ChainRank-DPO: Chain Rank Direct Preference Optimization for LLM Rankers

Anonymous ACL submission

Abstract

In information retrieval, large language models (LLMs) such as GPT-4 and open-source alternatives like RankVicuna and RankZephyr have proven essential for reranking tasks. However, the gap between ranking and language modeling persists, leading to degraded general capabilities of LLMs in pursuit of specialized reranking tasks through SFT. To overcome this challenge, we introduce ChainRank – a new open-source LLM ranker for zero-shot listwise reranking, which not only achieves comparable (even better) performance over the closed-source RankGPT-3.5, but also maintains the general LLM capabilities. Experiment results on TREC 2019 and 2020 Deep Learning Tracks show that our model achieves a higher score than RankVicuna with less distillation data, and benefits from our Chain-of-Thoughts reranking prompt and DPO training, our model obtains higher scores than other fine-tuned reranking LLMs on Massive Multitask Language Understanding.

1 Introduction

Text reranking is essential in information retrieval, dialogue systems, and content summarization. It involves refining the ordering of a set of candidate texts to identify the most relevant ones in response to a query. Traditional reranking methods can be divided into pointwise, pairwise, and listwise approaches. In listwise reranking, the model considers the entire set of candidates when determining their order, optimizing the list as a whole rather than focusing on individual items or pairs.

In a typical retrieval scenario, given a corpus $C = \{D_1, D_2, ..., D_n\}$ containing a collection of documents and a query q, the retriever's task is to efficiently return a subset of k documents from C that are most relevant to the query according to some metric, such as normalized Discounted Cumulative Gain (nDCG). The reranker then takes this list that the retriever has created and further improves its quality, aiming to reorder the documents to maximize relevance by the selected evaluation metric.

Following the emergence of large language models (LLMs), such as GPT-4 (Achiam et al., 2023), researchers have investigated utilizing text comprehension and reasoning abilities of LLMs for zero-shot listwise text reranking (Ma et al., 2023). Among these efforts, RankGPT (Sun et al., 2023) stands out as an effective approach, directly generating a final ranking for multiple passages and optimizing for more accurate results. While RankGPT$_{3.5}$ and RankGPT$_4$ achieve state-of-art performance there are still two major limitations. Firstly, like other LLM-based models, RankGPT requires significant inference costs when ranking a large set of candidate documents. The real-time rerankinging can quickly become expensive, as cloud-based inference on LLMs can cost several dollars per thousand queries. The high cost makes RankGPT less feasible for use cases that require frequent, large-scale ranking operations. Secondly,

USER: I will provide you with {num} passages, each indicated by a numerical identifier []. Rank the passages based on their relevance to the search query: {query}.
[1] {passage 1}
[2] {passage 2}
...
[{num}] {passage {num}}
Search Query: {query}.
Rank the {num} passages above based on their relevance to the search query. All the passages should be included and listed using identifiers. Rank the passages step by step based on the descending relevance. For example:
step 1: [4]
step 2: [4,2]
step 3: [4,2,3]
...
step {num}: [4,2,3,15,...,14]
final answer: [4,2,3,15,...,14].
Only respond with the chain of thoughts, do not say any word or explain, please include all the steps in your response.

Figure 1: CoT reranking prompt for ChainRank.

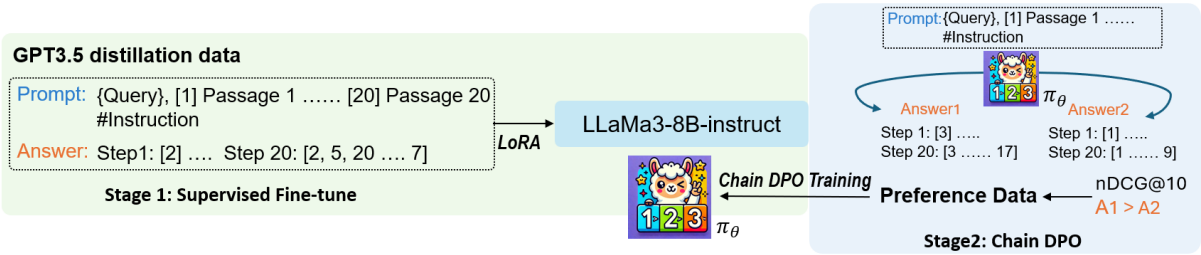

Figure 2: Training framework of our model. $Left$: Stage 1 aims to teach the student model to rank with LoRA. $Right$: In stage 2, we use the same prompt format to generated multiple answers, then pick the chosen and rejected answers based on nDCG@10, finally, we utilize the preference data to perform Chain DPO training.

transformer-based models like GPT, have a fixed input length limit. when the list of candidate passages is large, it is necessary to use the sliding window strategy (Sun et al., 2023).

To address the limitation of inference cost and speed, RankVicuna (Pradeep et al., 2023a) and RankZephyr (Pradeep et al., 2023b) were introduced by leverage instruction fine-tuning to distill the listwise reranking effectiveness of RankGPT down to the 7b Vicuna model and Zephyr model. Both of them are open-source LLaMA variants. With RankGPT$_{3.5}$ distillation data, the effectiveness of RankVicuna is on par with RankGPT$_{3.5}$ and slightly worse than RankGPT$_4$, while RankZephyr bridges the gap with 5K more query-candidates pairs generated by RankGPT$_4$.

Our investigations revealed that following fine-tuning for listwise ranking, RankVicuna and RankZephyr forfeited their general text-generating capabilities. Irrespective of the input prompt, they might alone generate a series of numbers. Furthermore, while they exhibited comprehension of listwise ranking, they encountered difficulties with pairwise ranking, as their pairwise outcomes did not correspond with those generated by the listwise ranking.

To resolve these challenges, we provide our approach, ChainRank, which is built upon the LLaMA3-8b-instruct (Dubey et al., 2024). Utilizing the identical training data as RankVicuna, we developed an innovative chain-of-thought prompt to direct the LLM in sequentially ranking passages. Subsequently, we implement Chain-DPO to augment the reasoning capabilities of the fine-tuned model, hence providing superior and precise ranking performance. While Direct Preference Optimization (DPO) (Rafailov et al., 2024) and its variants (Zhao et al., 2024) (Pang et al., 2024)are effective for chat benchmarks, it offers minimal improvements for long-chain reasoning tasks like math and ranking problems, where errors often occur midway through the process. Rejecting entire answers can discard correct early steps, adding noise to training. Step-DPO (Lai et al., 2024) addresses this by focusing on specific erroneous steps, enabling the model to identify and correct mistakes more efficiently. We applied a similar paradigm as Step-DPO, but chose nDCG@10 as the reward function to identify the chosen and rejected answers. In our experiments, We found that our model retains its general text generation capabilities, effectively addressing the limitations observed in RankVicuna and RankZephyr. Also, ChainRank outperforms both RankVicuna and the teacher model, RankGPT, on the TREC 2019 and 2020 Deep Learning Tracks (Craswell et al., 2021, 2020). We summarize the contributions of this work as follows:

- We propose a novel chain-of-thoughts instruction (reranking) tuning that enables LLMs to rank passages based on relevance step by step.
- To our best knowledge, this is the first work to introduce the Supervised Fine-Tune—-DPO pipeline in the context of text reranking.
- Using the same training data, our model outperforms both RankVicuna and the teacher model, RankGPT-3.5, while retaining the general capabilities of LLMs.

2 Methodology

2.1 Prompt design

Recent listwise text reranking work (Pradeep et al., 2023a) (Pradeep et al., 2023b) (Liu et al., 2024) are build on the zero-shot listwise approaches (Ma et al., 2023) (Sun et al., 2023). All of them use similar prompt templates and define their problem

as follows: Given a query q and candidates documents $D_1, D_2, ..., D_n$ from the retrieval stage, return a reordered list of the input identifiers, e.g., $[1] > [2] > [3] .... > [20]$.

We consider listwise text reranking to be a chain-of-thoughts (CoT) reasoning problem. In each step, we select the most relevant document from the candidate list and add it to the reordered list. To achieve this, our prompt instructs the LLM to complete the reranking process incrementally, step by step until all documents are ranked. The exact CoT reranking prompt to LLaMa is shown in Figure 1.

Similar to RankVicuna, we prepend the system message used in training LLaMA to our prompt, aiming for more efficient training and improved generation accuracy.

2.2 Training Stage 1: Supervised Fine-Tune

As Figure 2 shows, during the Supervised Fine-Tune (SFT) phase, we utilize the ranked list generated by $\text{RankGPT}_{3.5}$ for 100k queries provided by RankGPT. We randomly selected 80% of the 100k query-candidates pair as our training dataset for the SFT stage. Since neither $\text{RankGPT}_{3.5}$ nor ChainRank requires human-labeled data, this approach can be identified as zero-shot. More details about our training dataset can be found in Appendix B.

Before training, the original LLaMa3-8B-Instruct model (Dubey et al., 2024) was unable to generate meaningful results using our CoT reranking prompt; all outputs simply followed the same order as the provided format example. Therefore, the fine-tune process is necessary. The original weights can be found under $meta{-}llama/Meta{-}Llama-3-8B-Instruct$ on the HuggingFace website. [1]

We fine-tuned our 8B parameter ChainRank with Low-Rank Adaption (LoRA) (Hu et al., 2021) for one epoch, with the effective bath size of 128 and learning rate of 1×10^{-6} in bfloat16. The training process was performed on 4 NVIDIA A100 80GB GPUs for approximately 39 hours.

2.3 Training Stage 2: Chain-DPO

After completing Stage 1, we prompt our ChainRank-SFT model to generate three ranking predictions y given the remaining 20% distillation permutations input x. We then selected the accepted y_w and rejected y_l instances based on the calculated nDCG@10 relative to the ground truth label, resulting in a preference dataset that includes prompts, chosen answers, and rejected answers. Unlike previous work, our final ranking output y consists of a sequence of n reasoning steps y = $\{s_1, s_2, \cdots, s_n\}$, where each step returns only one ranking conditioned on previous reasoning steps $\pi(s_k \mid x; s_{1:k-1})$. We then design our objective function to maximize the likelihood of the winning step s_w and minimize loss step s_l as,

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,s_w,s_l,s_{1:k-1})\sim D}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(s_w \mid x; s_{1:k-1})}{\pi_{ref}(s_w \mid x; s_{1:k-1})} - \beta\log\frac{\pi_\theta(s_l \mid x; s_{1:k-1})}{\pi_{ref}(s_l \mid x; s_{1:k-1})}\Big)\Big],$$

where $(x, s_w, s_l, s_{1:k-1})$ denotes prompt, desirable reasoning steps, undesirable reasoning steps, and overlapping steps.

We compared each reasoning step and recorded the overlapping steps s_o until the first divergent step was found. Subsequent steps were marked as chosen steps s_w and rejected steps s_l. Therefore, we have a dataset consisting of $(x, y_w, y_l, s_w, s_l, s_o)$. This phase aims to maximize the probability of the correct next reasoning step and minimize the probability of the incorrect one. Similar to DPO objective, we aim to optimize the probability of s_w and minize the probability of s_l, as the function above shows. The Chain-DPO stage was performed on 4 NVIDIA A100 80GB GPUs for one epoch, which requires 8 hours.

3 Experiments

3.1 Ranking Evaluation

We leverage TREC DL19 (Craswell et al., 2020) and DL20 (Craswell et al., 2021) Tracks as our ranking capability evaluation datasets. Both of them are retrieved from MS MARCO V1 (Bajaj et al., 2016) and have human-annotated relevance labels. TREC DL19 and DL20 test sets contain 43 and 54 queries respectively, each query consists of 100 candidate passages. Building on the evaluations conducted with RankZephyr (Pradeep et al., 2023b), we included two out-of-scope datasets: NEWS and COVID from the BEIR (Thakur et al., 2021) benchmark, to provide greater variety in our analysis. Due to the context window limit, we have to use the sliding window strategy to rerank 100 passages, with window size of 20 and stride size of 10.

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Models	TREC		BEIR		MMLU
	DL19	DL20	NEWS	COVID	AVG.
RankGPT$_{3.5}$	0.658	0.629	0.481	0.751	0.700
BM25	0.506	0.480	0.395	0.595	N/A
Contriever	0.616	0.599	0.451	0.626	N/A
Gemma-7B	0.533	0.530	0.396	0.573	0.649
LLaMa3	0.641	0.621	0.395	0.628	0.662
RankZephyr	0.742	0.709	0.396	0.592	0.000
RankVicuna	0.668	0.655	0.393	0.592	0.373
RankLLaMa3	0.666	0.635	0.395	0.662	0.628
ChainRank	0.691	0.657	0.396	0.717	0.663
ChainRank-DPO	0.691	0.663	0.396	0.721	0.662

Table 1: Performance of different models on TREC (nDCG@10) and MMLU (exact match score) benchmarks. All the reranking tasks are based on BM25 retrieval results.

3.2 Massive Multitask Language Understanding

We employed the Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) benchmark to assess whether our model maintains its text understanding abilities after being fine-tuned for ranking tasks. MMLU is designed to test a model's proficiency across 57 diverse subjects, ranging from the humanities to STEM fields, covering elementary-level knowledge up to advanced professional topics. We evaluated our model's performance on MMLU against baseline LLMs to ensure that fine-tuning for ranking did not compromise its ability to generate complex, open-ended text. This comprehensive evaluation allows us to assess our model's retention of general-purpose language understanding and generation capabilities alongside other baselines, ensuring its versatility in ranking and broader language tasks.

3.3 Results

Table 1 compares the performance of different models on three major benchmarks, TREC, BEIR, and MMLU. Among open-source models, our 8B ChainRank model outperforms RankVicuna, even with only 80% of the training data after completing just one epoch of Stage 1 training. Additionally, it surpasses the 175B teacher model RankGPT$_{3.5}$ [2] on both DL19 and DL20 benchmarks. Furthermore, compared to RankLLaMa3, ChainRank achieves a higher score under the same training conditions, demonstrating the effectiveness of our CoT reranking prompt. In the DPO training stage, ChainRank-DPO maintains the same performance on DL19 and shows improvement on DL20 with just one epoch of training. More importantly, as Appendix C shows, DPO training enhances the model's robustness when varying the number of passages passed to it.

On the BEIR dataset (Thakur et al., 2021), RankGPT$_{3.5}$ outperforms all other models, likely due to its training on related data. Our model achieved the highest score on the COVID dataset, while all open-source LLMs performed similarly on the NEWS dataset. These results demonstrate that our method can enhance the model's reasoning capability in ranking, even on out-of-scope evaluation datasets.

Regarding MMLU performance, ChainRank maintains results that are on par with the base model LLaMa3. In contrast, RankLLaMa3 shows a slight drop in performance, while RankVicuna exhibits significantly lower scores. Additionally, since RankZephyr has completely lost its ability to generate meaningful outputs, it receives a score of 0 on MMLU. More examples are provided in Appendix D .

4 Conclusion and Future Work

In this paper, we proposed a novel approach, ChainRank, a novel zero-shot listwise text reranking model built on LLaMa3. We demonstrated the effectiveness of CoT reranking prompt and our SFT—Chain DPO pipeline. The results of our experiments show that our model achieves superior performance compared to many open-source and closed-source LLMs, such as RankVicuna and RankGPT$_{3.5}$.

In future work, we plan to explore the application of our method to other models, such as Mistral, Zephyr, and LLaMa3.1, to evaluate its generalizability and performance across different architectures. Additionally, our current approach utilized only the RankGPT$_{3.5}$ distillation permutations, which involved a fixed number of passages per instance. Future research could incorporate higher-quality datasets with varying numbers of passages per instance to enhance the diversity and robustness of the training data. Expanding the dataset in this manner could lead to improved overall performance.

[2] RankGPT is the only closed-source model and the biggest model in the table.

5 Limitations

While this study provides valuable insights, several limitations should be acknowledged. Firstly, the need to include long examples in our prompts leads to increased training and inference times.

Secondly, although LLaMa3 has a larger context window (8192 tokens) compared to Vicuna and Zephyr (4096 tokens), we are still limited by the maximum number of passages in our training data, which is set to 20. As a result, we are restricted to ranking within a window size of 20 and require nine passes to complete the reranking of 100 passages.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. *Preprint*, arXiv:2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

Qi Liu, Bo Wang, Nan Wang, and Jiaxin Mao. 2024. Leveraging passage embeddings for efficient listwise reranking with large language models. *arXiv preprint arXiv:2406.14848*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Yang Zhao, Yixin Wang, and Mingzhang Yin. 2024. Ordinal preference optimization: Aligning human preferences via ndcg. *Preprint*, arXiv:2410.04346.

A Baseline Models

To illustrate the efficacy of ChainRank, we selected several baseline models, including the existing supervised ranking method (BM25 (Robertson and Zaragoza, 2009)and Contriever (Izacard et al., 2021)), unsupervised LLM-based models (RankGPT$_{3.5}$ and Gemma-7B (Team et al., 2024)), LLM-based models trained with distillation (RankVicuna, RankZephyr, and PE Rank (Liu et al., 2024)), as well as our base model LLaMa3-8B-Instruct. To show the effectiveness of our CoT reranking prompt, we trained a model with a similar paradigm to RankVicuna but used our base model LLaMa3, denoted as RankLLaMa3.

B Training Dataset

To create this dataset, the authors randomly selected 100K queries from the MS MARCO v1 passage ranking training set and used Pyserini (Lin et al., 2021) to retrieve 20 candidates for each query using BM25. These candidates were then passed to RankGPT$_{3.5}$ to produce teacher orderings. The distillation permutations aim to minimize the disparity between RankGPT$_{3.5}$ and our 8B student model, ChainRank. To enhance the dataset quality, the authors of RankVicuna have removed the malformed generations, which are 12% of the outputs. Most of these generations are missing numerical identifiers or existing duplicated identifiers.

C Experiments of Different Window Size

Models	**Window/Stride Size**		
	2/1	**10/5**	**20/10**
Gemma-7B	0.541	0.593	0.533
LLaMa3	0.549	0.660	0.641
RankZephyr	0.562	0.612	0.742
RankVicuna	0.551	0.655	0.668
RankLLaMa3	0.567	0.663	0.666
ChainRank	0.549	0.667	0.691
ChainRank-DPO	0.603	0.691	0.691

Table 2: Performance of different models with different window size and step size, BM25 performs all of the retrieval stage, and all of the evaluations are performed on DL19

D Examples of general LLM tasks

To illustrate the degradation in general text understanding and generation capabilities of RankVicuna and Rankzephyr, we provide two examples below. Due to the length of ChainRank-DPO's response, we have omitted the middle portion of the answer.

D.1 Example 1

Prompt

Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.

RankVicuna

Ultimately, showcasing Hawaii's rich history and cultural significance.
Title: Embracing Hawaiian Culture: Discovering Historical Landmarks and Rich Traditions on the Big Island

RankZephyr

[1] > [2] > [3] > [4] > [5] > [6] > [7] > [8] > [9] > [10] > [2] > [11] > [12] > [13]

ChainRank-DPO

Aloha and Mahalo: Discovering the Aloha Spirit in Hawaii
As I stepped off the plane in Honolulu, I couldn't help but feel a sense of excitement and anticipation wash over me. I had dreamed of visiting Hawaii for years, and finally, I was here. The warm tropical air enveloped me, and the sweet scent of plumeria filled my senses as I made my way to my hotel. Little did I know, this trip would be a journey not only to breathtaking natural wonders but also to a rich cultural heritage that would leave me feeling grateful and inspired.
Immersing in the Local Culture:
(Content omitted for brevity.)
Must-See Attractions:
(Content omitted for brevity.)
A Taste of Aloha:
(Content omitted for brevity.)
Conclusion:
(Content omitted for brevity.)
Practical Tips and Recommendations:

- Visit Pearl Harbor and USS Arizona Memorial in the morning to avoid the crowds.
- Try local specialties like poke, laulau, and shave ice.
- Take a surf lesson or snorkeling trip to explore the islands' stunning coastlines.
- Visit the Bishop Museum to learn about the history and geology of Hawaii.
- Explore the islands' many farmers' markets for fresh produce and local products.
- Learn traditional Hawaiian crafts, like leis and ukulele strings, from local artisans.

Mahalo (thank you) for joining me on this journey to the Aloha State. I hope my story has inspired you to plan your own trip to Hawaii and experience the warmth and beauty of this incredible destination.

D.2 Example 2

Prompt

Can you help me write a resignation letter to my current employer, while leaving on good terms and expressing gratitude for the opportunities provided?

RankVicuna

(Nothing was generated from the model.)

RankZephyr

[1] > [2] > [3] > [4] > [5] > [6] > [7] > [8] > [9] > [10] > [2] > [3] > [4] > [5] > [6]

ChainRank-DPO

I'd be happy to help you write a resignation letter to your current employer. Here's a sample template you can use as a guide:

Dear [Recipient's Name],
I am writing to inform you of my decision to resign from my position as [Your Position] at [Company Name], effective [Date of Last Day of Work]. I want to express my deepest gratitude to you and the entire [Company Name] team for the opportunities I've had during my time here.
(Content omitted for brevity.)
Please don't hesitate to reach out to me if you need anything after my departure. I can be reached at [Your Email Address] or [Your Phone Number].
Sincerely,
[Your Signature]

Some tips to keep in mind when writing your resignation letter:

- Be professional and respectful: Use a formal tone and language throughout the letter.
- Express gratitude: Let your employer know how much you appreciate the opportunities you've had and the experience you've gained.
- Offer to help with the transition: Show that you're committed to making the transition as smooth as possible by offering to complete outstanding tasks and assist with the handover of your responsibilities.
- Keep it brief: Aim for a letter that is no more than one page in length.
- Proofread: Make sure to proofread your letter carefully to ensure it is free of errors and typos.

Remember to customize the letter to fit your own circumstances and experiences. Good luck with your resignation, and I hope it leads to a positive and successful transition for you!