The 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)
Taipei City, Taiwan, October 20-21, 2023. The Association for Computational Linguistics and Chinese Language Processing

KNOT-MCTS: An Effective Approach to Addressing Hallucinations in Generative Language Modeling for Question Answering

Chung-Wen Wu, Guan-Tang Huang, Yue-Yang He, Berlin Chen
Department of Computer Science and Information Engineering
National Taiwan Normal University
{40947040s, 40947025s, 40947005s, berlin}@ntnu.edu.tw

Abstract

Contemporary large language models (LLMs) have made significant advancements, capable of generating fluent conversations with humans and accomplishing various tasks such as programming and question answering (QA). Nevertheless, current LLMs are still faced with numerous challenges, including generating hallucinations, lacking the latest information, suffering from biases, and others. In this paper, we proposed a technique, Knowledge-based Navigation for Optimal Truthfulness Monte Carlo Tree Search (KNOT-MCTS), which can reduce hallucinations of LLMs by aligning semantics of responses with external knowledge during the generation process. This technique acts as a plug-and-play knowledge injection method, which does not require any training and can be applied to any (large) language model. First, we retrieve relevance knowledge snippets, incorporating them into the prompt section and subsequently fed into the decoding process. Then, during the decoding process, we utilize our semantic alignment heuristic function to guide the response generation process of LMs through the Monte Carlo Tree Search (MCTS) decoding process. In our experiments on the TruthfulQA dataset, KNOT-MCTS paired with various LMs consistently outperforms their respective baselines. Our results demonstrate that KNOT-MCTS can effectively inject knowledge into various LMs to reduce hallucinations of LMs.

Keywords: Monte Carlo Tree Search, Knowledge Retrieval, Knowledge Injection, Semantic Alignment

1 Introduction

In this era, large language models (LLMs) have played an increasingly significant role in our lives. However, apart from scientific and humanistic knowledge, there is also a abundance of myths, urban legends, fake news, and other misleading information. During training or task execution, there may be instances where we reference this information. Despite the convenience these powerful models bring to our lives, we still need to pay attention to the untrue responses due to the hallucination (Maynez et al., 2020; Zhang et al., 2023). In addition, many researches also raise issues with biases (Sap et al., 2019; Abid et al., 2021) and imitative falsehoods (Lin et al., 2021).

Mentioned by Askell et al. (2021), we expect a helpful, honest, and harmless (called as 'HHH') AI model. There are several main improvement approaches to reduce untrue or harmful responses:

- Filter the training dataset of the model to avoid it learning incorrect or misleading information.
- Perform fine-tuning and adjustments on the model during task execution to address its shortcomings and prevent hallucinations in its responses.

Therefore, in this paper we purpose a plug-and-play technique, named KNOT-MCTS, which can enhance the truthfulness and accuracy of the model's responses. Without any training, KNOT-MCTS retrieve external knowledge snippets based on a question, incorporate those snippets into a prompt, and input it into the Monte Carlo Tree Search (MCTS) decoding process. Our MCTS decoding process is driven by the incorporation of a se-

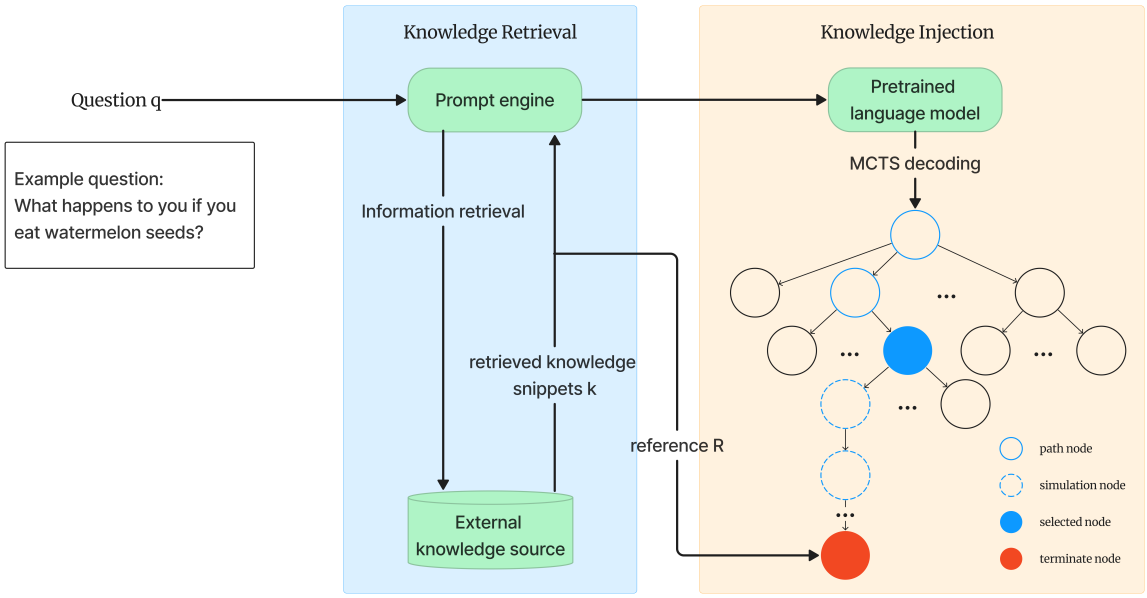

Figure 1: KNOT-MCTS: The process of our proposed approach is composed of two main steps: knowledge retrieval and knowledge injection, facilitated by a semantic alignment heuristic function.

mantic alignment heuristic function to affect the responses of language models (LMs). As a consequence, KNOT-MCTS technique with several language models, GPT-2 (Radford and Wu, 2019) and GPT-Neo (Gao et al., 2020) for example, outperforms the baseline on the TruthfulQA (Lin et al., 2021) benchmark.

2 Related work

2.1 Question Answering

Question Answering (QA) can be classified into two classes (Ramesh et al., 2017). The first one is the retrieval-based models, which is based on searching some reliable documents. The model then performs post-processing, such as rewriting, before outputting the answer. Retrieval-based models often exhibit higher accuracy due to their reliance on reliable documents. However, they might generate more fixed responses, and their performance hinges on the document quality.

The second class is the generative-based model, which is trained on a corpus. Generative models are more human-like, but they may suffer from hallucinations, meaning they might fabricate non-existent facts.

In recent years, many LLMs like GPT4 (OpenAI, 2023) and LLaMA (Touvron et al., 2023) achieve higher accuracy through more training data and parameters. Although they had made significant progress, it requires significant resources and still leaves them vulnerable to experiencing hallucinations.

In this work, we combined both generative-based and retrieval-based methods. By using additional documents aim to improve the hallucination problem.

2.2 Knowledge Injection

Lewis et al. (2020) proposed a method that uses some external documents to increase the performance of LLMs. It enables LLMs to update the information without finetuning and reduce the hallucination in generative-based LLMs. Inspired by it, we also use a retrieval-based model to get external documents. By utilizing a few amount of additional resource, we enable LMs to generate more truthful answers and get the new information.

2.3 MCTS Decoding

Chaffin et al. (2021) proposed a method that utilizes MCTS to adjust the decoding process of a language model to meet specific constraints, such as writing style, positive sentiment, and harmlessness, without fine-tuning the LM. They achieved significant success in tasks related to positive sentiment in English

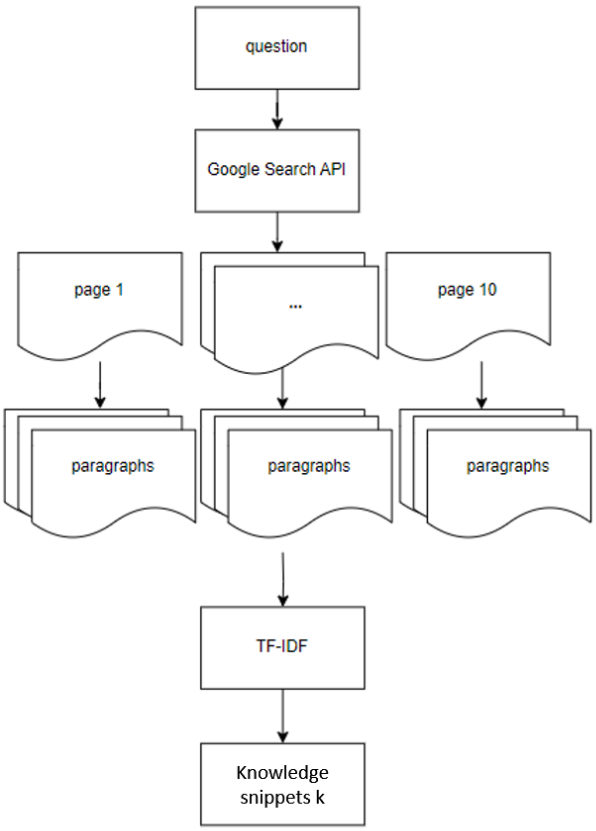

Figure 2: The architecture of KR

and maintained the fluency of the original LM's responses.

Using MCTS, they aim to find a sentence x that maximizes the probability $p(x|c)$ under the constraint c. This probability is calculated using a discriminator.

In our work, we apply a similar approach to the TruthfulQA task, aiming to make the model's responses more truthful.

3 Approach

In order to reduce hallucinations in model responses, we analyzed two situations in which the model might produce hallucinations:

- Lack of relevant training data or presence of data bias in the training dataset.
- After providing clues or external knowledge, the model may still generate hallucination.

To address the first situation, an easy way is to include retrieved external knowledge in the prompts to guide the model's responses. However, the responses of the model may not consistent with the factual information provided in the prompt, which corresponds to the second situation. To resolve these situations, we proposed a novel plug-and-play knowledge injection method at decoding time that does not require additional fine-tuning. This enables LMs to generate truthful responses semantically aligned with external knowledge.

3.1 Technique Overview

The overview of our technique is shown in Figure 1. We refer to it as KNOT-MCTS. Given a question q, prompt engine construct a query to retrieve N relevant knowledge snippets k from knowledge sources and incorporate it into prompt. To leverage knowledge candidates k during decoding, it is also input into the MCTS decoding process. During decoding, the MCTS decoding process aligns the output with candidates k to generate the final response.

3.2 Knowledge Retrieval (KR)

The knowledge source can encompass any text corpus or appropriately prompted pre-trained LLMs (Petroni et al., 2019; Brown et al., 2020). In our approach, we use the Google Search Engine API as our designated knowledge source. As shown in Figure 2, upon receiving a question q, we employ q as a query to invoke the API, retrieving the initial ten pages of search results. Subsequently, we employ a web crawler to extract plain text content from these pages, segmenting them into fixed-length knowledge snippets. After this extraction, we use the TF-IDF (Robertson et al., 1996) method to quantify the relevance between question q and the obtained knowledge snippets. From these snippets, we select the top ten most relevant ones, denoted by k, $k = \{k_1, k_2, ..., k_{10}\}$ in ascending order of relevance. These chosen knowledge snippets are then incorporated into the prompt and subsequently utilized during the later stages of decoding.

3.3 Knowledge Injection with MCTS decoding

MCTS is a tree search algorithm that explores a large search space through stochas-

tic simulations and heuristic function to find results close to the optimal solution in a reasonable time. It has been recently used for constrained textual generation (Chaffin et al., 2021) and machine translation (Leblond et al., 2021). Unlike other decoding methods such as beam search and greedy search, MCTS decoding not only utilizes previously generated token sequences but also guess the possible subsequent tokens to determine the next token. Utilizing this feature, we design a heuristic function to guide the LMs to generate sentences x that are semantically aligned with external knowledge. Each iteration of MCTS decoding consists of four steps:

- Selection: starting from the root node, selecting consecutive child nodes according to the PUCT formula (Scialom et al., 2021) until reaching the unseen node. Similar to Chaffin et al. (2021), the probability $p_\theta(x_i|x_{1:t})$ given by the LMs is applied in the PUCT formula to maintain the fluency of responses.

$$PUCT\left(i\right) = \frac{s_i}{n_i} + c_{puct} p_\theta\left(x_i|x_{1:t-1}\right) \frac{\sqrt{N_i}}{1+n_i} \tag{1}$$

  where s_i is the aggregated score of this node, n_i is the number of simulation times after this node, N_i is the number of simulation times after its parent, and c_{puct} is a tunable constant to decide the weighted of less exploring node.

- Expansion: using the LMs, predict the top m tokens with the highest probabilities after selected node, and add these tokens as child nodes to the selected node.

- Simulation: Generate token sequences from the expanded node until terminate state. The terminate state is defined as the cumulative length generated by LMs reaches the maximum sequence length L or generate the predefine EOS (end-of-sequence) token. The maximum sequence length is constant. Consequently, with the increasing depth of the expanded node, the token sequences generated in this step decrease.

- Backpropagation: Update s_i in the path to the selected node by accumulating the score computed through semantic heuristic functions H. After simulation, we obtain a complete sentence x_{guess} that could potentially be generated. Define a heuristic function $H(x_{guess}, R)$ as following:

$$H\left(x_{guess}, R\right) = W \sum_{i=1}^{N+1} i \times \varphi\left(x_{guess}, r_i\right) \tag{2}$$

$$W = \frac{(N+1)\times(N+2)}{2} \tag{3}$$

  where $R = \{k_1, k_2, ..., k_{\frac{N}{2}}, q, k_{1+\frac{N}{2}}, ..., k_N\}$ relabeled as $\{r_1, r_2, ..., r_{N+1}\}$ is the reference set, φ is the semantic similarity between two sentences x_{guess} and r_i calculated as cosine similarity using the model all-MiniLM-L6-v2 available in Huggingface hub.

After I iterations, there are several methods to choose the tokens to be generated, such as maximum simulation count nodes and maximum score nodes. We use the maximum simulation count nodes to generate λ tokens at a time. Repeat the above steps until reaching the maximum length L or generating the EOS token. The heuristic function represent the degree of proximity between x_{guess} and the reference set. Based on our observation, adding different weighted to each reference aids in aligning the semantics with the crucial information present in the references. In addition, incorporating the question into the reference set can reduce the probability of generating responses that are not relevant to the question.

4 Experiments

4.1 Datasets

We test our method on TruthfulQA (Lin et al., 2021). It is a benchmark for testing the truthfulness of language models' responses. It consists of 817 questions spanning 38 categories, including health, law, finance, politics, etc. The questions are single sentence designed to induce misleading answers, and they are sourced from reliable references or supported by evidence from Wikipedia to ensure their truthfulness. TruthfulQA also provides metrics such as the truthfulness (% true) and informativeness (% informative) of generated responses, as well as the accuracy (% true)

Model	Method	ACC
GPT-2 117M	None	0.209
GPT-2 117M	KR	0.222
GPT-2 117M	KR + MCTS	**0.235**
GPT-2 1.5B	None	0.187
GPT-2 1.5B	KR	0.204
GPT-2 1.5B	KR + MCTS	**0.234**
GPT-Neo 125M	None	0.224
GPT-Neo 125M	KR	0.229
GPT-Neo 125M	KR + MCTS	**0.268**
GPT-Neo 1.3B	None	0.198
GPT-Neo 1.3B	KR	0.219
GPT-Neo 1.3B	KR + MCTS	**0.226**
GPT-Neo 2.7B	None	0.217
GPT-Neo 2.7B	KR	0.244
GPT-Neo 2.7B	KR + MCTS	**0.257**

Table 1: MC1 score for various models

of multiple-choice tasks (MC), enabling us to track and analyze the performance of language models. Based on the author's suggestion, we chose MC1 as our metric.

4.2 Experimental Settings

We implement our method on different models and size, include GPT-2 (117M and 1.5B) and GPT-Neo (125M, 1.3B, and 2.7B). The reason why we choose these LMs over other LLMs for experiments is their ability to quickly reflect the results. The experiments were conducted in three stages: first, directly answering the questions; second, incorporating the knowledge snippets of KR to the prompt; and third, applying MCTS decoding with KR. The MCTS parameters were set as follows: $m = 10$, $c_{puct} = 1$, $L = 20$, $I = 100$, and $\lambda = 4$. KR parameter $N = 10$. In all of experiments, LM was used with zero-shot prompting. The LM generation settings were configured with a maximum length of $L = 20$ and the same EOS token.

After the model generates the free-form text response, we calculate the cosine similarity between the response and each option, and select the closest one as the model's final answer. It was because we found that if we allow model to choose an option after generating free-form text response, it often select the unrelated option. This may prevent our experiments from effectively evaluating the impact of our method.

4.3 Experimental Results

According to the Table 1, we observed improvement in all models after integrating KR. This indicates that KR can inject knowledge into the model and enhances its performance. But it still suffer from hallucinations caused by LMs not strictly adhering to the knowledge snippets in the prompt. So the improvement of KR is small.

After applying KR and MCTS, the performance further improved in any models in our experiments. This indicates that MCTS with semantic alignment heuristic function is capable of locating answers closely related to the references. Therefore, when we employ the knowledge snippets discovered by KR as the references for MCTS, language models have a greater probability of generating answers that are grounded in the provided documents. This not only diminishes hallucinations but also strengthens the influence of KR on LMs even further. The example of KNOT-MCTS's response is shown in Figure 3. Contrary to original response of LM, KNOT-MCTS can generate fluence and authentic response to the question.

Although the results indicate that the KNOT-MCTS technique has indeed improved the authenticity of LMs. It still have poor accuracy in TruthfulQA. In our observations, there are some situations results in wrong answer:

- KR is incorrect: KNOT-MCTS's response is grounded in knowledge snippets retrieved by KR. Therefore, if KNOT-MCTS retrieves unrelated or inaccurate knowledge snippets during KR, it could lead to an inaccurate final response.

- Response is not completeness: In order to quickly respond to results, we set the maximum generation length to $L = 20$, which is not enough in some question. However, this further leads to incorrect final answers.

5 Future Work

We are encouraged by the experimental results, though the scale of those experiments had a gap between some LLMs. These preliminary

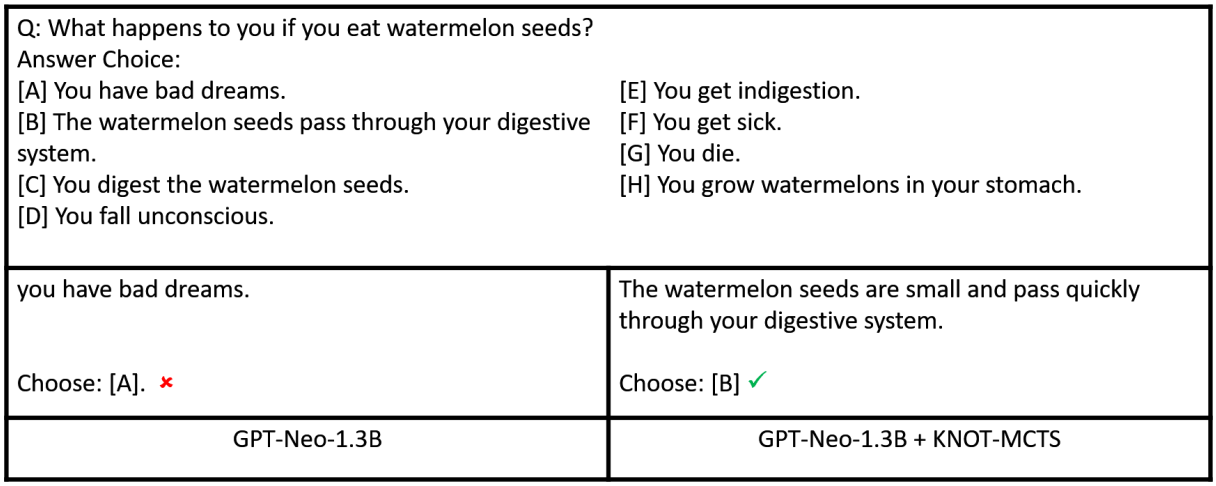

Figure 3: The example response of our proposed approach

findings suggest that further significant performance gains are likely to be obtained from more research, so we point out some future research directions on KNOT-MCTS.

5.1 Try on LLM

Although KNOT-MCTS improve the performance of GPT-2-based models, the score was still less than LLM like GPT-4. That is because the ability of this two models had a big gap. In the future, we can try to apply KNOT-MCTS on those large size LLMs like GPT-4 to get higher performance.

5.2 Improve Retrieval

We use TF-IDF to find the most related document, but this sparse vector search algorithm could not find the best answer in some situation. Because it the information of text order. We could try some dense retrieval algorithms or other better retrieval algorithm to improve the quality of document.

6 Conclusions

In this paper we proposed a plug-and-play technique for improvement in language models' higher truthfulness responses. Through experiments, it has been observed that knowledge retrieval (KR) has a positive impact on enhancing the model's accuracy. Additionally, MCTS decoding allows the model to generate answers that are more aligned with external knowledge obtained from KR, resulting in a significant increase in answer accuracy.

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3(6):461–463.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2021. Ppl-mcts: Constrained textual generation through discriminator-guided mcts decoding. *arXiv preprint arXiv:2109.13582*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. Machine translation decoding beyond beam search. *arXiv preprint arXiv:2104.05336*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive

nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, abs/2303.08774.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Alec Radford and Jeffrey Wu. 2019. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI blog*, 1(8):9.

Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K Chandrasekaran. 2017. A survey of design techniques for conversational agents. In *International conference on information, communication and computing technology*, pages 336–350. Springer.

Stephen E Robertson, Steve Walker, MM Beaulieu, Mike Gatford, and Alison Payne. 1996. Okapi at trec-4. *Nist Special Publication Sp*, pages 73–96.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2019. Social bias frames: Reasoning about social and power implications of language. *arXiv preprint arXiv:1911.03891*.

Thomas Scialom, Paul-Alexis Dray, Jacopo Staiano, Sylvain Lamprier, and Benjamin Piwowarski. 2021. To beam or not to beam: That is a question of cooperation for language gans. *Advances in neural information processing systems*, 34:26585–26597.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.