Synthetic Knowledge Ingestion: Towards Knowledge Refinement and Injection for Enhancing Large Language Models

Jiaxin Zhang[1,2], **Wendi Cui**[2], **Yiran Huang**[2], **Kamalika Das**[1,2], **Sricharan Kumar**[1,2]

[1]Intuit AI Research   [2]Intuit

{jiaxin_zhang, wendi_cui, yiran_huang, kamalika_das, sricharan_kumar}@intuit.com

Abstract

Large language models (LLMs) are proficient in capturing factual knowledge across various domains. However, refining their capabilities on previously seen knowledge or integrating new knowledge from external sources remains a significant challenge. In this work, we propose a novel synthetic knowledge ingestion method called Ski, which leverages fine-grained synthesis, interleaved generation, and assemble augmentation strategies to construct high-quality data representations from raw knowledge sources. We then integrate Ski and its variations with three knowledge injection techniques: Retrieval Augmented Generation (RAG), Supervised Fine-tuning (SFT), and Continual Pre-training (CPT) to inject and refine knowledge in language models. Extensive empirical experiments are conducted on various question-answering tasks spanning finance, biomedicine, and open-generation domains to demonstrate that Ski significantly outperforms baseline methods by facilitating effective knowledge injection. We believe that our work is an important step towards enhancing the factual accuracy of LLM outputs by refining knowledge representation and injection capabilities. [1]

1 Introduction

Large language models (LLMs) demonstrate proficiency in capturing vast amounts of factual information across a wide range of fields, attributable to their extensive pre-training datasets (Petroni et al., 2019; Cohen et al., 2023; Hu et al., 2023). Although these models hold an impressive repository of knowledge, integrating new information via external datasets or enhancing their capacity on previously seen information still poses several challenges. One primary challenge is *outdated*
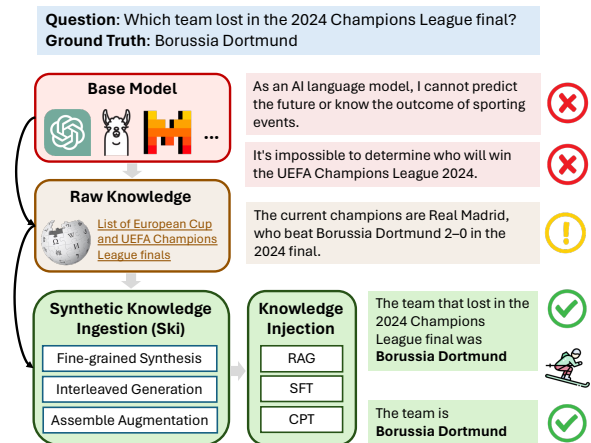

Figure 1: Illustrative example. Base models are often unable to handle certain questions due to limited knowledge. Even with raw knowledge provided, the output answer may still be incorrect if not well-digested by LLMs. Our proposed Synthetic Knowledge Ingestion method, Ski, incorporates three key innovations to easily transform raw knowledge into refined data representations that LLM can effectively digest. By utilizing injection pipelines such as RAG, SFT, and CPT, knowledge or information will be injected into LLM to ensure accurate and correct answers.

knowledge, where the static nature of the information fails to evolve over time. Another issue is the *domain knowledge deficit* where language models, typically generalists, lack detailed, specialized knowledge in sectors like finance (Wu et al., 2023) and healthcare (Singhal et al., 2023). Additionally, there is the problem of *catastrophic forgetting*, where language models may lose previously acquired knowledge (Luo et al., 2023), which particularly affects rare facts that are minimally represented in the training data (Kandpal et al., 2023). These issues underscore the necessity for ongoing enhancements to the knowledge capabilities.

In response to emerging challenges, the practice of customizing Large Language Models (LLMs) to specialized fields, coupled with regular updates

---

[1]The source code and datasets will be publicly available for research purposes.
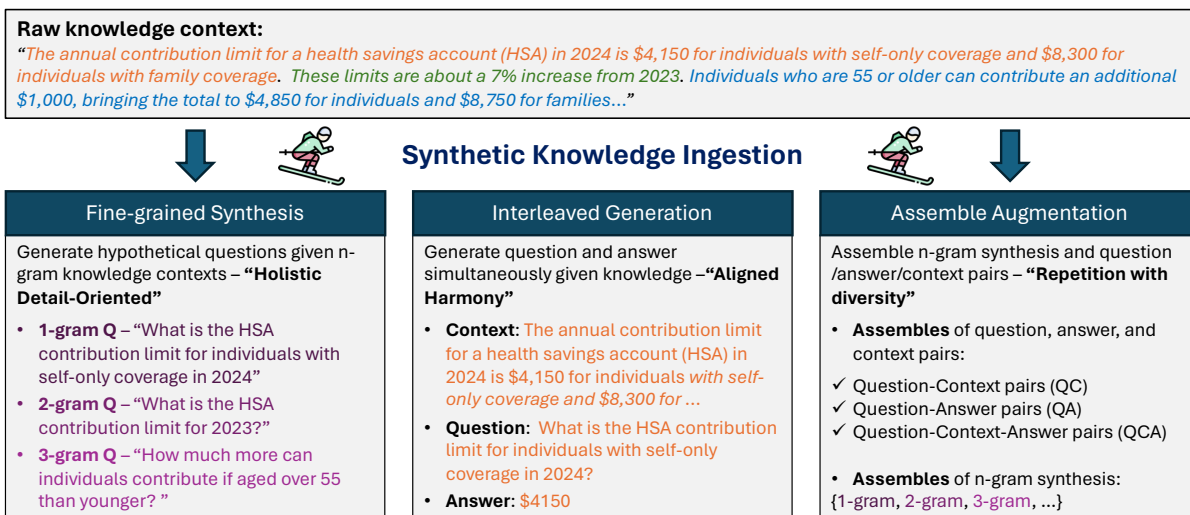
Figure 2: Overview of the proposed method: Synthetic Knowledge Ingestion (Ski), comprises of three essential components. Firstly, fine-grained synthesis incorporates generating hypothetical questions based on an n-gram detail-oriented principal. Secondly, interleaved generation generates questions and answers simultaneously by maintaining aligned harmony, given a specific knowledge piece. Lastly, assemble augmentation combines question, answer, and context pairs, along with n-gram synthesis, to improve the repetition with diversity.

to their knowledge base, is becoming increasingly prevalent (Yu et al., 2022; Wang et al., 2023). Various strategies have been developed to enhance the factual accuracy and domain-specific expertise of LLMs. These initiatives are crucial for integrating precise and current knowledge, thus broadening the utility and effectiveness of LLMs within professional settings. Despite these advances, the performance of LLMs often remains suboptimal, with issues such as hallucination (Manakul et al., 2023; Zhang et al., 2023a), especially in tasks that require extensive knowledge. Consequently, assessing the LLM's capability to understand, memorize, and retrieve knowledge is substantially essential. Given some knowledge in the form of a raw text corpus, our objective is to understand two key concepts and fundamental questions: (i) **Knowledge injection**: What is the optimal approach to encoding targeted knowledge into a language model to enhance its capability and functionality? and (ii) **Knowledge ingestion**: What are the most effective strategies for formatting knowledge or databases to ensure that such representations are readily assimilable and digestible by large language models (LLMs) prior to injection?[2]

A prevalent technique for injecting external information into model responses is Retrieval Aug-

mented Generation (**RAG**) (Lewis et al., 2020). This method relies on external knowledge sources, yet its effectiveness is contingent upon the efficacy of its retrieval system (Chen et al., 2024). If this retrieval process is flawed, it might not secure the necessary information, resulting in inappropriate responses. Additionally, a misalignment in knowledge representation could hinder RAG's comprehension of the context (Gao et al., 2023). Another approach is Supervised Fine-Tuning (**SFT**) (Dettmers et al., 2024), which improves a model by continuing its training on relevant tasks. Instruction tuning (Zhang et al., 2023c; Taori et al., 2023) under SFT has shown substantial enhancements in language model performance, though it doesn't necessarily enrich the model's knowledge base (Ouyang et al., 2022; Chia et al., 2023; Zhou et al., 2024). Techniques in fine-tuning also include reinforcement learning (RL) strategies such as RLHF (Touvron et al., 2023; Achiam et al., 2023) and DPO (Rafailov et al., 2024), which refine the model's alignment post-training but do not expand its knowledge capacity. Hence, developing a precise supervised mechanism for raw knowledge implementation poses a significant challenge. A third strategy is Continual Pre-Training (**CPT**) (Ke et al., 2023), or unsupervised fine-tuning, where the model is further trained on specific knowledge datasets tailored to certain tasks or domains (Wu et al., 2024). While CPT does not require labeled data, structuring the data in a format that closely re-

---

[2]Ingestion centers on the acquisition and processing of knowledge for storage and future use, whereas injection focuses on actively integrating knowledge to enhance a language model, either during inference or training phases.

flects the specific goals and tasks can be particularly effective. Recent studies have garnered attention for these three knowledge injection strategies (Ovadia et al., 2023; Balaguer et al., 2024; Mecklenburg et al., 2024), yet there remains a gap in knowledge ingestion. Given unprocessed and unstructured knowledge, two key questions arise: (i) *What data representations are most effective for each injection strategy?* (ii) *How can we systematically construct diverse and high-quality representations to facilitate effective knowledge injection?*

Inspired by this gap, we introduce Synthetic knowledge ingestion (Ski), an innovative synthetic data generation method that automates and enhances knowledge ingestion, as illustrated in Fig.1. Ski leverages three key innovations to generate high-quality and diverse data representation from a raw knowledge base. First, *Fine-grained Synthesis* creates hypothetical questions based on n-gram knowledge contexts, ensuring a detailed match in relevance, minimizing the semantic gap between questions and answers, and increasing representation diversity. Second, *Interleaved Generation* simultaneously generates both questions and answers based on specific knowledge. This synthetic question-answering (QA) format naturally mirrors the process of information-seeking, providing direct contextual alignment and relevance between the questions and their respective answers. Third, *Assemble Augmentation* combines fine-grained synthesis across different n-gram spans and their QA pair iterations, balancing repetition with diverse elements. By integrating these components, the Ski approach significantly enhances the refinement of knowledge from its raw state, thus facilitating effective knowledge ingestion into LLMs.

Ski offers a generic solution for generating synthetic data from raw knowledge, which can be integrated with all three knowledge injection strategies: RAG, SFT, and CPT. With this advancement, Ski not only incorporates new information using external datasets but also refines LLMs' capability on previously seen knowledge and information. We provide a comprehensive evaluation of Ski on two open-source LLMs, Llama2-7B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023), over cross-domain QA tasks, including those in the finance, biomedical, open-generation and multi-hop domain. Through performing extensive experiments, our approach Ski achieves substantial improvements upon the baselines by a large margin. In summary, our contributions are threefold:

- We develop Ski, a novel synthetic knowledge ingestion method that integrates three essential innovations, advancing the capability of refining knowledge representation from raw contexts.

- We integrate Ski with three knowledge injection strategies—RAG, SFT, and CPT—to effectively inject and refine knowledge in LLMs.

- We conduct experiments on various question-answering tasks across different domains. Our results demonstrate that Ski significantly outperforms baseline methods by a large margin.

2 Problem Formulation

Knowledge Ingestion. It generally refers to the methodology of acquiring, integrating, and transforming information from a variety of knowledge sources. It encompasses gathering, absorbing, and morphing knowledge to build a database that facilitates future use. Mathematically, let's $\mathcal{M}$ represent a language model, $\mathcal{Q}$ denote a set of factual questions, and assume that we only have access to relevant raw knowledge base $\mathcal{K}_{\mathcal{Q}}$. We aim to enhance the representation of the knowledge base, represented as $\mathcal{K}_{\mathcal{Q}}^*$

$$\mathcal{K}_{\mathcal{Q}}^* = \mathcal{T}(\mathcal{K}_{\mathcal{Q}}), \quad \mathcal{T}(\cdot) : \text{text} \rightarrow \text{text} \quad (1)$$

where $\mathcal{T}$ signifies a transformation process that converts raw knowledge into refined knowledge $\mathcal{K}_{\mathcal{Q}} \rightarrow \mathcal{K}_{\mathcal{Q}}^*$. Although advanced knowledge representations such as knowledge graphs show promise, their discussion falls beyond the scope of this study.

Knowledge Injection. It involves actively encoding or integrating specific knowledge into a pre-trained language model to enhance its performance by incorporating new information from external datasets or refining the model's capabilities on previously seen information. Specifically, when the refined knowledge applied improves the understanding of the pre-trained model $\mathcal{M}$ regarding questions $\mathcal{Q}$ through a transformation $\mathcal{F}$,

$$\mathcal{M}^* := \mathcal{F}(\mathcal{M}, \mathcal{K}_{\mathcal{Q}}^*) \quad \text{s.t.} \quad \mathcal{S}_{\mathcal{M}^*, \mathcal{Q}} > \mathcal{S}_{\mathcal{M}, \mathcal{Q}} \quad (2)$$

where $\mathcal{S}$ is a score metric, e.g., an accuracy that measures the performance of the pre-trained model in answering the questions $\mathcal{Q}$, and $\mathcal{M}$ is the updated language model. We explore three options for $\mathcal{F}$: RAG, SFT, and CPT, and provide a systematic evaluation of how our proposed synthetic knowledge ingestion method, Ski, can enhance the capabilities of LLMs across various QA tasks.

3 Synthetic Knowledge Ingestion

The human learning process often involves asking questions and inquiring "why", which can enhance comprehension. Drawing inspiration from this, Ski also leverages the power of questions. By utilizing three key innovations, Ski transforms raw knowledge (such as documents and articles) into question-augmented representations. The following section provides a detailed description of these three key innovations, as illustrated in Figure 2.

3.1 Fine-grained Synthesis

Considering the scenario where we possess a paragraph of text as a knowledge base, crafting effective questions to transform and augment this knowledge base poses a challenging task. Questions that are too broad may struggle to encompass all pertinent knowledge points, whereas excessively detailed questions risk losing sight of the overall content. Moreover, ensuring that these questions are both non-repetitive and diverse adds extra challenges.

To tackle these challenges, we introduce a fine-grained synthesis approach, inspired by n-gram language models, allowing for a balanced incorporation of both detailed and overarching content. Assume the knowledge base $\mathcal{K}_{\mathcal{Q}}$ paragraph consists of m sentences, $\mathcal{K}_{\mathcal{Q}} = \{k_1, k_2, ..., k_m\}$ where k_i means the i-th sentence in $\mathcal{K}_{\mathcal{Q}}$, we generate hypothetical questions by querying an LLM model, given a specific set of sentences conditioning on the whole knowledge paragraph $\mathcal{K}_{\mathcal{Q}}$:

$$q_j^n \leftarrow \mathcal{L}_{P_{fs}}(\{k_j, k_{j+1}, ..., k_{j+n-1}\}; \mathcal{K}_{\mathcal{Q}}), \quad (3)$$

where $\mathcal{L}$ is a pioneer LLM model with meta-prompt P_{fs}, q_j^n is the j-th ($1 \leq j \leq m - n + 1$) generated hypothetical questions given the specific sentence set $\{k_j, k_{j+1}, ..., k_{j+n-1}\}$ and the raw knowledge context $\mathcal{K}_{\mathcal{Q}}$. We denote $\tilde{Q}^n$ as the set of n-gram hypothetical questions: $\tilde{Q}^n = \{q_1^n, q_2^n, q_3^n, ..., q_{m-n}^n\}$. We define a set of n-gram knowledge context C^n

$$C^n = \{\{k_1, ..., k_n\}, ..., \{k_{m-n+1}, k_m\}\} \quad (4)$$

and therefore have $C^1 = \{k_1, k_2, ..., k_m\}$, $C^2 = \{\{k_1, k_2\}, \{k_2, k_3\}, ..., \{k_{m-1}, k_m\}\}$, etc. The question-context pair $\{\tilde{Q}^n, C^n\}$ can be written by:

$$\{\tilde{Q}^n, C^n\} = \{\{q_1^n, \{k_1, ..., k_n\}\}, ..., \\ \{q_{m-n}^n, \{k_{m-n+1}, k_m\}\}\}. \quad (5)$$

Specifically, we have two variants of Ski leveraging n-gram synthesis principle:

- Ski-Q-n: synthetic hypothetical questions $\tilde{Q}^n$ by n-gram given knowledge contexts C^n

- Ski-QC-n: synthetic hypothetical questions with knowledge context pair $\{\tilde{Q}^n, C^n\}$ by n-gram

where n typically ranges from 1 to 3. Specifically, Ski-Q-1 refers synthetic 1-gram questions over m sentences, $\tilde{Q}^1 = \{q_1^1, q_2^1, ..., q_m^1\}$, and Ski-QC-1 means synthetic 1-gram question-context (QC) pair $\{\tilde{Q}^1, C^1\} = \{\{q_1^1, k_1\}, ..., \{q_m^1, k_m\}\}$ that includes m pairs in this set. More details and examples can be found in the Appendix A.2.

3.2 Interleaved Generation

Hypothetical question and context pairs have the potential to transform knowledge articles into questions (Ski-Q-n) and question-context (QC) pairs (Ski-QC-n), which can revolutionize representations of knowledge articles. However, SFT requires succinct QA pairs, while the context portion of the QC pair consists of sentences, which are too lengthy to be formulated as answers.

To deal with this issue, we introduce an interleaved generation strategy that simultaneously generates QA pairs based on a specific knowledge context. This synthetic QA format naturally emulates the information-seeking process, providing direct contextual alignment and relevance between the questions and their corresponding answers. This strategy can be built upon hypothetical questions yet delivers QA pairs tailored to the specific knowledge context $\mathcal{K}_{\mathcal{Q}}$:

$$\{q_j^n, a_j^n\} \leftarrow \mathcal{L}_{P_{ig}}(\{k_j, ..., k_{j+n-1}\}; \mathcal{K}_{\mathcal{Q}}), \quad (6)$$

where P_{ig} is the meta-prompt (see details in Appendix) different from the prompt P_{fs} used above. We denote $\tilde{A}^n$ as the set of answers corresponding to the n-gram hypothetical questions: $\tilde{A}^n = \{a_1^n, a_2^n, a_3^n, ..., a_{m-n}^n\}$. The question-answering pairs $\{\tilde{Q}^n, A^n\}$ can be written by:

$$\{\tilde{Q}^n, \tilde{A}^n\} = \{\{q_1^n, a_1^n\}, ..., \{q_{m-n}^n, a_{m-n}^n\}\}. \quad (7)$$

The synthetic questions $\tilde{Q}^n$ and corresponding answers $\tilde{A}^n$ align well with the given knowledge context C^n due to interleaved mechanism. We thus denote the following Ski variants from the perspective of the QA pair:

- Ski-QA-n: synthetic QA pair $\{\tilde{Q}^n, \tilde{A}^n\}$ by n-gram given specific knowledge contexts C^n

- Ski-QCA-n: synthetic QA pair combined with the knowledge context, $\{\tilde{Q}^n, C^n, \tilde{A}^n\}$

3.3 Assemble Augmentation

To enhance the effectiveness of knowledge ingestion, we propose a strategic assembly approach that harnesses the benefits of fine-grained and interleaved generation. This strategy is built upon two main pillars: (1) article/document augmentation for optimized retrieval, and (2) data augmentation for both supervised and unsupervised fine-tuning. Specifically, we first combine all question-context pairs from the same document into a singular article, thereby improving the retrieval quality

$$[\tilde{Q}C]^n = \{q_1^n, \{k_1, ..., k_n\}\} \oplus \{q_2^n, \{k_2, ..., k_{n+1}\}\} ... \oplus \{q_{m-n}^n, \{k_{m-n+1}, k_m\}\} \tag{8}$$

where $\oplus$ denotes the concatenation operator, which combines all QC pair strings into a single article. The notation $[\tilde{Q}C]^n$ represents the augmented article composed of all QC pairs using n-gram techniques. As the same information can exist in different formats under different n-gram strategies, combining pairs from different n-gram allows repetition and diversity, enhancing the depth and breadth of knowledge integration.

Assemble augmentation also offers scalability suitable for synthesizing expansive sets of question-answer pairs utilizing various n-gram generation techniques. For fine-tuning schemes, this strategy can be employed to generate a diverse ensemble of n-gram QA pairs for data augmentation purposes. Specifically, the QA assembly aggregates all n-gram QA pairs as follows:

$$[\tilde{Q}\tilde{A}]^n = \{\tilde{Q}^1, \tilde{A}^1\} \cup ... \cup \{\tilde{Q}^n, \tilde{A}^n\} \tag{9}$$

where $[\tilde{Q}\tilde{A}]^n$ is a combination of QA pairs from 1-gram to n-gram. Similarly, we have the ensemble of QCA pairs from 1-gram to n-gram:

$$[\tilde{Q}C\tilde{A}]^n = \{\tilde{Q}^1, C^1, \tilde{A}^1\} \cup ... \cup \{\tilde{Q}^n, C^n, \tilde{A}^n\}, \tag{10}$$

where $\cup$ denotes the union of sets, which refers to containing all the elements in a large set. We thus summarize three `Ski` variants from the perspective of assemble that typically performs article augmentation (for retrieval) or data augmentation:

- `Ski-QC-ASM`: assemble of all question-context pairs into one augmented set $[\tilde{Q}C]^n$

- `Ski-QA-ASM`: assemble of all question-answer pairs from 1-gram to n-gram $[\tilde{Q}\tilde{A}]^n$

- `Ski-QCA-ASM`: assemble of all question-context-answer pairs from 1-gram to n-gram $[\tilde{Q}C\tilde{A}]^n$

4 Knowledge Injection for Enhancing Language Models

4.1 Retrieval Augmented Generation (RAG)

We employ three variations of the `Ski` approach, namely `Ski-Q-n`, `Ski-QC-n`, and `Ski-QC-ASM`, when utilizing the RAG pipeline for knowledge injection. At the retrieval stage, `Ski-Q-n` matches the query's embedding with embeddings of all questions generated in an n-gram pattern from the knowledge documents. This can be interpreted as an inverse HyDE implementation, where HyDE (Gao et al., 2022a) transfers a query to a potential answer and performs an answer-to-knowledge search. For `Ski-Q-n` we transform knowledge into questions and execute a question-to-question (Q-Q) search. In contrast, `Ski-QC-n` compares the query's embedding with embeddings of all question-context (QC) pairs, facilitating a question-to-(question+context) (Q-QC) search. Additionally, `Ski-QC-n` introduces another variation, `Ski-QC-ASM` , where QC pairs from the same document are amalgamated into a single article, and the query's embedding is compared with embeddings of such articles. After identifying the most relevant items, `Ski` consolidates only the most pertinent snippets or sentences from the knowledge documents related to the retrieved items into an LLM for generating the final answer. This process minimizes noise from sentences in the same documents but not directly pertinent to user queries.

4.2 Supervised Fine-tuning (SFT)

We utilize three variants of the `Ski` method, namely question-answer (QA) pairs, question-context (QC) pairs, and question-context-answer (QCA) pairs, to incorporate knowledge via supervised fine-tuning. In the QCA approach, we concatenate the question (Q) and context (C) input and use the answer as the output, following the setting in Viswanathan et al. (2023). To conform to the n-gram principle, we create separate training datasets for every n-gram (n=1,2,3) for SFT, resulting in a decrease in dataset size as n increases. We focus on evaluating three fine-grained generations (n=1), i.e., `Ski-QA-1`, `Ski-QC-1`, and `Ski-QCA-1`.

4.3 Continual Pre-training (CPT)

Continual pre-training can leverage the dataset used for SFT, but it typically requires a larger volume of data to enhance the capabilities of LLMs (Lin et al., 2024; Ke et al., 2023). To address

Model	Method	BioASQ				NQ				FiQA			
		nDCG@1	nDCG@10	Recall@1	Recall@10	nDCG@1	nDCG@10	Recall@1	Recall@10	nDCG@1	nDCG@10	Recall@1	Recall@10
Contriever	Raw Article	0.810	0.720	0.133	0.578	0.748	0.831	0.644	0.915	0.298	0.323	0.134	0.406
	Ski-Q-1 (iHyDE)	**0.845**	0.733	**0.142**	0.574	0.752	0.804	0.641	0.876	0.452	0.482	0.231	0.562
	Ski-QC-1	0.835	**0.751**	0.132	**0.597**	0.810	0.865	0.704	0.926	0.455	0.493	0.232	0.584
	Ski-QC-ASM	0.820	0.719	0.131	0.575	0.773	0.839	0.670	0.908	0.398	0.435	0.193	0.542
BM25	Raw Article	0.232	0.297	0.189	0.389	0.325	0.389	0.272	0.478	0.315	0.318	0.162	0.368
	Ski-Q-1 (iHyDE)	0.175	0.219	0.142	0.289	0.218	0.282	0.176	0.368	0.238	0.252	0.121	0.307
	Ski-QC-1	0.210	0.282	0.173	0.373	0.3	0.377	0.245	0.482	0.289	0.301	0.142	0.362
	Ski-QC-ASM	**0.268**	**0.328**	**0.221**	**0.417**	**0.350**	**0.422**	**0.291**	**0.513**	**0.377**	**0.390**	**0.192**	**0.454**

Table 1: Retrieval performance on three datasets using two different retrievers (Contriever and BM25).
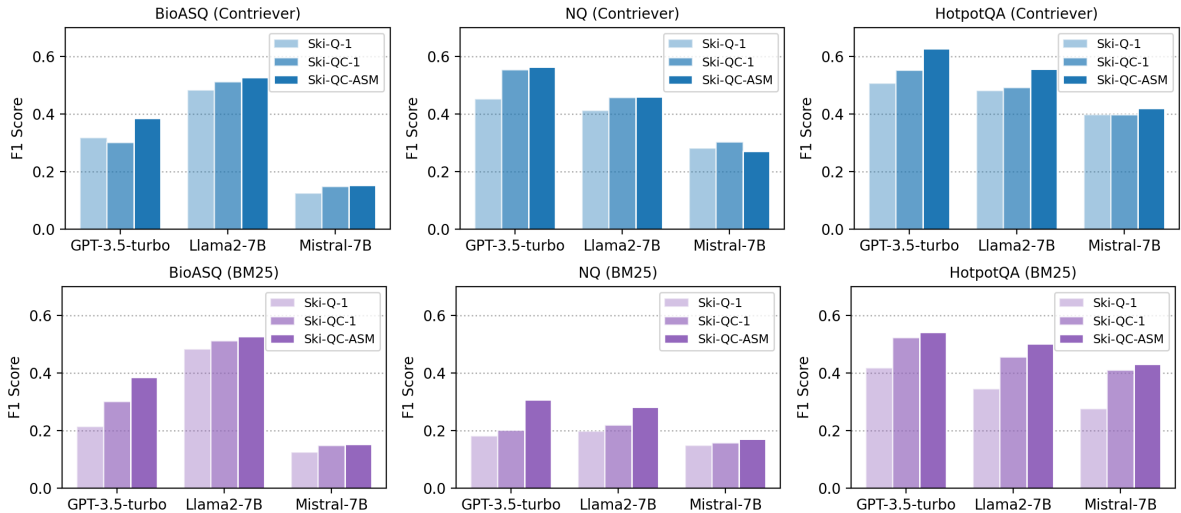

Figure 3: Effect of `Ski` method on the RAG performance using different retrievers and generators.

this, we amalgamate all the generated pairs from SFT to create a comprehensive augmented training set, `Ski-QC-ASM`, `Ski-QA-ASM`, `Ski-QCA-ASM`. This strategy not only amplifies repetition but also preserves diversity (Ovadia et al., 2023).

5 Experiments

5.1 Experiment Setup

Datasets. We evaluate our approach on multiple cross-domain question-answering tasks, including those in the finance domain (e.g., FiQA (Maia et al., 2018)), biomedical domain (e.g., BioASQ (Tsatsaronis et al., 2015)), open-generation domain (e.g., NQ (Kwiatkowski et al., 2019) and multi-hop domain (e.g., HotpotQA (Yang et al., 2018)). Please find more relevant details about data usage for RAG, SFT, and CPT in the Appendix A.3-A.4.

Evaluation Models and Metrics. We utilize `GPT-3.5-turbo` from OpenAI as our target agent model to generate synthetic questions, answers, and their corresponding pairs. To evaluate our approach, we employ two open-source LLMs, `Llama2-7B` (Touvron et al., 2023) and `Mistral-7B` (Jiang et al., 2023), which serve as the base models for RAG, SFT and CPT, implemented using

the LLaMA-Factory pipeline[3]. Our evaluation encompasses a comparison between two retrieval methods: the Dense-Retrieval approach, Contriever (Izacard et al., 2021), and the Sparse-Retrieval method, BM25. We assess the performance of three generators, `GPT-3.5-turbo`, `Llama2-7B`, and `Mistral-7B` using the F1 score as a measure of accuracy, following the RAGGED (Hsia et al., 2024) pipeline[4] across RAG, SFT, and CPT scenarios.

Baseline Methods. For retrieval and RAG tasks, we utilize raw knowledge — specifically, the original article as a standard baseline. Our approach draws inspiration from Hypothetical Document Embeddings, known as HyDE (Gao et al., 2022b). We introduce an adaptation termed "inverse" HyDE (iHyDE) (Gao et al., 2023) established by `Ski-Q-1`. In SFT evaluations, the base model without fine-tuning serves as the standard baseline. We also employ a vanilla QA baseline, which generates a set of question-answer pairs by querying LLMs using extracted documents/articles, as outlined in (Mecklenburg et al., 2024; Balaguer et al., 2024). For the CPT pipeline, our standard baselines encompass the base model and raw context, which is

[3] https://github.com/hiyouga/LLaMA-Factory
[4] https://github.com/neulab/ragged

aligned with the training dataset setup as detailed in (Ovadia et al., 2023). Additionally, we integrate two novel baseline methods: `Ski-C-1`, concentrating uniquely on the 1-gram knowledge context, and `Ski-C-ASM`, an ensemble approach that integrates knowledge contexts ranging from 1-gram to 3-gram. Both are variants of the `Ski` approach.

5.2 Main Results

RAG Results. We evaluate the retrieval performance of `Ski` using two different retrievers, as illustrated in Table 1. For Contriever, `Ski-QC-1` outperformed both the inverse HyDE (`Ski-Q-1`) and the raw article baselines, particularly achieving a significant improvement (+15%) on the FiQA task. For BM25, `Ski-QC-ASM` surpassed all the baselines and `Ski` variations across three datasets. Further evaluations were conducted on our method within RAG systems, detailed in Table 2. Inverse HyDE fell short of the raw article baseline, indicating that the question-to-question search approach might not be as effective as anticipated. `Ski-QC-ASM` emerged as the superior method, showing consistent improvements compared to other baselines across all tasks. The results of BM25 are aligned with the retrieval-only task, confirming the effectiveness of assemble augmentation.

Model	Method	BioASQ	NQ	HotpotQA
Contriever +GPT-3.5	Raw Article	0.361	0.504	0.585
	Ski-Q-1 (iHyDE)	0.317	0.452	0.507
	Ski-QC-1	0.300	0.553	0.553
	Ski-QC-ASM	**0.385**	**0.563**	**0.627**
BM25 +GPT-3.5	Raw Article	0.227	0.258	0.443
	Ski-Q-1 (iHyDE)	0.162	0.181	0.274
	Ski-QC-1	0.186	0.201	0.377
	Ski-QC-ASM	**0.243**	**0.306**	**0.478**

Table 2: End-to-end RAG results with two retrievers.

Effect of Different Retrievers and Generators. Fig.3 presents a detailed evaluation of how different retrievers and generators influence our `Ski` methods. In the NQ and HotpotQA tasks, `Ski` methods that incorporate Contriever generally outperform those using BM25. In terms of generation, `GPT-3.5-turbo` demonstrates superior performance over `Llama2-7B` and `Mistral-7B`. For the BioASQ task, both Contriever and BM25 perform comparably, with `Llama2-7B` surpassing the other two generators in effectiveness. We compare three variations of `Ski`, namely `Ski-Q-1`, `Ski-QC-1`, and `Ski-QC-ASM`, across different combinations of retrievers and generators. Consistently across tasks,

`Ski-QC-ASM` tends to perform the best, followed by `Ski-QC-1`, outperforming the baseline `Ski-Q-1`.

SFT Results. Beyond the RAG evaluation, we assess the enhancement of pre-trained models using SFT in our study with `Ski`. The synthetic QA pairs, `Ski-QA-1`, exhibit superior performance across three datasets. Specifically, `Mistral-7B` achieves an average performance gain of 5.13% over the base model and 4.07% improvement compared to the vanilla QA baseline, while `Llama2-7B` markedly improves by 19.3% and 11.1% respectively. Generally, the data representation provided by `Ski-QA-1` is the more effective method. We also observe that `Llama2-7B` demonstrates a stronger capability in refining knowledge compared to `Mistral-7B`, aligning with findings reported by Ovadia et al. (2023).

Model	Method	BioASQ	NQ	HotpotQA
Mistral-7B	Base Model	0.125	0.098	0.113
	Vanilla QA	0.131	0.084	0.155
	Ski-QA-1	**0.162**	**0.159**	**0.171**
	Ski-QC-1	0.138	0.113	0.155
	Ski-QCA-1	0.160	0.157	0.163
Llama2-7B	Base Model	0.123	0.082	0.136
	Vanilla QA	0.235	0.150	0.203
	Ski-QA-1	0.357	**0.218**	**0.266**
	Ski-QC-1	**0.436**	0.217	0.258
	Ski-QCA-1	0.271	0.201	0.252

Table 3: SFT performance on two pre-trained models.

Effect of n-gram Synthesis. The impact of n-gram on SFT is illustrated in Fig. 4. Across various tasks, the 1-gram consistently outperforms the 2-gram and 3-gram in all `Ski` variations, underscoring the significance of fine-grained generation for enhancing both the quantity and quality of data synthesis. Compared to QA pairs and QCA pairs, the performance of QC pairs drops slightly and is inferior to the other two variations. This suggests that the interleaved generation in QA pairs is crucial for improving SFT performance.

CPT Results. `Llama2-7B` demonstrates a superior capability for knowledge injection compared to `Mistral-7B` in the SFT evaluation. Building on this observation, we further explore the impact of `Ski` through the CPT approach with `Llama2-7B`. While QA pairs excel in SFT, `Ski-QC-ASM` and `Ski-QCA-ASM` show significantly enhanced performance in the CPT evaluation. Specifically, these methods lead to performance gains of +21.2% in
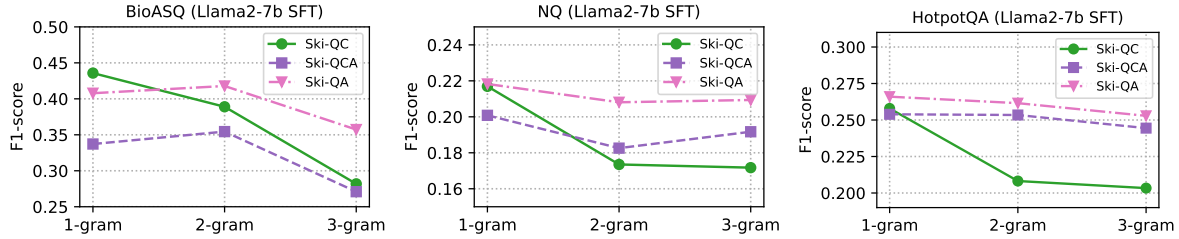
Figure 4: SFT performance on different n-gram settings using `Llama2-7B` base model across three datasets. Note that the F1-score often decreases with the increase of n-gram, but not very significantly drop.

BioASQ and +11.5% in NQ task relative to the base model. `Ski-C-ASM` presents a slight improvement over other QC and QCA variations in HotpotQA. This highlights the benefit of assembly augmentation where longer contexts, which include more comprehensive information than short answers, aid in deepening the pretrained models' understanding.

Method	BioASQ	NQ	HotpotQA
Base Model	0.123	0.082	0.136
Raw Article (only context)	0.219	0.166	0.242
`Ski-C-1` (1-gram context)	0.208	0.178	0.204
`Ski-C-ASM` (1-3-gram context)	0.241	0.163	**0.253**
`Ski-QA-1`	0.269	0.184	0.205
`Ski-QA-ASM`	0.228	0.194	0.218
`Ski-QC-1`	0.294	0.178	0.226
`Ski-QC-ASM`	**0.335**	0.182	0.215
`Ski-QCA-1`	0.211	0.188	0.218
`Ski-QCA-ASM`	0.220	**0.197**	0.191

Table 4: CPT performance on `Llama2-7B`.

Holistic Comparison of RAG, SFT, and CPT. Fig.5 presents a comprehensive comparison of the `Ski` method across various knowledge injection pipelines. While RAG demonstrates marked improvement over the raw model and baselines, the gains from CPT are relatively modest. Despite this, `Llama2-7B` consistently shows significant improvements over `Mistral-7B` across RAG, SFT, and CPT. `GPT-3.5-turbo` is utilized solely as the generator for RAG, where it exhibits relatively low-performance gains due to its already high base performance. Our `Ski` method appears promising in enhancing the RAG performance of open-source pre-trained models, suggesting that `Llama2-7B` could be an effective candidate for knowledge injection, potentially achieving competitive gains similar to RAG when applied through SFT and CPT.

6 Related Work

Incorporating new information into LLMs or refining their capabilities with previously seen information has posed a substantial challenge (Chen et al.,
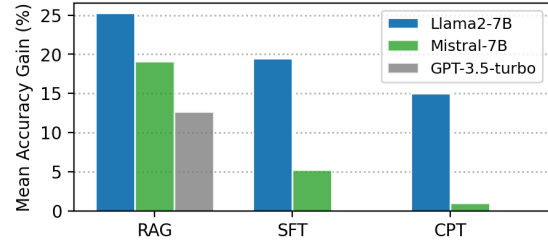

Figure 5: The relative performance (F1-score) gain for RAG, SFT, and CPT over averaged across all datasets.

2022; Martino et al., 2023; Zhang et al., 2023d; Ye et al., 2023; Zhang et al., 2023b), the lack of effective approaches and evaluations is a notable concern. Recently, Xu et al. (2023) explored continued pre-training with a knowledge infill objective to enhance fact retention, while Tian et al. (2023) applied constrained optimization through direct preference optimization (Rafailov et al., 2024) to reinforce factuality during post-training. Additionally, Ovadia et al. (2023) aimed to compare retrieval, fine-tuning, and their combinations but the optimal way to inject knowledge into LLMs remains an open and sometimes contentious debate.

However, a critical knowledge ingestion gap remains: what are the data representations that work best for each injection pipeline with unprocessed and unstructured knowledge? (Baek et al., 2024) To answer this, Balaguer et al. (2024) extracts information from raw PDF documents, leading to questions and answers for supervised fine-tuning pipelines. Mecklenburg et al. (2024) presents two data generation processes, token-based and fact-based, to inject knowledge via SFT. In contrast, our study aims to develop an innovative and generic synthetic knowledge ingestion approach, `Ski`, which achieves significant improvement for each knowledge injection pipeline.

7 Conclusions

We propose a novel method for refining knowledge representation and injection to enhance LLMs.

Through extensive empirical analysis, our work has revealed several important findings. *Firstly*, synthetic QA generations are highly effective in digesting knowledge from its raw format, resulting in consistent enhancement for each injection pipeline. *Secondly*, fine-grained assemble augmentation has proven to be proficient in constructing high-quality, diverse datasets for knowledge injection. *Lastly*, RAG shows the potential to enhance knowledge, whereas the advancement of SFT and CPT largely depends on selecting appropriate base models and leveraging their existing knowledge capabilities.

8 Ethics Statement

This paper studies knowledge ingestion and representation in LLMs, which has significant broader impacts on the field of natural language processing and helps to address ethical considerations regarding factuality and reliability. The research outcome may contribute to the development of more accurate and factual LLMs by mitigating the risks of misinformation and biased outputs and promoting accountability and trust in AI systems.

9 Limitations

Our current experiments focus on tasks in a question-answering setting which requires a knowledge base. Further research is needed to expand to more complex tasks such as conversational or dialogue-based prompting. Also, the quality of the results introduced by Ski depends on the quality of the knowledge base. For OOD questions or edge cases where the knowledge base is corrupted or provides conflicting information, Ski is not able to solve that. Ski does require LLM calls to generate different representations. Such jobs can be conducted offline and only once. Thus the cost will be relatively reduced as it serves more queries. In addition, although advanced knowledge representations such as knowledge graphs show promise, their discussion falls beyond the scope of the current study but will appear in our future work.

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*.

Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar.

2024. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024*, pages 3355–3366.

Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O Nunes, et al. 2024. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv e-prints*, pages arXiv–2401.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022*, pages 2778–2788.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint arXiv:2306.04757*.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. 2023. Crawling the internal knowledge-base of language models. *arXiv preprint arXiv:2301.12810*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022a. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022b. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Jennifer Hsia, Afreen Shaikh, Zhiruo Wang, and Graham Neubig. 2024. Ragged: Towards informed design of retrieval augmented generation systems. *arXiv preprint arXiv:2403.09040*.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

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*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. 2024. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, et al. 2024. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

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*.

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.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28.

Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Wu, and Graham Neubig. 2023.

Prompt2model: Generating deployable models from natural language instructions. *arXiv preprint arXiv:2308.12261*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Yan Xu, Mahdi Namazifar, Devamanyu Hazarika, Aishwarya Padmakumar, Yang Liu, and Dilek Hakkani-Tür. 2023. Kilm: Knowledge injection into encoder-decoder language models. *arXiv preprint arXiv:2302.09170*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Qichen Ye, Junling Liu, Dading Chong, Peilin Zhou, Yining Hua, and Andrew Liu. 2023. Qilin-med: Multi-stage knowledge injection advanced medical large language model. *arXiv preprint arXiv:2310.09089*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023a. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

Qinggang Zhang, Junnan Dong, Hao Chen, Xiao Huang, Daochen Zha, and Zailiang Yu. 2023b. Knowgpt: Black-box knowledge injection for large language models. *arXiv preprint arXiv:2312.06185*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023c. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Huadong Wang, Deming Ye, Chaojun Xiao, Xu Han, Zhiyuan Liu, Peng Li, Maosong Sun, et al. 2023d. Plug-and-play knowledge injection for pre-trained language models. *arXiv preprint arXiv:2305.17691*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Synthetic Knowledge Pipeline Prompts

Table 5: Prompt to generate Questions for `Ski-QC-n`

You are a question generator. Given a series of paragraphs, you have job is to generate questions for each paragraph.

The question generated should be able to be answered ONLY based on the information in the paragraph.
The question generated should be about the main topic of the paragraph.

##Paragraphs:
{*paragraphs*}

Return the questions in a list.
["1. question 1", "2. question 2", "3. question 3" ...]

##Questions:

Table 6: Prompt to generate Questions And Answers for `Ski-QCA-n`

You are a question generator. Given a series of paragraphs, you have job is to generate questions for each paragraph and abstract the corresponding answers.

The question generated should be able to be answered ONLY based on the information in the paragraph.
The question generated should be about the main topic of the paragraph.
The answer should be about the generated question and based on the information in the paragraph.

##Paragraphs:
{*paragraphs*}

Return the questions in a list.
[{"q": "question 1", "a": "answer 1"}, {"q": "question 2", "a": "answer 2"}, ...]

##Questions:

A.2 Examples of Data generated under N-gram

Table 7: Synthetic Data Example

Configuration	Generated Synthetic Data
Original Article	Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž2013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric. Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe. He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas.", "answer": "Antonio Vivaldi was a Baroque composer, virtuoso violinist, teacher and cleric.
Ski-QC-1	[{"**question**": " Who was Antonio Lucio Vivaldi?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž2013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric."}, {"**question**": " Where was Vivaldi born and what is he recognized for?", "**context**": "Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe."}, {"**question**": " What types of music did Vivaldi compose and what is his most famous work?", "**context**": "He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas."}]
Ski-QC-2	[{"**question**": " Who was Antonio Lucio Vivaldi?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž2013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric. Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe."}, {"**question**": " What types of music did Vivaldi compose?", "**context**": "He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas. His best-known work is a series of violin concertos known as Ṫhe Four Seasons."}]

Continuation of Table 7

Ski-QC-3	[{"**question**": " Who was Antonio Lucio Vivaldi?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric. Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe. He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas."}, {"**question**": " What type of music did Vivaldi compose?", "**context**": "His best-known work is a series of violin concertos known as Ṫhe Four Seasons."}]
Ski-QCA-1	[{"**question**": "What was Antonio Lucio Vivaldi's profession?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric.", "**answer**": "Antonio Lucio Vivaldi was a composer, virtuoso violinist, teacher and cleric."}, {"**question**": "Where was Antonio Lucio Vivaldi born?", "**context**": "Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe.", "**answer**": "Antonio Lucio Vivaldi was born in Venice."}, {"**question**": "What is Antonio Lucio Vivaldi recognized as?", "**context**": "He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas.", "**answer**": "Antonio Lucio Vivaldi is recognized as one of the greatest Baroque composers."}, {"**question**": "What types of works did Antonio Lucio Vivaldi compose?", "**context**": "His best-known work is a series of violin concertos known as Ṫhe Four Seasons.", "**answer**": "Antonio Lucio Vivaldi composed many instrumental concertos, sacred choral works, and more than forty operas."}]

Ski-QCA-2	[{"**question**": "What was Antonio Lucio Vivaldi's profession?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric. Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe.", "**answer**": "Antonio Lucio Vivaldi was a composer, virtuoso violinist, teacher and cleric."}, {"**question**": "Where was Antonio Lucio Vivaldi born?", "**context**": "He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas. His best-known work is a series of violin concertos known as T̈he Four Seasons.̈", "**answer**": "Antonio Lucio Vivaldi was born in Venice."}]
Ski-QCA-3	[{"**question**": "What was Antonio Vivaldi's profession?", "**context**": "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric. Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe. He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas.", "**answer**": "Antonio Vivaldi was a Baroque composer, virtuoso violinist, teacher and cleric."}, {"**question**": "Where was Antonio Vivaldi born?", "**context**": "His best-known work is a series of violin concertos known as T̈he Four Seasons.̈", "**answer**": "Antonio Vivaldi was born in Venice."}]

A.3 Details of SFT Pipeline

Training. For the SFT Pipeline, we use the QLoRA (Dettmers et al., 2024) strategy with 4bit quantization built on top of Llama-Factory (Zheng et al., 2024)[5]. For float format we use bf16. The batch size is set to 2, and the number of update steps to accumulate the gradients is set to 4. We use a cosine scheduler with a learning rate of 5e-5 and a warm-up ratio of 0.1. The training lasts for 3 rounds, with the maximum gradient norm set to 1. We use Amazon Sagemaker g5.12xlarge and g5.24xlarge instances for training, which are powered by NVIDIA A10 Tensor Core GPUs and comprise 4 GPUs with 24 GB of memory each. For training data size please refer to Table 8.

Generation. We will generate 1 most likely generation with sampling strategies. This generation will be used to evaluate the correctness. The temperature of generation is fixed at 1.0, and top_k is fixed at 50. The maximum number of new tokens in each generation is set to 40 tokens. Instruction used for generation is *Respond to questions with concise and to-the-point answers. No explanation is needed. Keep your response within 20 words*. The test data for each dataset consists of 200 question-answer or question-context pairs.

A.4 Details of CPT Pipeline

Training. For CPT, we use the QLoRA 4bit quantization with the LoRA+ strategy with a lambda value of 16.0 and float16 mixed precision training built on top of Llama-Factory (Zheng et al., 2024). The batch size is set to 2, and the number of update steps to accumulate the gradients is set to 4. We use a cosine

---

[5] https://github.com/hiyouga/LLaMA-Factory

scheduler with a learning rate of 5e-5 and a warmup ratio of 0.1. The training lasts for 3 rounds, with the maximum gradient norm set to 1. We use Amazon Sagemaker g5.12xlarge and g5.24xlarge instances for training, which are powered by NVIDIA A10 Tensor Core GPUs and comprise 4 GPUs with 24 GB of memory each.

A.5 Details of Dataset

For each dataset, we select the first 200 queries in the test set as the final test set. Considering the huge amount of documents for retrieval, we select the articles that have been referred to at least one test query as the ground truth of the retrieval task to be the knowledge base, see Table 8.

Table 8: SFT/CPT Training Dataset Size.

Dataset	Ski-QA-1	Ski-QC-1	Ski-QA-ASM	Ski-QC-ASM
BioASQ (Tsatsaronis et al., 2015)	5000	5000	34405	34966
NQ (Kwiatkowski et al., 2019)	5000	5000	13707	13528
HotpotQA (Yang et al., 2018)	5000	5000	12183	9191

Table 9: CPT Training Data Example

Configuration	Generated Synthetic Data
Ski-QA-n	[{"**text**": "Question: What was Antonio Lucio Vivaldi's profession? Answer: Antonio Lucio Vivaldi was a composer, virtuoso violinist, teacher and cleric."}, {"**text**": "Question: Where was Antonio Lucio Vivaldi born? Answer: Antonio Lucio Vivaldi was born in Venice."}, {"**text**": "Question: What is Antonio Lucio Vivaldi recognized as? Answer: Antonio Lucio Vivaldi is recognized as one of the greatest Baroque composers."}]

Continuation of Table 9

Ski-QC-n	[{"**text**": "Question: Who was Antonio Lucio Vivaldi?" Context: "Antonio Lucio Vivaldi (] ; 4 March 1678Ŏ0a0Ž013 28 July 1741) was an Italian Baroque composer, virtuoso violinist, teacher and cleric."}, {"**text**": "Question: Where was Vivaldi born and what is he recognized for? Context: Born in Venice, he is recognized as one of the greatest Baroque composers, and his influence during his lifetime was widespread across Europe."}, {"**text**": "Question: What types of music did Vivaldi compose and what is his most famous work? Context: He composed many instrumental concertos, for the violin and a variety of other instruments, as well as sacred choral works and more than forty operas."}]

Generation. We will generate 1 most likely generation with sampling strategies. This generation will be used to evaluate the correctness. The temperature of generation is fixed at 1.0, and top_k is fixed at 50. The max number of new tokens of each generation is set to 40 tokens. Instruction used for generation is *Respond to questions with concise and to-the-point answers. No explanation needed. Keep your response within 20 words.* The test data for each dataset consist of 200 question-answer or question-context pairs.

Table 10: CPT Test Data Example

[{"**query**": "The 1978 NBA World Championship Series had as MVP which Hall of Fame class member of 1988?",
"**answer**": "Westley Sissel Unseld"},

{"**query**": "Who was born first out of Leopold Lummerstorfer and Laurent Touil-Tartour?",
"**answer**": "Leopold Lummerstorfer"},

{"**query**": "In what city did Charlie Spiller play college football?",
"**answer**": "Lorman"}]

A.6 Additional Experiments

A.6.1 Effect of N-gram on Retrieval

To gauge the impact of N-Gram on retrieval strategy, we compared the retrieval performance on FiQA task using the `Ski-Q-n` strategy. Results can be observed in Table 11. 1-Gram strategy, despite the fact that it generates the most amount of questions to compare with, leads to the best retrieval result.

A.6.2 Effect of Query Augmentation and Document Augmentation

To better understand the effectiveness of Query Augmentation and Document Augmentation on retrieval, we compare `Ski` with HyDE (Gao et al., 2022b) using the FiQA dataset and Contriver. Full results can be seen in Table 12. For the `Ski` variations, both `Ski-Q-n` and `Ski-QC-n` outperform HyDE (Gao et al., 2022b).

Table 11: Effective of N-gram for Retrieval

	nCDG@1	nCDG@5	nCDG@10	Recall@1	Recall@5	Recall@10
Article	0.316	0.311	0.343	0.145	**0.476**	**0.562**
3-Gram	0.451	0.428	0.456	0.225	0.328	0.429
2-Gram	0.448	0.435	0.468	0.221	0.454	0.554
1-Gram	**0.452**	**0.454**	**0.482**	**0.232**	**0.476**	**0.562**

Table 12: Effective of Query Augmentation vs Document Augmentation

	nCDG@1	nCDG@5	nCDG@10	Recall@1	Recall@5	Recall@10
HyDE (Gao et al., 2022b)	0.427	0.442	0.478	0.221	0.472	0.576
Ski-Q-1	0.452	0.454	0.482	**0.232**	0.476	0.562
Ski-QC-1	**0.455**	**0.458**	**0.493**	0.230	**0.480**	**0.584**
Ski-QC-ASM	0.398	0.398	0.435	0.193	0.429	0.541