implementation aspects of LLMs. In practice, to avoid sampling tokens with very low probability, LLMs may not sample directly from the distribution over tokens d_i at each time step i. Instead, a common practice is to sample from a distribution $\hat{d}_i \in \Delta(V_i)$, where $\hat{d}_{i,t} \propto d_{i,t}$ if $t \in V_i$ and $\hat{d}_{i,t} = 0$ otherwise, where V_i is either the set of most likely tokens of size k under d_i—known as "top-k" sampling—or the set of most likely tokens whose cumulative probability exceeds a given value p under d_i—known as "top-p" or "nucleus" sampling [14]. We can readily implement top-k sampling and top-p sampling in the SCM $\mathcal{M}$ by restricting the argmax in Eq. 3 to the respective set V_i. However, in general, the resulting model is not guaranteed to satisfy counterfactual stability.

In all state-of-the-art LLMs, to ensure that the distribution d_i over tokens at each time step i is a valid probability distribution, the final layer in their neural network is a softmax layer. A crucial feature of this layer is the *temperature* parameter τ, which controls the level of uncertainty in d_i. Intuitively, higher values of τ result in a more uniform distribution, while as τ approaches zero, the distribution concentrates increasingly on the most probable next token. In the next section, we perform a series of experiments in which we analyze the performance of counterfactual token generation, examining the effects of varying temperature values, as well as the application of top-k and top-p sampling.

4 Experiments

In this section, we experiment with an implementation of our model on Llama 3 8B-instruct [49], a popular open-weights large language model. We start by analyzing the similarity between factual and counterfactual text. Further, we demonstrate an application of counterfactual token generation in detecting model biases towards demographic groups.[5]

[5]All experiments ran on an internal cluster of machines, each equipped with 24 Intel(R) Xeon(R) 3GHz CPU cores, 1024GBs of memory and 2 NVIDIA A100 80GB GPUs.



(a) Gumbel-Max SCM (b) Top-p Gumbel-Max SCM (c) Top-k Gumbel-Max SCM

Figure 2: **Comparison between interventional and counterfactual token generation.** The panels show the edit distance between the factual token sequence and the sequence generated by interventional and counterfactual token generation using (a) the Gumbel-Max SCM defined in Eq. 3 and (b) its top-p and (c) its top-k variant discussed at the end of Section 3, against various values of the temperature parameter τ, p and k, respectively. In panels (b, c) the temperature parameter is set to $\tau = 0.6$. In all panels, the edit distance is averaged over $4{,}000$ output sequences, resulting from two independent interventions per factual sequence, and shaded areas represent 95% confidence intervals.

4.1 How similar is counterfactually generated text to the factual one?

As discussed in Section 3, by using the Gumbel-Max SCM, our approach to counterfactual token generation is guaranteed to satisfy the property of counterfactual stability—counterfactual token generation "prioritizes" selecting the same tokens T_i that were selected during the factual generation. As a consequence, we expect the counterfactual text generated using counterfactual token generation to be similar to the factual text. Here, we empirically verify this expectation using a quantitative analysis and explore how it is affected by the model parameters. For a qualitative analysis of the similarities and differences between factual and counterfactual texts, refer to Appendix B.

Experimental setup. We first use the implementation of our model on Llama 3 8B-instruct to generate (factual) outputs to $2{,}000$ question prompts sourced from the LMSYS Chat 1M dataset [50]. As a system prompt we use *"Keep your replies short and to the point."*. Further, for each factual output, we perform two interventions where we replace a randomly selected token t_i with a token $t' \neq t_i$.[6] One of the two interventions restricts the choice of t_i to the first half of the output sequence and the other restricts it to the second half. Then, for each intervened factual output, we feed the concatenation of the question prompt and the first part of the intervened factual output up to including token t' as an input to our model and regenerate the second part of the output after token t' using two approaches:

1. **Interventional token generation:** it uses vanilla autoregressive token generation, that is, it samples new noise values u_j for the second part of the output, as shown in Figure 1c.
2. **Counterfactual token generation:** it uses Algorithm 1, that is, it reuses the same noise values u_j used in the factual generation for the second part of the output, as shown in Figure 1d.

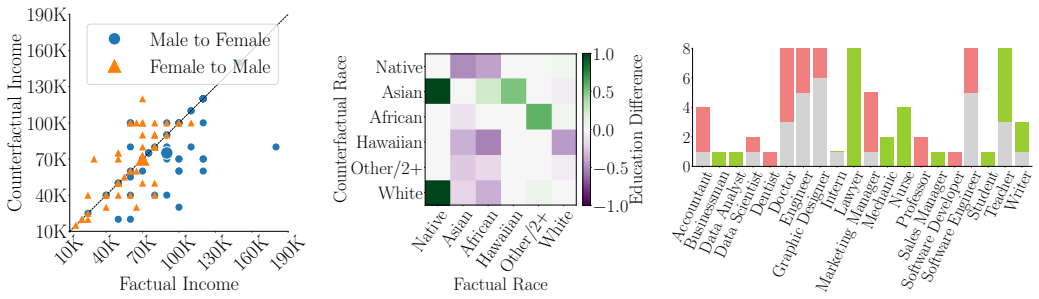
Finally, we measure the lexicographic similarity between the regenerated second part of the output and its factual counterpart using their (normalized) Levenshtein edit distance [51]. In our experiments, we implement our model using the Gumbel-Max SCM defined in Eq. 3 as well as the top-p Gumbel-Max SCM and top-k Gumbel-Max SCM discussed at the end of Section 3.

Results. Figure 2 summarizes the results, which show that the output sequences generated using counterfactual token generation are more similar to the factual sequences (*i.e.*, the edit distance is lower) than the output sequences generated using interventional token generation. This suggests that, even though the top-p and top-k Gumbel-Max SCMs are not guaranteed to satisfy counterfactual stability, in practice, counterfactual token generation under both models do "prioritize" selecting the same tokens T_i that were selected during the factual generation.

4.2 Does counterfactual token generation reveal model biases?

Common approaches to addressing questions of bias and fairness rely on making counterfactual comparisons based on sensitive attributes [52]. For example, would a person's income have been the same if their race or sex were different? In this section, we focus on a census data generation task,

[6] To select t', we set the probability of t_i in d_i to 0, re-scale the values of d_i and use top-p sampling ($p = 0.9$).



(a) Change in income upon intervenion on sex

(b) Change in education level upon intervention on race

(c) Change in occupation upon intervention on race

Figure 3: **Comparison between factual and counterfactual income, education, and occupation.** Panel (a) shows the income of male (female) individuals had they been female (male). Enlarged points correspond to the median income. Panel (b) shows the average difference in the education level of individuals of each race had their race been different. Each race is represented with a short description for visibility; refer to Appendix C for the full descriptions. Panel (c) shows the distribution shift of occupations among Asian American individuals had they been Black or African American. Green (red) sections indicate the increase (decrease) in the number of Asian American individuals that practice each occupation had they been Black or African American. In all experiments, the temperature parameter is set to $\tau = 0.8$.

and demonstrate the use of counterfactual token generation to investigate potential biases of the LLM towards demographic groups.

Experimental setup. We first use the implementation of our model on Llama 3 8B-instruct to generate (factual) census data. To this end, we use the same input prompt three times with different seeds (see Appendix C for details), requesting 50 individuals each time. The factual data generated by the model consist of 114 fictional individuals including their name, age, sex, citizenship, race, ethnicity, marital status, number of children, occupation, income, and education, in this given order. For each fictional person, we consider all possible interventions on each of the sensitive attributes of sex and race. Then, for each intervention, we concatenate the input prompt with the initial part of the output that describes the fictional person (up to and including the intervened sensitive attribute). This concatenated input is then used by our model to regenerate the latter part of the output, following the intervention, using counterfactual token generation (*i.e.*, Algorithm 1). Finally, we compare the factual and counterfactual values of attributes such as income, education and occupation.

Results. Figure 3 summarizes the results, which reveal several interesting insights. Figure 3a shows that, for most male individuals, their generated income would have decreased had they been female, whereas, for female individuals, it would have sometimes increased and sometimes decreased had they been male. This suggests that the model of the world constructed by the LLM does not only present bias but also exhibits inconsistencies in its perceived relationship between a person's sex and income. Figure 3b shows that, for individuals of all (generated) races, there exists at least one other race that, had they belonged to it, they would have experienced a significant increase or decrease in their education level (refer to Appendix C for the assignment of each education level to a numerical value). Finally, Figure 3c shows that, for Asian American individuals, their occupation would have shifted from STEM to humanities related occupations had they been Black or African American.

5 Conclusions

In this work, we have introduced a methodology that enhances state-of-the-art LLMs with the ability to perform counterfactual token generation, allowing them to reason about past alternatives to their own outputs. We have experimentally analyzed the similarity between an LLM's original output and the one generated by counterfactual token generation, and we have demonstrated the use of our methodology in bias detection. Our work opens many avenues for future work. First, our causal model of autoregressive token generation in LLMs crucially relies on the Gumbel-Max SCM. It would be interesting to understand the sensitivity of counterfactual token generation to that choice and consider alternative SCMs. Moreover, we have showcased our model on a single LLM, namely Llama 3 8B-instruct. It would be useful to implement our model on other LLMs and use counterfactual token generation to compare the underlying models of the world constructed by different LLMs.

References

[1] Neal J Roese. Counterfactual thinking. *Psychological bulletin*, 121(1):133, 1997.

[2] Ruth MJ Byrne. Precis of the rational imagination: How people create alternatives to reality. *Behavioral and Brain Sciences*, 30(5-6):439–453, 2007.

[3] Nicole Van Hoeck, Patrick D Watson, and Aron K Barbey. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in human neuroscience*, 9:420, 2015.

[4] Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking. *Personality and social psychology review*, 12(2):168–192, 2008.

[5] Keith D Markman, Matthew N McMullen, and Ronald A Elizaga. Counterfactual thinking, persistence, and performance: A test of the reflection and evaluation model. *Journal of Experimental Social Psychology*, 44(2):421–428, 2008.

[6] Neal J Roese and Kai Epstude. The functional theory of counterfactual thinking: New evidence, new challenges, new insights. In *Advances in experimental social psychology*, volume 56, pages 1–79. Elsevier, 2017.

[7] Robert J Sternberg and Joyce Gastel. If dancers ate their shoes: Inductive reasoning with factual and counterfactual premises. *Memory & Cognition*, 17:1–10, 1989.

[8] David A Lagnado, Tobias Gerstenberg, and Ro'i Zultan. Causal responsibility and counterfactuals. *Cognitive science*, 37(6):1036–1073, 2013.

[9] Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological review*, 128(5):936, 2021.

[10] Yang Xiang, Jenna Landy, Fiery A Cushman, Natalia Vélez, and Samuel J Gershman. Actual and counterfactual effort contribute to responsibility attributions in collaborative tasks. *Cognition*, 241:105609, 2023.

[11] Stratis Tsirtsis, Manuel Gomez Rodriguez, and Tobias Gerstenberg. Towards a computational model of responsibility judgments in sequential human-ai collaboration. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, 2024.

[12] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[15] Judea Pearl. *Causality*. Cambridge university press, 2009.

[16] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[17] Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

[18] Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053, 2019.

[19] Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, 2020.

[20] Changying Hao, Liang Pang, Yanyan Lan, Yan Wang, Jiafeng Guo, and Xueqi Cheng. Sketch and customize: A counterfactual story generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12955–12962, 2021.

[21] Jiangjie Chen, Chun Gan, Sijie Cheng, Hao Zhou, Yanghua Xiao, and Lei Li. Unsupervised editing for counterfactual stories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10473–10481, 2022.

[22] Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. A survey on natural language counterfactual generation. *arXiv preprint arXiv:2407.03993*, 2024.

[23] Ziao Wang, Xiaofeng Zhang, and Hongwei Du. Beyond what if: Advancing counterfactual text generation with structural causal modeling. In *IJCAI*, 2024.

[24] Van Bach Nguyen, Paul Youssef, Jörg Schlötterer, and Christin Seifert. Llms for generating and evaluating counterfactuals: A comprehensive study. *arXiv preprint arXiv:2405.00722*, 2024.

[25] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[26] Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. Prompting large language models for counterfactual generation: An empirical study. *arXiv preprint arXiv:2305.14791*, 2023.

[27] Van Bach Nguyen, Jörg Schlötterer, and Christin Seifert. Ceval: A benchmark for evaluating counterfactual text generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 55–69, 2024.

[28] Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

[29] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

[30] Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International conference on learning representations*, 2022.

[31] Keyon Vafa, Justin Y Chen, Jon Kleinberg, Sendhil Mullainathan, and Ashesh Rambachan. Evaluating the world model implicit in a generative model. *arXiv preprint arXiv:2406.03689*, 2024.

[32] Jörg Frohberg and Frank Binder. Crass: A novel data set and benchmark to test counterfactual reasoning of large language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2126–2140, 2022.

[33] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[34] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

[35] Nick Pawlowski, James Vaughan, Joel Jennings, and Cheng Zhang. Answering causal questions with augmented llms. 2023.

[36] Haitao Jiang, Lin Ge, Yuhe Gao, Jianian Wang, and Rui Song. Large language model for causal decision making. *arXiv preprint arXiv:2312.17122*, 2023.

[37] Lorenzo Betti, Carlo Abrate, Francesco Bonchi, and Andreas Kaltenbrunner. Relevance-based infilling for natural language counterfactuals. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 88–98, 2023.

[38] Xiao Liu, Da Yin, Chen Zhang, Yansong Feng, and Dongyan Zhao. The magic of if: Investigating causal reasoning abilities in large language models of code. *arXiv preprint arXiv:2305.19213*, 2023.

[39] Xin Miao, Yongqi Li, and Tieyun Qian. Generating commonsense counterfactuals for stable relation extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5654–5668, 2023.

[40] Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*, 2023.

[41] Allen Nie, Yuhui Zhang, Atharva Shailesh Amdekar, Chris Piech, Tatsunori B Hashimoto, and Tobias Gerstenberg. Moca: Measuring human-language model alignment on causal and moral judgment tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. *arXiv preprint arXiv:2402.11655*, 2024.

[43] Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*, 2024.

[44] Stratis Tsirtsis, Abir De, and Manuel Gomez-Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. *Advances in Neural Information Processing Systems*, 34:30127–30139, 2021.

[45] Kimia Noorbakhsh and Manuel Gomez-Rodriguez. Counterfactual temporal point processes. *Advances in Neural Information Processing Systems*, 35:24810–24823, 2022.

[46] Nina L Corvelo Benz and Manuel Gomez Gomez-Rodriguez. Counterfactual inference of second opinions. In *Uncertainty in Artificial Intelligence*, pages 453–463. PMLR, 2022.

[47] Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 144–155, 2024.

[48] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *Proceedings of the 7th International Conference on Learning Representations*, 2018.

[49] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[50] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023.

[51] V Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*, 1966.

[52] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in Neural Information Processing Systems*, 2017.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

A Efficient counterfactual token generation using a Gumbel-Max SCM

ALGORITHM 1: It returns a counterfactual sequence of tokens using a Gumbel-Max SCM

Input: Random number generator states r, factual output sequence s, intervention $(i, \tilde{s})$.
Output: Counterfactual output sequence s'.
for $j = 1, \ldots, K$ **do**
  if $j < i$ **then**
    $s'_j = s_j$
  else if $j = i$ **then**
    $s'_j = \tilde{s}$
  else if $j > i \wedge \texttt{last}(s'_{j-1}) \neq \perp$ **then**
    $u_j = \text{GenGumbel}(r_j)$
    $d'_{j,t} = f_D(s'_j)$
    $t_j = \text{argmax}_{t \in V}\{\log d'_{j,t} + u_{j,t}\}$
    $s'_j = s'_{j-1} \circ t_j$
  else
    $s'_j = s'_{j-1}$
Return s'_K

Algorithm 1 presents an efficient procedure that uses a Gumbel-Max SCM to sample a sequence of counterfactual tokens with minimal additional memory requirements compared to vanilla token generation. Recall that, to generate the counterfactual output sequence, one needs to use the same values u_j for the noise variables that were used during the factual generation and then perform an autoregressive computation based on Eq. 2. Instead of storing the values u_j for all time steps $j \in [K]$, whose dimensionality matches the size of the vocabulary V, Algorithm 1 employs a simple idea: it stores the state of the random number generator r_j used at each time step $j \in [K]$ of the factual generation. Then, during the counterfactual generation, it regenerates the values $u_j = \text{GenGumbel}(r_j)$ on the fly. Storing the realized values of the Gumbel variables requires storing $\mathcal{O}(KV)$ float values since $u_j \in \mathbb{R}^V$. On the other hand, the states of random number generators take values in $\mathbb{N}^d$, where, for instance, $d = 16$ in `pytorch` [53]. Thus, our approach requires $\mathcal{O}(K)$ additional integer memory compared to vanilla token generation.

Captain **Lyra** stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(a) Factual story

Captain **Maeve** stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, she finally spotted its shimmering shores. As she stepped onto the island, a radiant light enveloped her, and she wished for her lost love, the brave sailor, Eamon, to return to her. Suddenly, Eamon emerged from the mist, frail but alive. Overjoyed, Maeve scooped him into her arms, and they shared a tender kiss as the sun dipped into the horizon, their love reunited at last.

(b) Counterfactual story with a modified name for the captain

Figure 4: **Example of a factual and a counterfactual story.** Panel (a) shows the factual story, as given by the LLM. Panel (b) shows the story resulting from counterfactual token generation using Algorithm 1. For the counterfactual generation, we give as input to the LLM the original prompt along with the first sentence of the factual output (non-highlighted text), modified by replacing "Lyra" with "Maeve". Green-highlighted text indicates the part of the output that is identical in both the factual and counterfactual stories, while red-highlighted text indicates the difference. In both panels, the temperature parameter is set to $\tau = 0.9$.

B How would the story have unfolded for "Captain Maeve"?

As discussed in Sections 3 and 4.1, due to the property of counterfactual stability, we expect the counterfactual text generated using counterfactual token generation to be similar to the factual text. Here, we investigate this qualitatively through an anecdotal example of story generation.

We use the implementation of our model on Llama 3 8B-instruct with the system prompt *"Be creative and keep your response as short as possible."* and a query prompt *"Tell me a fantasy story about a captain. The story should have either a happy or a sad ending."* Figure 4a shows the (factual) generated story about Captain Lyra, her ship the Maelstrom's Fury, and her quest to find a treasure on the Golden Isle. We use the original prompt along with part of the factual output (*i.e.*, the first sentence of the story) as input to the model, modifying the protagonist's name from "Lyra" to "Maeve". We then use counterfactual token generation to regenerate the rest of the output.

The counterfactually generated story shown in Figure 4b reveals several interesting insights. As expected, due to the counterfactual stability property of the Gumbel-Max SCM and the minor nature of changing the protagonist's name, the initial part of the counterfactual output remains identical to the factual output. Although one may expect this to apply for the rest of the counterfactual output, thinking that the protagonist's name would be irrelevant to the narrative of this particular story, this is not the case. Perhaps surprisingly, the use of "Maeve" instead of "Lyra" results in a partially different output, illustrating that the LLM's probability distributions over next tokens are sensitive even to minor changes. In Figure 5, we also observe differences between the factual and counterfactual outputs resulting from other seemingly irrelevant interventions, such as changing the name of the ship, removing the adjective "trusty" or replacing the word "sea" with "blue".

Captain Lyra stood at the helm of her trusty ship, the "**Maelstrom's Fury**", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(a) Factual story

Captain Lyra stood at the helm of her trusty ship, the "**Cyclone's Rage**", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold coins appeared. But, just as she reached for it, a storm brewed, and a rival captain, seeking the treasure for himself, appeared. Lyra fought valiantly, but in the end, she was defeated. As she lay on the beach, her ship destroyed and her crew scattered, Lyra realized her wish had been granted — not for wealth or power, but for the courage to sail through life, no matter the outcome.

(b) Counterfactual story with a modified name for the ship

Captain Lyra stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless **sea**. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(c) Factual story

Captain Lyra stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless **blue**. For years, she'd sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she anchored the ship and stepped onto the sandy beach, a warm light enveloped her. With trembling hands, she reached out and touched the glittering treasure. The Golden Isle's magic surged through her, and Lyra's greatest desire took hold: to reunite with her lost love, who'd perished at sea many years ago. With tears of joy streaming down her face, Lyra opened her eyes to find her beloved standing before her, alive and well. The two shared a long, tender kiss, surrounded by the sweet scent of tropical flowers. The "Maelstrom's Fury" sailed away, its crew rejoicing, as Lyra and her love disappeared into the sunset, their hearts full of happiness.

(d) Counterfactual story with a modified word

Captain Lyra stood at the helm of her **trusty** ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(e) Factual story

Captain Lyra stood at the helm of her _____ ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever lost at sea.

(f) Counterfactual story with a removed word

Figure 5: **Comparison between the factual story and counterfactual variants.** Panels (a, b, c) show the same factual story as in Figure 4. Panels (b, d, f) show the story resulting from various interventions. In each case, the first sentence (non-highlighted text) is provided as input to the LLM, with the word(s) in bold (or left empty) representing the intervention. The remainder of the output is regenerated using counterfactual token generation. Text highlighted in green indicates identical content in both the factual and counterfactual stories, while text highlighted in red indicates the differences.

System: Return only the following information: Name, Age, Sex, Citizenship, Race, Ethnicity, Marital Status, Number of Children, Occupation, Income, Education. For Race, choose only between following options: White American, Black or African American, American Indian or Alaska Native, Asian American, Native Hawaiian or Other Pacific Islander, Other or Two or more races (multiracial). For Ethnicity, choose only between following options: Non-Hispanic/Latino or Hispanic/Latino. Return a list in json format delimited by \"```\".
User: Generate census data of 50 fictional people.

Figure 6: The prompt used for census data generation.

C Additional details on the experimental setup of Section 4.2

In this section, we provide additional details about the census generation experiment discussed in Section 4.2. Figure 6 shows the complete system and user prompts used to generate the census data. For race and ethnicity, we instructed our model, through the system prompt, to select values among those reported in the latest (2020) US Census. We used this prompt three times with different seeds. Despite our request for 50 individuals per generation, the LLM only generated 34, 39 and 41 individuals each time, resulting in a total of 114 individuals. Table 1 contains the full descriptions of the race attribute values, of which shortened versions were used in Figure 3b. Finally, Table 2 lists the numerical values assigned to the (categorical) education attribute values, used to compute the difference in education level shown in Figure 3b.

Table 1: Short and full description of all races

Short	Full
Native	American Indian or Alaska Native
Asian	Asian American
African	Black or African American
Hawaiian	Native Hawaiian or Other Pacific Islander
Other/2+	Other or Two or more races (multiracial)
White	White American

Table 2: Numerical value assigned to each (categorical) value of the education attribute

Education	Numerical Value
High School Diploma	1
High school diploma	1
Associate's degree	2
Some college	2
Bachelor's degree	3
Master's degree	4
Ph.D.	5
Law Degree	5
Law degree	5
Juris Doctor	5
Medical Degree	5
Medical degree	5
Dental degree	5
Dentistry degree	5