Enhancing Factuality in Language Models through Knowledge-Guided Decoding

Stanford CS224N Custom Project

Jirayu Burapacheep
Department of Computer Science
Stanford University
`jirayu@stanford.edu`

Abstract

Large language models (LLMs) have shown impressive performance in text generation tasks, but they often struggle to maintain factual consistency and reliability in their outputs. Existing methods for enhancing LLM factuality do not guarantee that the generated text is consistent with the provided information. In this paper, we propose Knowledge-Guided Decoding (KGD), a novel framework that addresses these limitations by directly influencing the language model's generation process at the token level based on retrieved knowledge. KGD incorporates knowledge-guided rewards, such as semantic similarity and entailment, to guide the model's output toward more accurate and trustworthy text. We evaluate our approach on both short-form and long-form text generation tasks, focusing on measuring the factuality of the generated output. Through extensive experiments, we demonstrate that KGD effectively improves the factual accuracy of the generated text while maintaining fluency, outperforming traditional decoding baselines.

1 Key Information to include

- Mentor: N/A
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in text generation but often struggle with factual consistency and reliability (Mallen et al., 2023). Traditional methods to enhance LLM factuality typically involve retrieval-augmented generation strategies, where prompts are supplemented with relevant information retrieved from an external knowledge source (Lewis et al., 2020). However, these methods can reduce the versatility of LLMs and introduce irrelevant information, leading to low-quality output (Shi et al., 2023). Moreover, these approaches don't guarantee the outputs are consistent with the information since the models are not explicitly trained to leverage and follow facts from provided information (Gao et al., 2023).

In this paper, we propose a complementary framework, Knowledge-Guided Decoding (KGD), that addresses these limitations by directly influencing the language model's generation process at the token level based on retrieved knowledge. KGD incorporates knowledge-guided rewards, such as semantic similarity and entailment, to guide the model's output toward more accurate and trustworthy text (Figure 1). By assigning higher rewards to tokens that align with the retrieved information, KGD aims to generate text that is both fluent and factually consistent.

We evaluate our approach on both short-form and long-form text generation tasks, focusing on measuring the factuality and fluency of the generated output. Through extensive experiments,

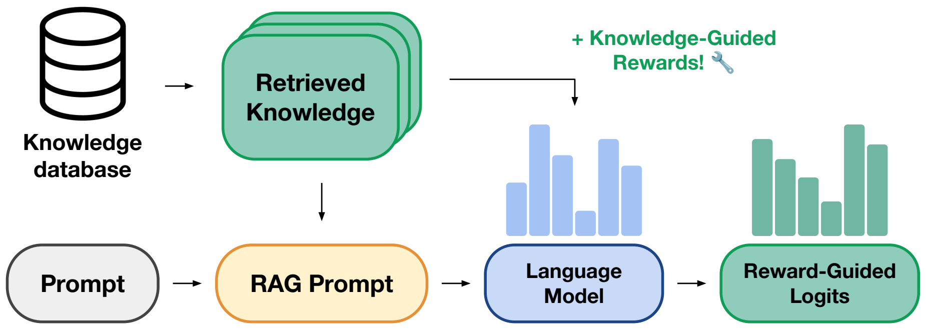


Figure 1: Illustration of the Knowledge-Guided Decoding (KGD) framework. The language model takes the prompt as input, retrieves relevant knowledge from a database, and incorporates this knowledge along with reward-guided logits to generate the final output.

we demonstrate that KGD effectively improves the factual accuracy of the generated text while maintaining fluency, outperforming several strong baselines. We further conduct ablation studies to investigate the impact of various hyperparameters on the performance of KGD and provide a qualitative analysis to gain deeper insights into its strengths and limitations.

3 Related Work

Retrieval-Augmented Generation. Prior work has shown that augmenting language models with retrieved text passages can significantly improve performance on knowledge-intensive tasks (Guu et al., 2020; Lewis et al., 2020; Ram et al., 2023). Recent approaches include instruction-tuning a language model with a fixed number of retrieved passages prepended to the input (Luo et al., 2023), jointly pre-training a retriever and language model followed by few-shot fine-tuning (Izacard et al., 2022), and adaptively retrieving passages during generation (Jiang et al., 2023; Schick et al., 2023). However, these methods often trade off runtime efficiency (Mallen et al., 2023), robustness to irrelevant context (Shi et al., 2023), and lack of attributions (Liu et al., 2023; Gao et al., 2023).

Guided decoding. Our works distinguishes itself in the token-level guided decoding literature by using a reward model that guides generation at the token level, rather than focusing on step-level verifiers that typically emphasize sentence-level analysis (Welleck et al., 2022; Uesato et al., 2022; Lightman et al., 2023; Krishna et al., 2022; Li et al., 2023b; Khalifa et al., 2023; Xie et al., 2023; Yao et al., 2023). While token-level guided decoding has been explored in the past (Dathathri et al., 2020; Krause et al., 2021; Yang and Klein, 2021; Lu et al., 2021; Chaffin et al., 2022; Liu et al., 2021; Li et al., 2023a; Deng and Raffel, 2023; Khanov et al., 2024), they have not connected language decoding directly to the factuality correcting problem of our interest, especially in the context of utilizing a reward mechanism on retrieved knowledge.

4 Methodology

In this section, we introduce Knowledge-Guided Decoding (KGD), a simple and complementary framework that directly influences language model generation at the token level based on the retrieved information. Our key idea is to (1) retrieve information from the knowledge source using the query and (2) adjust the model's probabilistic prediction by a reward signal at each decoding step. By assigning higher rewards to tokens that align with the retrieved information, we aim to guide the LLM's output toward more reliable and trustworthy text. We detail the decoding procedure in Section 4.1, different knowledge-guided rewards in Section 4.2, and the computational efficiency in Section 4.3. We provide the complete pipeline of our framework in Algorithm 1.

4.1 Knowledge-Guided Decoding

We aim to steer the language model's generation process by incorporating retrieved knowledge at each decoding step. Let $\mathcal{K}$ denote the knowledge source (e.g., a textual knowledge base) and q be the input query. Similar to retrieval-augmented generation, we first retrieve a set of relevant information $k = \{k_i\}_{i=1}^{N}$ from $\mathcal{K}$ based on the similarity between q and each knowledge entry k_i. This retrieval step can be performed using standard retrieval techniques like sparse or dense retrieval methods.

At each decoding step t, the language model outputs logits $z(x_t|x_{<t})$ over the vocabulary, where $x_{<t}$ represents the tokens generated so far. In standard decoding, the next token x_t is sampled directly from the probability distribution $p(x_t|x_{<t})$, which is obtained by applying a softmax function over the logits. KGD modifies this distribution by applying a reward r that scores each token based on its alignment with the retrieved knowledge:

$$z_{\text{KGD}}(x_t|x_{<t}) = z(x_t|x_{<t}) + w \cdot r(y_t, k), \tag{1}$$

where w denotes the weight assigned to the knowledge-guided reward term and $y_t = [x_{<t}, x_t]$ denotes the concatenation of the previously generated tokens $x_{<t}$ and the current token x_t.

Intuitively, tokens that are more relevant to the retrieved knowledge receive higher rewards, increasing their probability of being generated. The weighting parameter w controls the influence of the reward. As w approaches 0, KGD reduces to standard language model decoding, while larger values of w result in generations that more closely follow the retrieved knowledge. We discuss the trade-off comprehensively in Section 5.4.

4.2 Knowledge-Guided Rewards

The performance of KGD relies on the choice of reward function r that determines the token-level knowledge alignment scores. By incorporating these knowledge-guided rewards into the decoding process, KGD aims to generate more semantically coherent text that aligns with the retrieved knowledge. The flexibility of the reward formulation allows for easy extension and adaptation to various knowledge sources and tasks, making KGD a versatile framework for knowledge-guided language generation. In this paper, we propose and evaluate several reward schemes:

Similarity-based reward. To capture semantic similarities between the generated text and retrieved information, we compute a dense vector representation of each candidate y_t and knowledge entry k_i using a pre-trained sentence embedding model. The semantic similarity reward is then defined as:

$$r_{\text{sim}}(y_t, k) = \max_{i=1,\dots,N} \cos(\mathbf{e}(x_t), \mathbf{e}(k_i)) \tag{2}$$

where $\mathbf{e}(\cdot)$ denotes the embedding function and $\cos(\cdot,\cdot)$ is the cosine similarity. This reward captures semantic alignment between each candidate and the most similar retrieved knowledge entry.

Entailment-based reward. The retrieved knowledge can also be used to guide generation by rewarding tokens that lead to model outputs that entail the knowledge. We use a pre-trained natural language inference (NLI) model to compute the entailment classification between the candidate y_t and each knowledge entry k_i:

$$r_{\text{ent}}(y_t, k) = \begin{cases} \alpha & \text{if } y_t \text{ entails } k_i \text{ for some } i \in [N] \\ -\beta & \text{if } y_t \text{ contradicts } k_i \text{ for some } i \in [N] \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where α and β are positive hyperparameters that control the strength of the reward for entailment and contradiction, respectively. This reward encourages the model to generate text that is logically consistent with the retrieved knowledge while penalizing contradictory statements. Note that we can set the weight w to 1, as the hyperparameters α and β already control the influence of the reward on the decoding process.

Combined reward. Finally, to leverage the benefits of both similarity-based and entailment-based rewards, we propose a combined reward function that incorporates both aspects:

$$r_{\text{comb}}(y_t, k) = \begin{cases} \alpha & \text{if } y_t \text{ entails } k_i \text{ for some } i \in [N] \\ -\beta & \text{if } y_t \text{ contradicts } k_i \text{ for some } i \in [N] \\ \gamma \cdot r_{\text{sim}}(y_t, k) & \text{otherwise,} \end{cases} \tag{4}$$

Algorithm 1 Knowledge-Guided Decoding (KGD) Framework

```
Input: Language model $\mathcal{M}$, knowledge source $\mathcal{K}$, query $q$, maximum new tokens $T$, top-$m$, weight
    $w$, reward function $r$
Output: Generated text $x_{<T+1}$
 1: Retrieve relevant knowledge $k = \{k_i\}_{i=1}^N$ from $\mathcal{K}$ based on query $q$
 2: Initialize generated tokens $x_{<1}$ with $q$ prepended with contexts $k$
 3: for $t = 1$ to $T$ do
 4:     Compute language model logits $z(x_t|x_{<t})$ over vocabulary
 5:     Select top-$m$ tokens $\mathcal{V}_m$ based on logit values
 6:     for $x_t \in \mathcal{V}_m$ do
 7:         $y_t \leftarrow [x< t, x_t]$
 8:         Compute knowledge-guided reward $r(y_t, k)$
 9:         Update logits: $z_{\text{KGD}}(x_t|x_{<t}) \leftarrow z(x_t|x_{<t}) + w \cdot r(y_t, k)$
10:     end for
11:     Compute $p_{\text{KGD}}(x_t|x_{<t})$ by applying softmax over updated logits $z_{\text{KGD}}(x_t|x_{<t})$
12:     Sample next token $x_t \sim p_{\text{KGD}}(x_t|x_{<t})$
13:     Update generated tokens $x_{<t+1} \leftarrow [x_{<t}, x_t]$
14: end for
15: return Generated text $x_{<T+1}$
```

where α, β, and γ are positive hyperparameters that control the strength of the reward for entailment, contradiction, and similarity, respectively. In cases where there is entailment or contradiction, the combined reward follows the entailment-based reward. If neither entailment nor contradiction exists, the combined reward falls back to the similarity-based reward scaled by γ.

It is worth noting that the choice of pre-trained models for computing embeddings and entailment scores can impact the performance of the reward functions. In our experiments, the choice of models are `sentence-transformers/all-mpnet-base-v2` for sentence embeddings and `cross-encoder/nli-deberta-v3-base` for natural language inference. However, the KGD framework is flexible and can accommodate other models or reward formulations based on the specific requirements of the task and the available knowledge sources.

4.3 Computational Efficiency

The time complexity of KGD at each decoding step depends on the number of retrieved knowledge entries N and the choice of reward function. For the similarity-based reward, computing the cosine similarity between the candidate and each knowledge entry takes $O(Nd)$ time, where d is the dimensionality of the sentence embeddings. The entailment-based reward requires running the NLI model for each candidate-knowledge pair, which takes $O(NT_{\text{NLI}})$ time, where T_{NLI} is the inference time of the NLI model. The combined reward has a time complexity of $O(N(d + T_{\text{NLI}}))$. Given that the reward needs to be computed for each token in the vocabulary $\mathcal{V}$, the total complexity at each decoding step can be as high as $O(|\mathcal{V}|N(d + T_{\text{NLI}}))$, which is computationally expensive, especially for large vocabularies.

To reduce the computational overhead, we propose a simple approximation strategy. At each decoding step, instead of considering all possible tokens in the vocabulary, we only use the top-m tokens with the largest logit values as candidates for reward computation and probability adjustment. This approximation reduces the time complexity to $O(mN(d + T_{\text{NLI}}))$, where $m \ll |\mathcal{V}|$.

5 Experiments

5.1 Tasks and Datasets

In this section, we aim to demonstrate that KGD improves the factual accuracy of generated text while maintaining fluency. Following the methodology proposed by Asai et al. (2023), we consider both short-form and long-form text generation tasks, focusing on measuring the factuality and fluency of the generated output. All of our experiments are based on open-source language models and datasets. We include the prompt templates in Appendix B.

Short-form generation. We use two well-regarded open-domain question-answering datasets for a short-form generation: TriviaQA (Joshi et al., 2017) and Natural Question (Kwiatkowski et al., 2019). Each entry in these datasets comprises a question along with reference documents that contain the corresponding answer. We randomly select 1,000 samples from each dataset to streamline computation. To evaluate the system performance, we employ a best-of-n accuracy metric, where we let the model generate n outputs for each query and compute whether the gold-standard answer appears in any of the outputs. We set $n = 4$ and the maximum number of new tokens to 16, as the answers are typically short.

Long-form generation. We consider a biography generation task. Following Min et al. (2023), we use their unlabeled split, which has 500 entities, and further extract their corresponding Wikipedia pages for reference contexts. We randomly select 20 samples to streamline evaluation costs. For each sample, we generate an output with a maximum of 100 new tokens and evaluate the factual accuracy using the FActScore (Min et al., 2023) metric. FActScore breaks down a generated text into a series of atomic facts and computes the percentage of these facts that are supported by the given reference knowledge. We use the official FActScore implementation for our evaluation.

5.2 Baselines

We compare KGD with several baseline decoding strategies, including (1) greedy decoding, where at each step, the token with the highest probability is selected; (2) beam search, a breadth-first search algorithm that keeps the top k most likely sequences at each step, and (3) contrastive search, a decoding strategy that penalizes repetitions and encourages diversity in the generated text (Su et al., 2022). For each baseline, we evaluate both with and without retrieval to assess the impact of incorporating external knowledge.

To add retrieval to each decoding method, we first use a retrieval system to find relevant contexts from the knowledge source based on the input query. We then prepend the retrieved contexts to the input query before passing it to the language model for decoding. We include the prompt templates for retrieval in Appendix B.

5.3 Experimental details

For the retrieval step, we chunk potentially long documents provided in each short-form generation dataset or Wikipedia page for the biography generation task into chunks of 500 characters with an overlap of 100 characters. We then use a dense retrieval system based on the `sentence-transformers/all-mpnet-base-v2` model to retrieve relevant knowledge from these chunked documents. We encode the input query and the document chunks into dense vectors using the sentence transformer and retrieve the top N most similar chunks based on cosine similarity. The retrieved chunks are then used as the knowledge context for the subsequent generation step. We set the number of retrieved contexts used in our approach to $N = 3$.

For the short-form generation tasks, we consider Llama 7B (Touvron et al., 2023a) and Llama 2 7B (Touvron et al., 2023b) models. These models are utilized to decode using a variety of strategies, including greedy decoding, beam search with beam width 4, and contrastive search ($k = 4$ and penalty $\alpha = 0.6$). For the long-form generation task, we consider a smaller Gemma 2B model (Gemma Team and Google DeepMind, 2024). We use contrastive search as our only baseline for the long-form task to mitigate the problem of the model simply copying the knowledge context verbatim. In our KGD framework, we set the default weight parameter w to 2 for the similarity-based reward. For the entailment-based reward and the combined reward, we set $\alpha = 5$, $\beta = 10$, and $\gamma = 1.5$. We use top-m approximation with $m = 4$ to improve computational efficiency.

5.4 Results

KGD effectively improves the generation factuality. As shown in Table 1, KGD with the similarity reward outperforms the best baselines on short-form generation tasks across both datasets. For the Llama 2 7B model, KGD similarity achieves an accuracy boost of 2.0% and 3.2% on TriviaQA and Natural Question, respectively. Similar increases are observed for the Llama 7B model, with KGD similarity surpassing the best baselines. Figure 2 presents the results for long-form biography generation using the Gemma 2B model. KGD combined, which incorporates both similarity and

Table 1: Comparison of short-form generation performance across various decoding methods. Numbers indicate the best-of-4 accuracy for each model and dataset for each method. **Bold** numbers indicate the best performance for each model and dataset.

Method	**Llama 7B** TriviaQA ↑	**Llama 7B** Natural Question ↑	**Llama 2 7B** TriviaQA ↑	**Llama 2 7B** Natural Question ↑
Generation methods without retrieval				
Greedy	59.3	21.2	57.6	20.0
Beam search	62.1	25.5	62.9	24.4
Contrastive search	61.2	24.1	62.6	24.0
Generation methods with retrieval				
Greedy	65.9	39.9	63.1	34.3
Beam search	66.8	45.2	68.8	42.9
Contrastive search	67.8	42.8	68.9	40.9
KGD similarity	**69.1**	**46.4**	**70.8**	**46.1**
KGD entailment	**74.9**	**47.4**	**73.8**	**46.8**
KGD combined	**75.2**	**46.9**	**73.8**	**46.7**

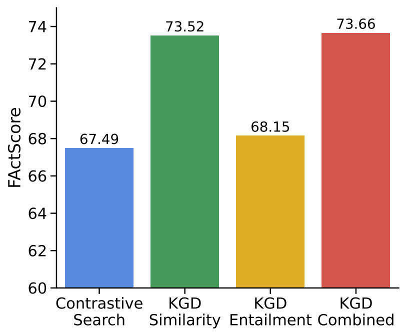


Figure 2: Comparison of long-form generation performance across various decoding methods. Numbers indicate factuality scores measured using FActScore (Min et al., 2023).

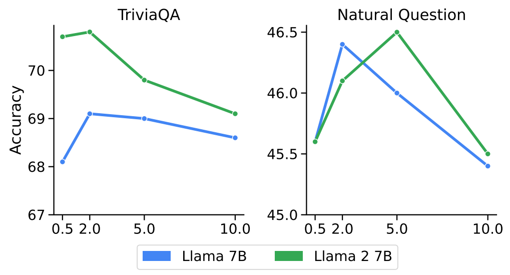


Figure 3: Ablation study on the weight parameter w in KGD similarity. The short-form generation accuracy improves as w increases from 0 to 5 but drops slightly when w becomes too large (e.g., $w = 10$).

entailment rewards, attains the highest FActScore of 73.66, outperforming KGD entailment, KGD similarity, and contrastive search (67.49). In contrast, KGD entailment achieves a more modest score of 68.15. Though still outperforming contrastive search, these results suggest that the similarity reward is particularly important for maintaining factual consistency in long-form texts.

Ablation study on the weight parameter. The weight parameter w in KGD controls the influence of the knowledge-guided rewards on the decoding process. We conduct an ablation study to investigate the impact of w on the generation performance. Figure 3 shows the results on TriviaQA and Natural Question using the Llama 7B and Llama 2 7B models with the similarity reward. As w increases from 0.5 to 2, the accuracy improves consistently, demonstrating the effectiveness of the reward in guiding the model towards more factual outputs. However, when w becomes too large (e.g., $w = 10$), the accuracy drops slightly. We observe a similar pattern in the long-form generation task, as shown in Figure 4. The FActScore improves as w increases from 0 to 2, highlighting the benefits of incorporating knowledge-guided rewards in enhancing the factual accuracy of the generated biographies. However, when w becomes excessively large, the performance slightly deteriorates, indicating that overemphasizing the reward may hinder the model's ability to generate fluent text. Based on these findings, we set $w = 2$ as the default value in our experiments to strike a balance between factuality and fluency.

Ablation study on α and β. We further examine the effect of the hyperparameters α and β in the KGD combined approach, which controls the relative importance of the entailment reward. Figure 5 illustrates the FActScore on the long-form biography generation task using the Gemma 2B model as we vary α and β in $\{1, 5, 10, 20, 40\}$ and $\gamma = 1.5$. The results show that the choice of hyperparameters does not yield a clear pattern in performance across different values. Given the randomness involved, careful hyperparameter tuning may be needed to optimize the impact

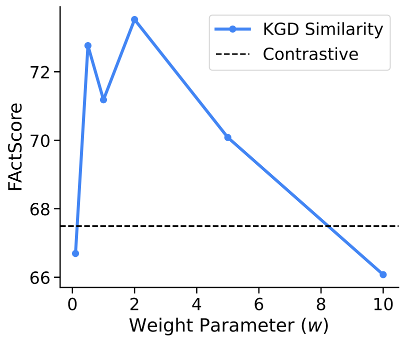


Figure 4: Ablation study on the weight parameter w in KGD similarity. The long-form generation accuracy improves as w increases from 0 to 2 but drops slightly when w becomes too large (e.g., $w = 10$).

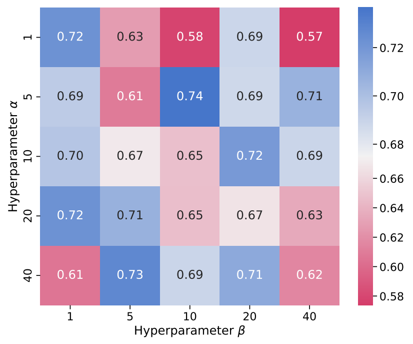


Figure 5: Ablation study on the α and β hyperparameters in KGD combined. Blue and red cells indicate FActScore performance better and worse than the contrastive search baseline, respectively. γ is fixed at 1.5.

of the entailment reward. This suggests that the KGD entailment approach may not consistently provide benefits for this task without further refinement, which is similar to the findings in short-form generation.

6 Qualitative Analysis

To gain deeper insights into the behavior and capabilities of our Knowledge-Guided Decoding (KGD) framework, we conduct a qualitative analysis by examining the generated outputs and comparing them with the baseline methods. We aim to understand the strengths and limitations of KGD and identify scenarios where it succeeds or fails in generating factually accurate and fluent text.

Generation quality. Comparing the KGD outputs to the contrastive search baseline, we observe that KGD is generally effective at incorporating the provided knowledge to generate more factual text. For example, in the Suthida Bajrasudhabimalalakshana biography (Figure 6), KGD correctly states her birthplace, graduation from Assumption University, and role as a flight attendant before marrying the King. The baseline omits or misrepresents some of these key facts. However, KGD struggles with accurately incorporating dates, as seen in the Felix Gallardo KGD similarity example (Figure 7). The dates in this case were notably confused, underscoring a potential flaw in KGD's reward mechanism. These errors suggest that the reward mechanism used in KGD does not directly enforce accurate representation and integration of dates and may allow for the hallucination of unsupported facts.

Output length. It is worth noting that the KGD combined approach generates an average of 16 facts per output, compared to 16.2 facts for the contrastive search baseline. This small difference in the number of facts, along with the slightly shorter responses in KGD's outputs, suggests that the observed performance improvements are primarily due to the increased factual accuracy of the generated facts rather than a significant difference in the quantity of information provided. While shorter responses may contribute to KGD's efficacy by reducing the potential for hallucination, the qualitative analysis indicates that the knowledge-guided rewards play a crucial role in enhancing the factuality of the generated text.

7 Conclusion

In this work, we introduced Knowledge-Guided Decoding (KGD), a novel framework for enhancing the factuality of text generation by leveraging retrieved knowledge at the token level. Through experiments on short-form and long-form generation tasks, we demonstrated that KGD effectively improves the factual accuracy of the generated text while maintaining fluency, outperforming several strong baselines. However, our ablation studies and qualitative analysis revealed some limitations, such as the sensitivity to hyperparameters and the difficulty in accurately representing and integrating dates. Future work could explore more sophisticated reward formulations that better capture the temporal and numerical aspects of the retrieved knowledge. Despite these limitations, KGD presents a promising direction for improving the reliability and trustworthiness of language generation systems by leveraging retrieved knowledge at the token level.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2022. PPL-MCTS: constrained textual generation through discriminator-guided MCTS decoding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2953–2967. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations.

Gemma Team and Google DeepMind. 2024. Gemma: Open models based on gemini research and technology.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Grace: Discriminator-guided chain-of-thought reasoning.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. Args: Alignment as reward-guided search. In *Proceedings of the International Conference on Learning Representations*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. In *Empirical Methods in Natural Language Processing*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023a. Contrastive decoding: Open-ended text generation as optimization. In *Association for Computational Linguistics*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202. PMLR.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback.

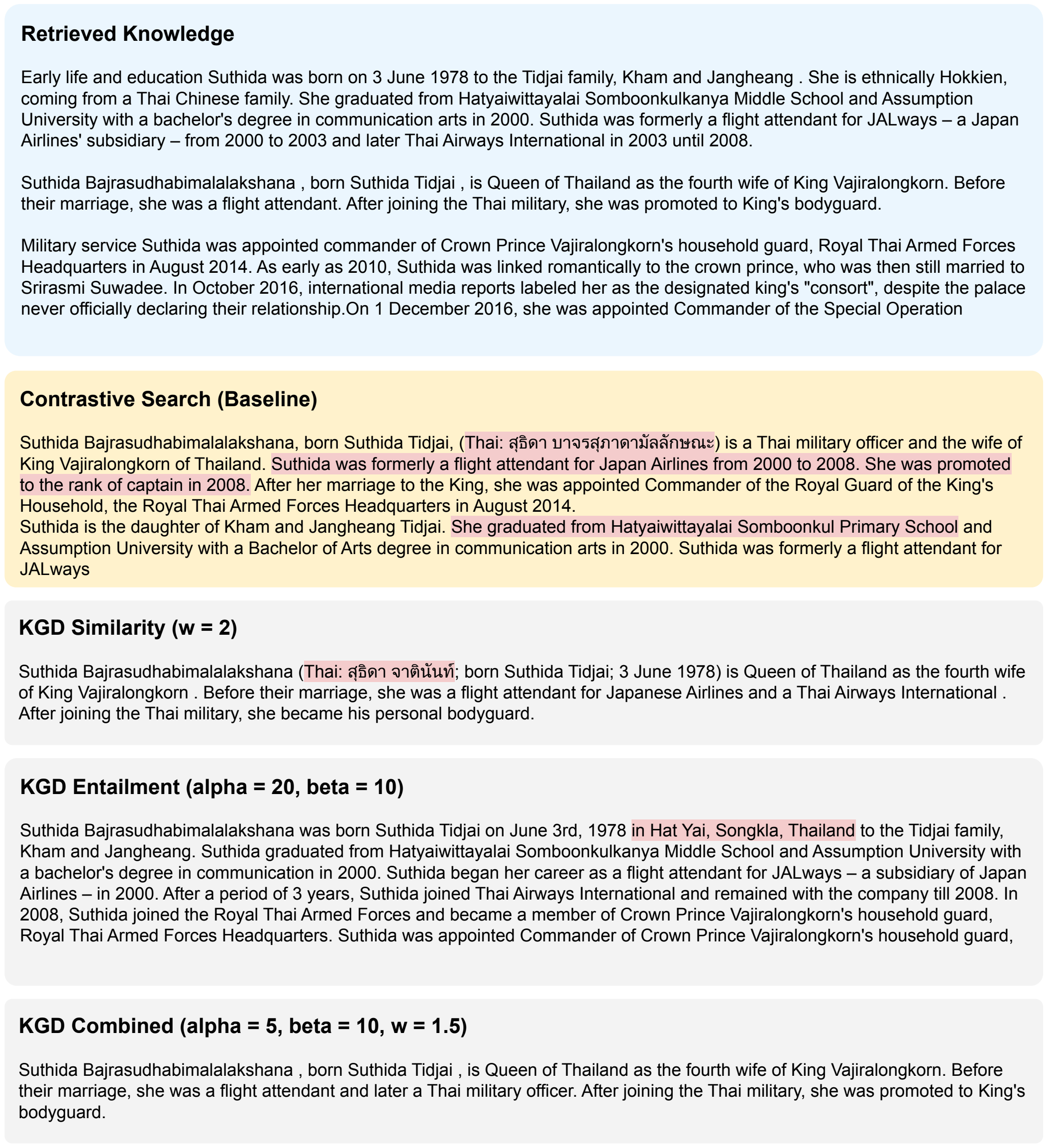
Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems*.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Decomposition enhances reasoning via self-evaluation guided decoding.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate problem solving with large language models.

A Qualitative Examples



Retrieved Knowledge

Early life and education Suthida was born on 3 June 1978 to the Tidjai family, Kham and Jangheang . She is ethnically Hokkien, coming from a Thai Chinese family. She graduated from Hatyaiwittayalai Somboonkulkanya Middle School and Assumption University with a bachelor's degree in communication arts in 2000. Suthida was formerly a flight attendant for JALways – a Japan Airlines' subsidiary – from 2000 to 2003 and later Thai Airways International in 2003 until 2008.

Suthida Bajrasudhabimalalakshana , born Suthida Tidjai , is Queen of Thailand as the fourth wife of King Vajiralongkorn. Before their marriage, she was a flight attendant. After joining the Thai military, she was promoted to King's bodyguard.

Military service Suthida was appointed commander of Crown Prince Vajiralongkorn's household guard, Royal Thai Armed Forces Headquarters in August 2014. As early as 2010, Suthida was linked romantically to the crown prince, who was then still married to Srirasmi Suwadee. In October 2016, international media reports labeled her as the designated king's "consort", despite the palace never officially declaring their relationship.On 1 December 2016, she was appointed Commander of the Special Operation

Contrastive Search (Baseline)

Suthida Bajrasudhabimalalakshana, born Suthida Tidjai, (Thai: สุธิดา บาจรสุภาดามัลลักษณะ) is a Thai military officer and the wife of King Vajiralongkorn of Thailand. Suthida was formerly a flight attendant for Japan Airlines from 2000 to 2008. She was promoted to the rank of captain in 2008. After her marriage to the King, she was appointed Commander of the Royal Guard of the King's Household, the Royal Thai Armed Forces Headquarters in August 2014.
Suthida is the daughter of Kham and Jangheang Tidjai. She graduated from Hatyaiwittayalai Somboonkul Primary School and Assumption University with a Bachelor of Arts degree in communication arts in 2000. Suthida was formerly a flight attendant for JALways

KGD Similarity (w = 2)

Suthida Bajrasudhabimalalakshana (Thai: สุธิดา จาตินันท์; born Suthida Tidjai; 3 June 1978) is Queen of Thailand as the fourth wife of King Vajiralongkorn . Before their marriage, she was a flight attendant for Japanese Airlines and a Thai Airways International . After joining the Thai military, she became his personal bodyguard.

KGD Entailment (alpha = 20, beta = 10)

Suthida Bajrasudhabimalalakshana was born Suthida Tidjai on June 3rd, 1978 in Hat Yai, Songkla, Thailand to the Tidjai family, Kham and Jangheang. Suthida graduated from Hatyaiwittayalai Somboonkulkanya Middle School and Assumption University with a bachelor's degree in communication in 2000. Suthida began her career as a flight attendant for JALways – a subsidiary of Japan Airlines – in 2000. After a period of 3 years, Suthida joined Thai Airways International and remained with the company till 2008. In 2008, Suthida joined the Royal Thai Armed Forces and became a member of Crown Prince Vajiralongkorn's household guard, Royal Thai Armed Forces Headquarters. Suthida was appointed Commander of Crown Prince Vajiralongkorn's household guard,

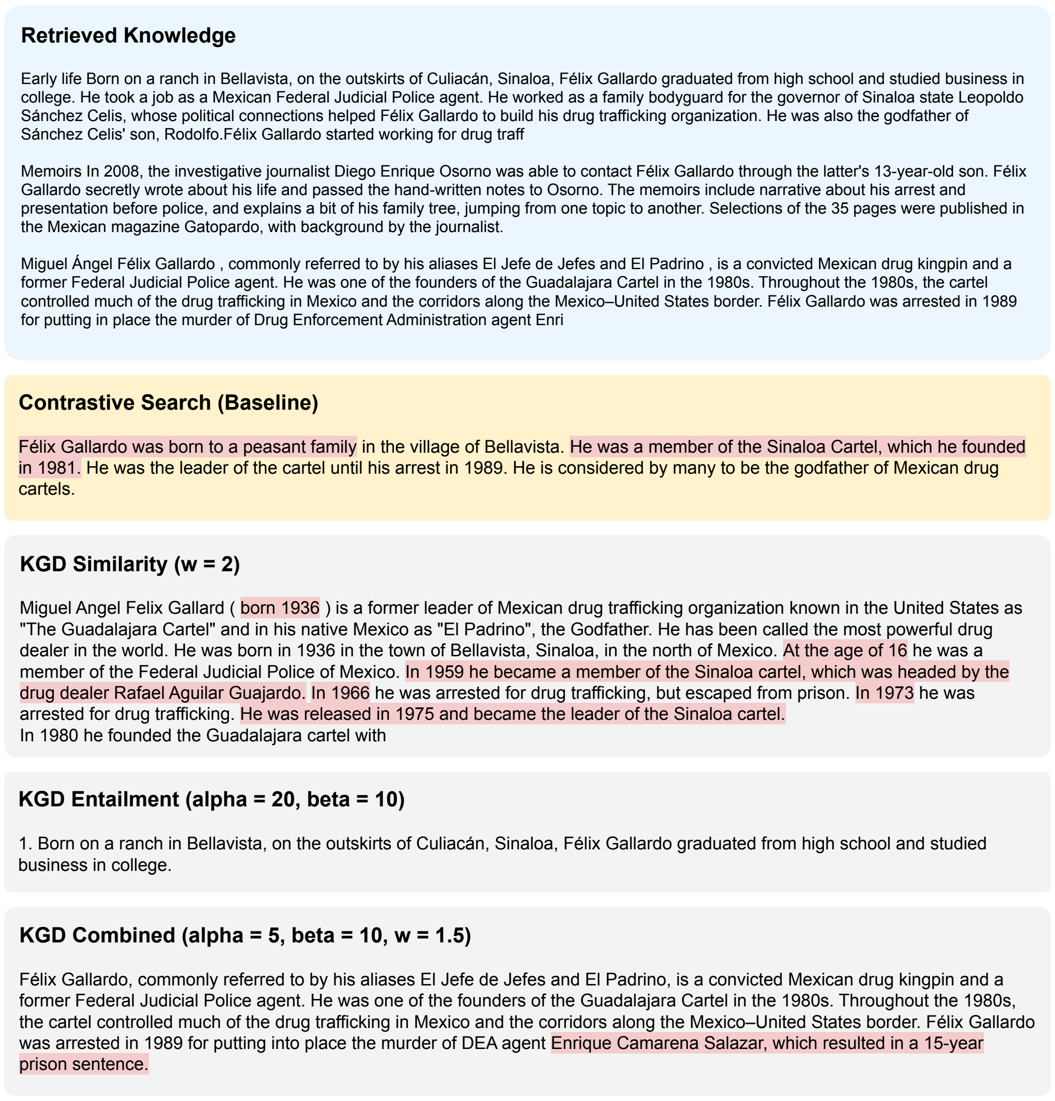
KGD Combined (alpha = 5, beta = 10, w = 1.5)

Suthida Bajrasudhabimalalakshana , born Suthida Tidjai , is Queen of Thailand as the fourth wife of King Vajiralongkorn. Before their marriage, she was a flight attendant and later a Thai military officer. After joining the Thai military, she was promoted to King's bodyguard.

Figure 6: Comparison of the retrieved knowledge (top), contrastive search baseline output (second), KGD similarity output (third), KGD entailment output (fourth), and KGD combined output (bottom) for the biography of Suthida Bajrasudhabimalalakshana. The outputs are generated based on the given context, with varying levels of accuracy and detail. Red text indicates inconsistencies with the retrieved knowledge.

B Generation Prompts

In this section, we provide the prompts used for generating text in our experiments. Figure 8 shows the prompt templates for both short-form question answering (QA) and long-form biography generation tasks. For each task type, we consider two variants of the prompt: one that includes the retrieved context information and one that does not. The context-enriched prompts contain placeholders for inserting the retrieved context (`contexts`) and the query or entity name (`question`). We also include prefix-only variants of the prompts, which omit the query/question and allow the model to continue generating text after the inserted context. These different prompt formulations enable us to study the impact of incorporating retrieved knowledge on the factuality and fluency of the generated text across various task settings.



Retrieved Knowledge

Early life Born on a ranch in Bellavista, on the outskirts of Culiacán, Sinaloa, Félix Gallardo graduated from high school and studied business in college. He took a job as a Mexican Federal Judicial Police agent. He worked as a family bodyguard for the governor of Sinaloa state Leopoldo Sánchez Celis, whose political connections helped Félix Gallardo to build his drug trafficking organization. He was also the godfather of Sánchez Celis' son, Rodolfo.Félix Gallardo started working for drug traff

Memoirs In 2008, the investigative journalist Diego Enrique Osorno was able to contact Félix Gallardo through the latter's 13-year-old son. Félix Gallardo secretly wrote about his life and passed the hand-written notes to Osorno. The memoirs include narrative about his arrest and presentation before police, and explains a bit of his family tree, jumping from one topic to another. Selections of the 35 pages were published in the Mexican magazine Gatopardo, with background by the journalist.

Miguel Ángel Félix Gallardo , commonly referred to by his aliases El Jefe de Jefes and El Padrino , is a convicted Mexican drug kingpin and a former Federal Judicial Police agent. He was one of the founders of the Guadalajara Cartel in the 1980s. Throughout the 1980s, the cartel controlled much of the drug trafficking in Mexico and the corridors along the Mexico–United States border. Félix Gallardo was arrested in 1989 for putting in place the murder of Drug Enforcement Administration agent Enri

Contrastive Search (Baseline)

Félix Gallardo was born to a peasant family in the village of Bellavista. He was a member of the Sinaloa Cartel, which he founded in 1981. He was the leader of the cartel until his arrest in 1989. He is considered by many to be the godfather of Mexican drug cartels.

KGD Similarity (w = 2)

Miguel Angel Felix Gallard (born 1936) is a former leader of Mexican drug trafficking organization known in the United States as "The Guadalajara Cartel" and in his native Mexico as "El Padrino", the Godfather. He has been called the most powerful drug dealer in the world. He was born in 1936 in the town of Bellavista, Sinaloa, in the north of Mexico. At the age of 16 he was a member of the Federal Judicial Police of Mexico. In 1959 he became a member of the Sinaloa cartel, which was headed by the drug dealer Rafael Aguilar Guajardo. In 1966 he was arrested for drug trafficking, but escaped from prison. In 1973 he was arrested for drug trafficking. He was released in 1975 and became the leader of the Sinaloa cartel.
In 1980 he founded the Guadalajara cartel with

KGD Entailment (alpha = 20, beta = 10)

1. Born on a ranch in Bellavista, on the outskirts of Culiacán, Sinaloa, Félix Gallardo graduated from high school and studied business in college.

KGD Combined (alpha = 5, beta = 10, w = 1.5)

Félix Gallardo, commonly referred to by his aliases El Jefe de Jefes and El Padrino, is a convicted Mexican drug kingpin and a former Federal Judicial Police agent. He was one of the founders of the Guadalajara Cartel in the 1980s. Throughout the 1980s, the cartel controlled much of the drug trafficking in Mexico and the corridors along the Mexico–United States border. Félix Gallardo was arrested in 1989 for putting into place the murder of DEA agent Enrique Camarena Salazar, which resulted in a 15-year prison sentence.

Figure 7: Comparison of the retrieved knowledge (top), contrastive search baseline output (second), KGD similarity output (third), KGD entailment output (fourth), and KGD combined output (bottom) for the biography of Felix Gallardo. The outputs are generated based on the given context, with varying levels of accuracy and detail. Red text indicates inconsistencies with the retrieved knowledge.

Short-form: QA Task Prompt

```
Context information is below.
---------------------
{retrieved context 1}

{retrieved context 2}

{retrieved context 3}
---------------------
Given the context information and not prior knowledge, answer the query.
Query: {question}
Answer:
```

Long-form: Biography Generation Task Prompt

```
[Context 1]
{retrieved context 1}

[Context 2]
{retrieved context 2}

[Context 3]
{retrieved context 3}
---------------------
Question: Tell me an at least 50 words bio of {entity}. You must summarize but not directly copy the
answer words by words from the contexts.
Answer:
```

Figure 8: Prompts used for short-form QA and long-form biography generation tasks. For each task type, we show the prompt templates with and without retrieved context. The retrieved context is inserted into the prompt at the placeholder `contexts`, while the query or entity name is inserted at `question`. The prefix-only variants omit the query/question to allow the model to continue generating after the inserted context.