From RAG to RICHES: Retrieval Interlaced with Sequence Generation

Anonymous ARR submission

Abstract

We present RICHES, a novel approach that interleaves retrieval with sequence generation tasks. RICHES offers an alternative to conventional RAG systems by eliminating the need for separate retriever and generator. It retrieves documents by directly decoding their contents, constrained on the corpus. Unifying retrieval with generation allows us to adapt to diverse new tasks via prompting alone. RICHES can work with any Instruction-tuned model, without additional training. It provides attributed evidence, supports multi-hop retrievals and interleaves thoughts to plan on what to retrieve next, all within a single decoding pass of the LLM. We demonstrate the strong performance of RICHES across ODQA tasks including attributed and multi-hop QA.

1 Introduction

Large language models (LLMs) have increasingly become the backbone for much of natural language processing and there has been a push to formulate a wide range of tasks as sequence to sequence transduction. However, when LLMs need to interact with non-parametric knowledge in the form of an external evidence corpus, the typical approaches chain LLM generations with calls to a separate retrieval model as part of a multi-system pipeline. In this paper we introduce a new approach, RICHES (Retrieval Interlaced with Sequence Generation) which can natively interleave text generations with retrievals from an evidence corpus using a single LLM and decoding process.

RICHES builds on previous work that demonstrated the application of *constrained decoding* to retrieval over a corpus (Jain et al., 2023; Bevilacqua et al., 2022) but extends this work to support multiple retrievals, entwined in a standard text generation procedure. In this approach, we retrieve documents by directly decoding their contents or related natural language *retrieval keys* that point to

Figure 1: Example RICHES outputs for multi-hop queries with a single LLM and decoding pass. The green quoted text is "retrieved" or generated verbatim from the retrieval corpus. RICHES generation natively interleaves thoughts and multiple retrieval evidences.

the documents they were generated from. For example, Figure 1 illustrates a solution from RICHES to multi-hop question answering (Yang et al., 2018), where evidence must be retrieved from multiple separate documents, by iteratively generating an unconstrained 'thought' about what needs to be retrieved and then generating a supporting proposition derived from an evidence corpus and tied to an original piece of supporting text. RICHES executes this task in a single decoder pass. For this example task, which is evaluated alongside others in Section 6, we have built on recent advances in chain-of-thought reasoning via prompting alone (Yao et al., 2022) but have directly integrated the retrieval step without needing to account for any interaction with an external retrieval system.

The observations we build this work on are:

1. *LLMs are knowledge warehouses*: They internalise and generalise over vast quantities of training data and are often able to generate surprisingly accurate knowledge in response

to complex inputs (Sun et al., 2022). However they are also susceptible to *hallucination* and cannot account for fresh knowledge, not available at the time of training. That is where retrieval shines.

2. *LLM decoding is a search process*: Language model decoders search for a single sequence in the set of all possible token sequences (Graves, 2012). Retrievers just need to constrain this search space to those sequences that are known to exist in a corpus of interest.

3. *Unifying tasks unlocks rapid development via prompting* By unifying retrieval with generation in a single decoder pass, we create a system that can be adapted to diverse new tasks via prompting alone, directly benefiting from the advances in instruction following. We later show that RICHES works with an off-the-shelf instruction-tuned model, without any additional training. This is in contrast to pipelines that need to be rebuilt/retrained on a task-by-task basis.

There is an another advantage of using language models as search agents. Of the two core operations in retrieval, indexing and search, indexing is constrained by corpus size, while search typically depends only on the index structure. Using large language models for indexing billion-token corpora is highly expensive, but search does not face the same bottle-neck. This enables us to unlock the knowledge stored in very large models for retrieval.

This work overlaps with a variety of related work focusing on retrieval, retrieval augmented generation (Lewis et al., 2020), reasoning in language models, and open domain question answering. We discuss their connections to RICHES in Section 2, then introduce the key components of the generalizable RICHES approach in Section 3.

While RICHES is applicable to any task that can be reduced to an interleaved generation of unconstrained text and pre-defined retrieval keys, we validate the approach with tasks in open domain question answering and show how it natively supports single-hop question answering, including the case where attribution to a source text is required; multi-hop question answering; and interleaving retrieval with 'planning steps' that enhance the retrieval performance. Results are presented in Section 6.2 along with qualitative examples and analysis in Section 6.3 to help motivate the approach.

2 Related Work

Retrieval Augmented Generation (RAG) ODQA tasks predominantly employ the RAG approach (Lewis et al., 2020) where typically a dense retriever (Karpukhin et al., 2020) retrieves documents from an evidence corpus and feeds to a language model for the final answer. These pipelines involve switching between heterogeneous models and are hard to train in concert. Moreover, Dense retrievers fail to generalize out-of-domain (Thakur et al., 2021).

Generative Retrieval (Metzler et al., 2021) techniques shifting the onus of Search from non-parametric nearest neighbor scan to language models. Differentiable Search Index (Tay et al., 2022) memorizes a mapping of query to opaque document identifiers, however memorization struggles to generalize to unseen corpus (Pradeep et al., 2023). An alternative approach is to use natural language keys as document identifiers, where keys are constrained decoded to lie in the corpus (De Cao et al., 2020; Bevilacqua et al., 2022). These systems still need an external model to generate answers. 1-Pager (Jain et al., 2023) unifies evidence and answer generation, by generating a sequence of keywords that map to a document. However, isolated keywords limit context understanding and suffer similar pitfalls as lexical matching.

Recitation Separate from retrieval augmentation, language models have been shown to recite entire passages from memory (Sun et al., 2022; Yu et al., 2022). But these passages are prone to hallucination. Our aim is to intersect contextual passage generation with corpus grounding. GopherCite (Menick et al., 2022), a noteworthy work in this direction, generates quotes verbatim from a small set of documents using constrained decoding. RICHES aims to scale this to a billion-token corpus.

Iterative reasoning and Search In recent times, there have been several efforts to improve multi-hop question answering by better reasoning (Asai et al., 2023) and planning (Adolphs et al., 2021; Yao et al., 2022). Language models have also been applied to the task of search to explore alternative paths (Yao et al., 2023; Hao et al., 2023).

Our work builds on these advances in reasoning while integrating search within generation.

3 Retrieving while Generating

We present a method of interleaving unconstrained text generation with the generation of *retrieval keys* that point into a retrieval corpus. For example, Figure 1 shows generations that interleave unconstrained 'thoughts' with evidence sentences drawn from a predefined corpus for a multi-hop question answering task. Later in this section we'll introduce a number of different choices of retrieval key as well as a variety of tasks that benefit from interleaved generation and retrieval. However, for now we simply define a retrieval key as a sequence of tokens that exists in a pre-defined finite set of sequences K where every entry is associated with one or more documents in an underlying corpus C.

Formally, we focus on the sequence to sequence transduction task where we predict an output sequence $\mathbf{y} = [y_0, \ldots, y_n]$ conditioned on an input sequence $\mathbf{x} = [x_0, \ldots, x_m]$ and we mark the start and end of a retrieval key in y with special markers « and ». If we let $Q(\mathbf{y})$ be a function that returns all retrieval key spans from y (i.e. $(i, j) \in Q([y_0, \ldots, «, y_i, \ldots, y_j», \ldots, y_n])$) then we can update the standard autoregressive language modeling probability

$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{i=0}^{|\mathbf{y}|} P(y_i|y_0, \ldots, y_{i-1}, \mathbf{x}, \theta) \quad (1)$$

to include the indicator function $\mathbb{1}_K(\mathbf{q})$ that maps elements of K onto one and otherwise to zero.

$$\begin{aligned} P_\theta(\mathbf{y}|, \mathbf{x}, K) = \frac{1}{Z} \prod_{\mathbf{q} \in \mathcal{Q}(\mathbf{y})} \mathbb{1}_K(\mathbf{q}) \\ \times \prod_{i=0}^{n} P(y_i|y_0, \ldots, y_{i-1}, \mathbf{x}, \theta) \end{aligned} \quad (2)$$

where Z is a normalizing term that accounts for the probability mass assigned by Equation 1 to disallowed sequences. In practice, we do not need to compute Z and can sample from Equation 2 in the usual way, one token at a time, by simply zeroing out the probability of disallowed continuations as presented in Section 3.1.

3.1 Constrained Beam Decoding

We opt for Beam Search (Graves, 2012) as our decoding strategy to simulate a heuristic Best-first search. Here, the action or next node space is the entire vocab. At each time step, the LLM estimates the value of each node (token) given the paths explored so far and adds them to the fixed-size queue (Beam). Figure 2 visualizes how the beam progresses over decoding timesteps. Unlike regular beam decoding where the top decoded sequences have only small variations, constraints impose sparsity over the search space resulting in diverse beams. In Section 3.3, we discuss how beam can hurt unconstrained generation and suggest hybrid decoding strategy as workarounds. Constrained decoding can also gain from more sophisticated algorithms such as value-based decoding (Ren et al., 2017), look-ahead scoring and planning (Lu et al., 2021; Hao et al., 2023).

3.2 Efficient Constraints via the FM-Index

During decoding, model outputs are constrained to the corpus by masking out any continuation not in the corpus. To compute the continuations of a sequence, we use FM-index (Ferragina and Manzini, 2000), a compressed suffix array augmented with additional data structures to support fast substring search operations. Unlike a Trie structure, it is also highly space economical due to the compression. Given a prefix, FM-Index can efficiently compute the next allowed tokens in O(Vocab), independent of the corpus-size. Below is the pseudo code for the modified decoding process.

```
def constrain(input_prefix):
  # Fetch continuations for prefix
  allowed_tokens = fm_index.
    get_continuations(input_prefix)
  # Get next token probabilities
  logprobs = LLM.logprobs(input_prefix)
  # Disallowed tokens are set to -inf
  for i in logprobs:
    token = vocab[i]
    if token not in allowed_tokens:
        logprobs[i] = -np.inf
  return logprobs
```

3.3 Adaptive Beam Size

In Section 5.2 we introduce some tasks that interleave constrained and unconstrained generation. The constrained generations must be precise—to match the target retrieval key exactly. The unconstrained generations are generally more robust to small variations in surface form—these only need to convey the correct information to a reader, or to provide the model room for a 'thought' trace when reasoning about a response.

To ensure that RICHES can properly make use of beam search, which is here intended to ensure the model does not get stuck irretrievably after generat-

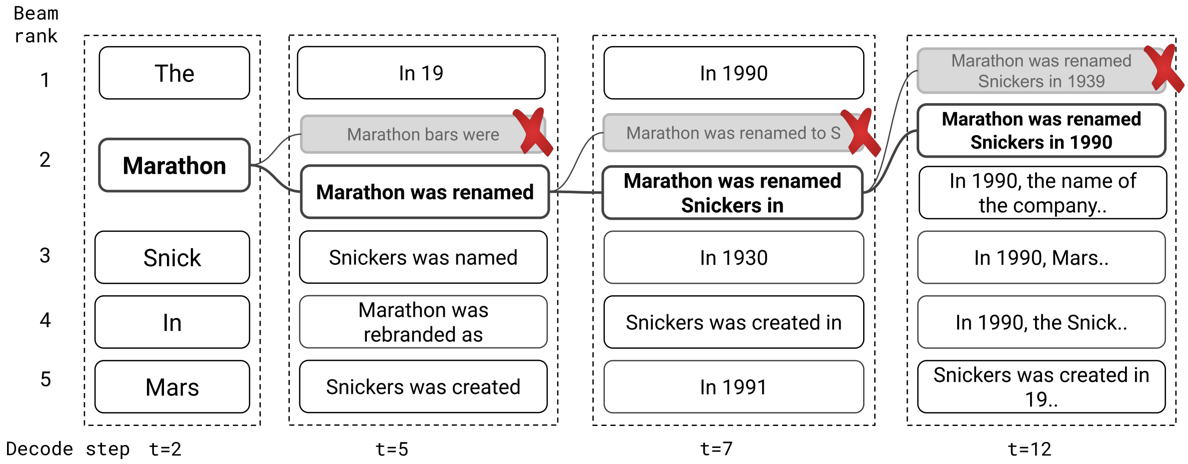

Figure 2: Visualization of constrained beam for query: *"when did marathon change its name to snickers?"*. The final RICHES output is *"Marathon was renamed Snickers in 1990"*. Bold boxes track the progress of the top-beam sequence. Grey crossed out boxes are sequences that the LLM preferred, but were blocked by corpus constraints.

ing an incorrect constrained prefix, we introduce an adaptive decoding strategy that switches between full beam decoding for the sensitive constrained sequences but opts for greedy decoding when unconstrained. In practise, a constrained prefix is expanded to next beam-size probable tokens while an unconstrained prefix is expanded to only the next one token. This is expected to provide room for rest of the beam to be utilized largely for constrained sequences. Section 6.1 shows experiments with multiple decode modes.

3.4 Indexing Strategies

The FM-Index used by RICHES supports efficient indexing of all sub-strings in a corpus, which is useful when we want to generate corpus text verbatim. However, it is not clear that this is the best option of retrieval key for the auto-regressive decoder in Section 3.1. A key question in index construction is the *document representation* used in indexing. In traditional lexical-based retrieval systems, documents are represented by the terms in it, with transformations such as stemming, weighing by corpus statistics (Robertson et al., 2009). Neural retrieval systems transform raw text into dense vector representations and offload representation computation to the neural network. But even in this case, proper document chunking and/or multi-vector document significantly impact final performance (Lee et al., 2021; Khattab and Zaharia, 2020).

In this section, we introduce a few different choices of retrieval keys, including a *propositional index* that requires indexing time neural computation. A key consideration here is the interplay between the retrieval index and the search strategy.

Document Title and Section Headers Many retrieval corpora such as Wikipedia have consistent structures in the form of titles and sometimes subtitles and metadata. This provides a hierarchical structure such that one can first decode titles, subtitles and then the document.

Paragraph Sub-string A natural option for retrieval key is any sub-string of the unit of text being indexed itself. In most open domain question answering approaches, paragraph is the de-facto unit of evidence. We can index paragraphs efficiently using the FM-index (Section 3.2) and decode sub-strings directly with RICHES to get pointers into the retrieval corpus. It should be noted that this yields an inherently many-to-many mapping between paragraphs and retrieval keys, but that the mapping is in-effect one-to-one for longer sequences of tokens.

Sentence Sub-string Similarly, individual sentences form a natural retrieval key. Sentence are smaller units of information than passage, but may not be interpretable stand-alone.

Propositional Index The above choices do not perform any non-trivial indexing step, unlike standard approaches in information retrieval where documents are mapped to sparse or dense vectors. The omission of this indexing step may be desirable but it also forces RICHES to deal with the non-uniform and diffused information in raw text. An alternative that is closer, in intent, to the offline indexing step used by other IR systems, is to map each indexed chunk to a set of uniformly structured propositions (Min et al., 2023; Chen et al., 2022). A proposition is a stand-alone unit that efficiently encodes small, atomic chunks of factual information. For example, instead of the sentence "He has 7M followers on Twitter" a proposition would be decontextualized to "Tom Cruise has 7M followers on Twitter." We adopt a pre-existing propositional index from Chen et al. 2023 described in Section 5.1.

Section 6.1 compares various Retrieval keys for the ODQA task with illustrations in Appendix A.4.

4 Interleaving Retrieval and Generation

We have presented a method of interleaving unconstrained text generation with constrained generation of retrieval keys. In this section we introduce a handful of tasks that make use of this interleaving either as a core task requirement, or as a means to an end by interleaving 'thoughts' with retrieval actions to help guide search.

Attributed Question Answering We apply RICHES to the open domain question answering (ODQA) task where we score both the ability to correctly predict a short answer string and retrieve attribution for that answer (Bohnet et al., 2022). See Table 1 for examples.

Multi-hop Question Answering Interleaving between generation and retrieval can be powerful in multi-hop reasoning, where the model needs to retrieve and stitch together knowledge from multiple sources. Examples of RICHES outputs for multi-hop QA are given in Table 2.

"Thinking" for Retrieval Multi-step questions often require breaking down a query into smaller steps and reasoning or planning what to retrieve next. Foreshadowing retrieval with thoughts is crucial in this context. It helps direct the retrieval process, avoid repetitions, and, more importantly, allows for iterating upon and correcting previously erroneous retrievals. A few such demonstrations can be found in Table 2.

5 Experimental Setup

5.1 Datasets

Queryset Our experiments are focused on open domain question answering tasks including both single and multi-hop benchmarks. For single-hop, we use the Open-NQ (Kwiatkowski et al., 2019) dataset. To evaluate multi-hop reasoning, we look into Hotpot-QA (Yang et al., 2018) and Musique-Ans (Trivedi et al., 2022). The latter includes varying hops and different composition operations, offering a rich test-bed for how well RICHES can generalize across a diverse range of queries.

Corpus Section 3.4 describes multiple strategies to index the corpus. Each type of retrieval key needs to be accompanied with its own corpus. Title, passage and sentence keys are derived from the Wikipedia corpus presented in Bohnet et al. 2022. For propositions, we re-use the Factoid-Wiki corpus built by Chen et al. 2023. This is derived from Bohnet et al. 2022 by decomposing passages into smaller, compact propositions using a finetuned Flan-T5-large (Wei et al., 2021) model. We drop the titles from Factoid-Wiki and only use the propositions (See Appendix A.2).

5.2 Evaluation

The standard metric for ODQA benchmarks has predominantly been F1 answer match accuracy. However, language models are prone to hallucinate and F1 stand-alone can be misleading as the answer may not be conditioned on the evidence. Attribution (Rashkin et al., 2021) helps us trade-off answer accuracy for faithfulness to the evidence. Thus, we measure two competing metrics: i) end-to-end answer accuracy with F1 and ii) attribution of the answer to evidence using AutoAIS (Bohnet et al., 2022). AutoAIS, or AIS for short, is automatically computed by classifying whether the evidence text entails the question and predicted answer pair. We re-use the NLI scorer and formulation from Bohnet et al. 2022 (See details in Appendix A.2). The evidence text here is the concatenation of all retrieval keys in the RICHES output. The unconstrained thoughts are discarded from evaluation. Only the *top beam output* is considered for evaluation.

5.3 Models and Inference

Throughout our experiments, we use off-the-shelf instruction-tuned models in a few-shot setting, without any fine-tuning. We test the instruction-tuned versions of PALM2-M and its larger variant PALM2-L (Anil et al., 2023) based on stacked Transformer architecture. We use 3 example demonstrations in our prompt (Appendix A.1), with different sets of examples for single-hop (NQ) and multi-hop (Hotpot, Musique) datasets. The unconstrained sequences or thoughts are formulated as

Query	Dense Retriever (GTR-xxl)	RICHES Retrieval Output
`# Leveraging Parametric knowledge`		
who was the actor that played ben stone on law and order	Benjamin Stone was a former Law and Order character	Ben Stone was played by Michael Moriarty
jharkhand which festival is associated with cattle worship	Gaijatra is the festival of cows	Sohrai is a cattle festival of Jharkhand.
`# Better Context understanding`		
who plays at the prudential center in newark	The Prudential Center is in Newark.	The Prudential Center is home to the New Jersey Devils
when was coffee first made into a drink	In 1724, society in Lima discovered an exotic drink called coffee.	The earliest credible evidence of coffee-drinking or knowledge of the coffee tree appears in the middle of the 15th century.

Table 1: Example of RICHES vs Dense Retrieval for single-hop QA. Only the retrieved text is shown for illustration.

Query	RICHES Output
who had a baby at 100 in the bible `(# Keywords guide retrieval)`	keyword: Sarah's age at birth of Isaac « Sarah was 90 years old when Isaac was born. » keyword: Abraham's age at birth of Isaac « Abraham was 100 years old when Isaac was born. » answer: Sarah and Abraham
when did the united states start using the death penalty `(# keywords help recover from inaccurate retrieval)`	keyword: Capital punishment in the United States « Capital punishment was reinstated in the United States in 1976. » keyword: First execution in the United States « The first recorded execution in the United States took place in 1608 at the Jamestown Colony in Virginia. » answer: 1608
What distinction is held by the former NBA player who was a member of the Charlotte Hornets during their 1992-93 season and was head coach for the WNBA team Charlotte Sting? `(# Planning what to retrieve)`	keyword: member of the Charlotte Hornets during their 1992-93 season « Muggsy Bogues played for the Charlotte Hornets from 1988 to 1997. » keyword: Muggsy Bogues' distinction « Muggsy Bogues is the shortest player ever to play in the National Basketball Association (NBA). » answer: shortest player ever to play in the National Basketball Association (NBA)

Table 2: Example Iterative retrieval outputs from RICHES. Remarks are annotated as `(# Comments)`

hint keywords. Our final setup uses a beam of 10 with constrained decoding (Section 3.1), adaptive beam size (Section 3.3) and propositions as retrieval keys. Later in Section 6, we ablate these choices. Note that only the *top-beam* result is considered for evaluation.

5.4 Baselines

We experiment with 2 types of baselines: the standard dense retriever and an iterative retriever suited for multi-hop QA. Since, RICHES itself is not trained on any in-domain task, we opt for setups that test the out-of-domain performance of our baselines as well.

Generalized Dense Retriever For single-hop QA, we compare our approach against the Generalized T5 retriever (GTR-xxl, 11B variant) (Ni et al., 2021). GTR undergoes multi-staged training, first on unsupervised web-mined corpus and then supervised search datasets including NQ. It has been shown to generalize well out-of-domain. However, GTR and other conventional dense retrievers provide only retrieved documents, not the answers themselves. To extract answers, we use the PALM2-M model in a few-shot setting (see Appendix A.1 for the details).

Since RICHES generates a single output with a varying number of interleaved documents, direct comparison with dense retrievers that fetch a fixed top-k documents is challenging. We set k to a value equivalent to the mean documents RICHES fetches for single-hop. When retrieval keys are different, such as passages vs propositions, we approximately match the tokens used by both setups. In our final experiments, we compare against k=1 passage and k=2 propositions for GTR-xxl.

Iterative Retrieval (*Iter*) For Multi-hop QA, we adopt a popular method where question is decomposed into sub-queries (Khot et al., 2022). At each step, passages are retrieved for a sub-query and fed as input for the next query, until one converges to an answer. The method has the same surface form as RICHES, except for the key distinction that each step requires switching between a heterogeneous mix of models. In our experiments, we retrieve top-1 document with GTR-xxl and use PALM2-M few-shot for both decomposing the query and generating the final answer (See prompt at Appendix A.1). Max allowed steps is set to 4 where most of the queries converge.

6 Results

Retrieval Key	Hits@1
Title	19.5
Paragraph with Title	15.5
Paragraph	19.0
Sentence with Title	19.1
Sentence	20.6
Proposition	33.9

Table 3: Comparison of Retrieval Keys on NQ

In the following sections, we investigate the key building blocks of RICHES: i) indexing strategies (Section 3.4) amenable to auto-regressive decoding ii) effect of beam decoding (Section 3.1) iii) suitable mechanisms to interleave thoughts and retrieval keys (Section 3.3). Finally, we compare RICHES against conventional retrieval systems. We also draw a detailed analysis of wins and losses to fathom the strengths and pitfalls of the system.

6.1 RICHES building blocks

Retrieval Keys We explore the following retrieval key candidates as detailed in Section 3.4: a) *Title*: Wikipedia page and section titles, ranking paragraphs within the section using TF-IDF scores. b) *Paragraph with Title*: Decodes the page title, section title, and full paragraph. c) *Paragraph*: Decodes the paragraph only. d) *Sentence*: Uses individual sentences. e) *Proposition*: Uses atomic information units derived from paragraphs. Table 3 shows that among the retrieval keys explored, the propositional index is best aligned with our decoding search strategy, perhaps its compact nature is most suited for autoregressive decoding. An in-depth analysis of retrieval keys is provided in Appendix A.4. In the following experiments, we use proposition as our retrieval key.

Effect of Beam size Table 5 shows how greedy decoding can get stuck with poor retrieval keys. A larger beam enables better search space exploration, albeit with diminishing returns. In our final experiments, we use a beam of 10.

Interleaving with Adaptive Beam Table 6 shows the impact of interleaving thoughts with retrieval keys. First, we note that an adaptive beam is crucial for interleaving unconstrained and constrained sequences. Without an adaptive beam, minor irrelevant variations in unconstrained thoughts can consume and overwhelm the available space in the beam. By greedily decoding unconstrained sequences, the beam space is preserved for backtracking during document search. Once we have an adaptive beam in place, the insertion of keywords enhances both answer and retrieval performance, reminiscent of chain-of-thought technique to enable better retrieval.

6.2 Overall Results

Table 4 shows the overall performance of RICHES across various datasets. For single-hop tasks, RICHES competes well with dense retrievers, offering higher answer accuracy at the expense of some attribution. In multi-hop QA, RICHES excels, outperforming iterative baselines by +15 F1 points on Hotpot and +11 on Musique, with comparable or better attribution. The increase in answer accuracy with the larger PALM2-L model suggests improved performance with larger model sizes. Notably, RICHES achieves these results with a single inference pass, unlike the Iterative baseline, which requires a model call at each sub-query step.

6.3 Qualitative analysis

We inspect 50 win and loss examples each to analyze the strength and weaknesses of the system.

Wins Several properties distinguish RICHES from dense retrievers: a) RICHES allows large language models to utilize their parametric knowledge for retrieval. Since the search operation in RICHES is independent of corpus size, it can employ much larger models at query time. b) The inherent alignment of instruction-tuned models enables them to retrieve contextually relevant passages, whereas

Retriever	Answerer	NQ F1	NQ AutoAIS	Hotpot F1	Hotpot AutoAIS	Musique F1	Musique AutoAIS
Dense Retrieval							
GTR Passage	PaLM2-M	41.9	48.7	34.9	19.6	7.2	17.9
GTR Proposition	PaLM2-M	36.6	63.2	27.4	18.5	10.5	20.4
Iterative	PaLM2-M	34.4	66.8	34.2	30.9	17.5	38.4
RICHES							
	PaLM2-M	40.2	59.2	41.0	36.5	19.1	39.6
	PaLM2-L	46.7	59.6	51.1	35.6	28.2	37.5

Table 4: Overall performance comparison for RICHES. For Dense retrievers, top-k documents are retrieved and fed to the few-shot Answerer, where k=1 for GTR passage, k=2 for GTR propositions. For Iterative retrieval upto 4 documents are retrieved with k=1 at each step.

Beam	F1	AutoAIS
1	19.3	26.1
5	35.8	58.7
10	40.2	59.2

Table 5: Effect of Beam size on NQ with PaLM2-M.

Failure mode	Queries(%)
Index Failure	40%
Search Failure	52%
Attribution Failure	8%

Table 7: Loss categories for RICHES on Hotpot-QA

Unconst. Keywords	Adaptive Beam	NQ F1	NQ AIS	Hotpot F1	Hotpot AIS
X	X	37.9	57.5	39.2	33.9
✓	X	36.9	51.5	38.4	32.3
✓	✓	40.2	59.2	41.0	36.5

Table 6: Interleaving unconstrained keywords and retrieval keys with Adaptive beam. Greedily decoding Unconstrained sub-sequences allows constrained retrievals to make the most of the beam search.

dense retrievers may sometimes latch onto keywords. c) The interleaved thoughts guide the model toward more accurate retrievals. Table 1 demonstrates these scenarios for single-hop retrievals and Table 2 for multi-hop retrievals.

Can the model retrieve what it doesn't know? A language model may hold stale or incorrect information. However, RICHES can often override model's pre-existing knowledge and generate correct answers by constraining on the corpus (Appendix A.3)

Losses We inspect 50 failed queries and categorize the losses (Table 7) as follows: a) Index failure: the proposition is absent from the index or not decontextualized. b) Search failure: Proposition exists in the index, but could not be generated c) Attribution failure: The answer is partially attributed, with LLM hallucinating based on partial evidence. (see Appendix A.3 for examples)

7 Conclusion

Retrieval has so far been alienated from the rapid progress in instruction tuning. This work makes the following contribution: i) an approach that can seamlessly integrate retrieval with generation. ii) a thorough investigation of indexing and search strategies that enable such an approach to be effective. iii) proof-of-concept of the capabilities of such a system on a variety of QA tasks. We hope the ideas introduced in this work fuel progress in aligning retrieval to generation and simplifying it.

8 Limitations

First we note the limitations in our experimental setup. All our experiments are based on Wikipedia, a corpus heavily seen during pre-training. This work does not analyze how RICHES fares on corpora unseen during pre-training. Furthermore, we only examine a handful of factoid question-answering tasks due to the lack of objective evaluations. Performance on tasks such as long-form QA is deferred for future work. There are also certain inherent limitations with RICHES. RICHES forces verbatim emission of corpus text, which might be an overkill for tasks where a similarity-based metric is sufficient. For long documents with diffused information, rewriting into propositions adds complexity and can be cumbersome. Lastly, while RICHES's search operation is independent of corpus size, the use of beam search and communication between the FM-index and Transformer model can slow down inference.

9 Ethical Considerations

All artifacts used in this paper, including models, datasets, and baselines, are under permissive licenses and publicly available. We have attempted to provide detailed information to facilitate the reproduction of our results.

Our findings are based on English-language data from Wikipedia, and we have not tested the generalizability of our claims to other languages or domains.

Lastly, the datasets used in this work are not expected to contain any offensive content. However, it is important to note that Large Language Models (LLMs) can exhibit biases related to gender, race, and region, and are also prone to hallucination. Although RICHES aims to ground its generation in an external corpus, some biases may still be present.

References

Leonard Adolphs, Benjamin Boerschinger, Christian Buck, Michelle Chen Huebscher, Massimiliano Ciaramita, Lasse Espeholt, Thomas Hofmann, Yannic Kilcher, Sascha Rothe, Pier Giuseppe Sessa, et al. 2021. Boosting search engines with interactive agents. *arXiv preprint arXiv:2109.00527*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

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*.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *arXiv pre-print 2204.10628*.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2022. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. *arXiv preprint arXiv:2212.10750*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *Proceedings 41st annual symposium on foundations of computer science*, pages 390–398. IEEE.

Alex Graves. 2012. Sequence transduction with recurrent neural networks.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Palak Jain, Livio Baldini Soares, and Tom Kwiatkowski. 2023. 1-pager: One pass answer generation and evidence retrieval. *arXiv preprint arXiv:2310.16568*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

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.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase retrieval learns passage retrieval, too. *arXiv preprint arXiv:2109.08133*.

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.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2021. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: making domain experts out of dilettantes. *ACM SIGIR Forum*, 55(1):1–27.

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. *arXiv preprint arXiv:2305.14251*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam D Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q Tran. 2023. How does generative retrieval scale to millions of passages? *arXiv preprint arXiv:2305.11841*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*.

Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 290–298.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

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*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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*.

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, 2023. *URL https://arxiv. org/pdf/2305.10601. pdf*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

A Appendix

Dataset	Split	Queries	Hops
Open-NQ	Test	3610	1
Hotpot	Dev	7405	2
MuSiQue-Ans	Dev	2412	2-4

Table 8: ODQA Datasets used in our experiments

Corpus	Docs	Avg Words
Passage	40M	58.5
Sentence	114M	21.0
Propositions	256M	11.0

Table 9: Retrieval Corpora used in our experiments

A.1 Experiment Details

In-context prompts We use 2 different sets of few-shot demonstration for single-hop (NQ) and multi-hop (Hotpot, Musique) datasets displayed in Table 10 and Table 11 respectively. Both prompts carry the same instruction, but the multi-hop variants provides demonstrations with multiple evidence passages.

Computing constraints An example of constrained decoding is illustrated in Figure 3.

Baselines For the dense-retriever baseline, answers are extracted from retrieved passages with an external reader. We use PALM2-M with a few-shot prompt (Table 12).

For iterative retrieval baseline, we use PALM2-M for both query decomposition and answering. At each step, the model can choose to generate a sub-query or the final answer. The unified prompt is provided at Table 13.

A.2 Evaluation

Datasets We use Musique-Ans (Trivedi et al., 2022) subset of Musique which consists of answerable queries. Details of query sets evaluated can be found in Table 8. To make retrieval challenging, we use the full Wikipedia corpus for retrieval (Table 9). This is different from the typical Hotpot and Musique setting which use the first Wikipedia paragraph (5M documents) and documents associated with query-set (1.3M) respectively.

Metrics AutoAIS is an automated way of measuring AIS (Attributable to Identified Source) (Rashkin et al., 2021). AutoAIS formulates evaluation as a Natural Language Inference task that asks a model whether the question and answer are entailed by the provided evidence. We re-use a T5-11B checkpoint finetuned on a collection of NLI tasks from (Bohnet et al., 2022). Question answering is formulated into NLI task as follows:

`hypothesis: <retrieved evidence1> <retrieved evidence2> ... premise: The answer to the question '<question>' is '<predicted answer>'` The NLI scorer provides a calibrated score between 0 (not attributed) to 1 (completely attributed) which is averaged over the dataset to get the final AutoAIS score.

A.3 Extended Qualitative Analysis

We provide examples for loss categories defined in Section 6 in Table 14. Table 15 showcases a few selected examples where the unconstrained model emits incorrect answer, but constraining on the corpus guides it towards correct answer.

A.4 Index representation qualitative analysis

In this section we analyse retrievals from a few select examples from the OpenNQ development dataset. We compare retrievals using different document representations in the index, including: **title + tf-idf**, **passage**, **sentence** and **proposition** (see Section 3.4).

Question who holds the world record for the most world records
Gold Answer Ashrita Furman

- **title + tf-idf**: List of world records in track cycling. World records in the sport of track cycling are ratified by the Union Cycliste Internationale (UCI).
- **Passage**: According to the Guinness Book of World Records, the record for the largest event was set in 2009.
- **Sentence**: As of January 2021, he has 1.7 million followers on Twitter
- **Proposition**: ***Ashrita Furman*** holds the Guinness world record for the most Guinness world records.
- **Source text for proposition**: Ashrita Furman (born Keith Furman, September 16, 1954) is a Guinness World Records record-breaker. As of 2017, Furman has set more than 600 official Guinness Records and currently holds 531 records, thus holding the Guinness world record for the most Guinness world records.

Analysis In this example, only the propositional-based retrieval was able to retrieve a correct answer. Part of the reason why passage or sentence representation is hard to retrieve with auto-regressive decoding is that the main evidence for this answer in the "source text for proposition" comes at the end of a complex sentence ("[...] thus holding the Guiness world record for the most Guiness world records").

Question who has the most number one single hits
Gold Answer The Beatles

For given input query, write 1-3 passages to answer the query. Write a hint keyword and a passage contained within « and ». A passage must be a complete sentence and not a phrase. It must contain complete context for answering the query and should not begin with it, he, they etc. Do not repeat any passages. Aim for new keywords.

question: The football manager who recruited Cristiano Ronaldo managed Manchester United during what timeframe?
passage: keyword: Cristiano Ronaldo's recruiting manager « Alex Ferguson recruited Cristiano Ronaldo » keyword: Sir Alex Ferguson's tenure at Manchester United « Sir Alex Ferguson managed Manchester United from 1986 to 2013. »
answer: 1986 to 2013

question: Were Eatza Pizza and Your Pie founded in the same state?
passage: keyword: Eatza Pizza founded in state « Eatza Pizza was founded in Arizona » keyword: Your Pie founded in state « Your Pie was founded in Athens, Georgia »
answer: no

question: In which stadium do the teams owned by Myra Kraft's husband play?
passage: keyword: Myra Kraft's husband « Robert Kraft's wife is Myra Kraft. » keyword: Robert Kraft's team « Robert Kraft is the owner of the New England Patriots. » keyword: New England Patriots stadium « Gillette Stadium is the home of the New England Patriots. »
answer: Gillette Stadium

question: <question>
passage:

Table 10: Few-shot prompt used in RICHES for multi-hop QA

- **title + tf-idf**: List of Billboard Hot 100 number-one singles of the 1990s. This article is about the American Billboard Hot 100 chart held during the 1990s.
- **Passage**: According to the Recording Industry Association of America (RIAA), the album has sold more than 500,000 copies in the United States,
- **Sentence**: As of 2021, ***The Beatles*** remain the act with the most Christmas number-one singles in the United Kingdom.
- **Proposition**: ***The Beatles*** hold the record for most number-one hits on the Billboard Hot 100.
- **Source text for proposition**: The Beatles, Awards and achievements. As of 2017, they hold the record for most number-one hits on the Billboard Hot 100, with twenty.

Analysis The correct answer is retrieved in this example using both sentence and proposition based representations. However, the sentence representation here does not provide proper justification for the answer. Note that the propositional representation correctly decontextualizes the pronoun in "[...] they hold the record [...]" to "The Beatles hold the record [...]" making the retrieval easier using constrained decoding.

Question how many episodes of sabrina the teenage witch are there
Gold Answer 163

- **title + tf-idf**: Sabrina the Teenage Witch (1996 TV series). The first four seasons aired on ABC from September 27, 1996 to May 5, 2000. The final three seasons ran on The WB from September 22, 2000 to April 24, 2003.
- **Passage**: Sabrina the Teenage Witch is an American television sitcom created by Nell Scovell, based on the Archie Comics series of the same name.
- **Sentence**: Sabrina the Teenage Witch is an American television sitcom created by Nell Scovell, based on the Archie Comics series of the same name.
- **Proposition**: Sabrina the Teenage Witch had ***163*** episodes.
- **Source text for proposition**: This is an episode list for Sabrina the Teenage Witch, an American sitcom that debuted on ABC in 1996. From Season 5, the program was aired on The WB. The series ran for seven seasons totaling 163 episodes.

Analysis All retrievals using non-propositional representations select part of the main article for "Sabrina the Teenage Witch". This article, however, does not contain the answer to the question. In the propositional case, there is a straightforward proposition that is constructed from a passage from the "List of Sabrina the Teenage Witch episodes". Note that the source passage contains a reference that becomes ambiguous out-of-context ("The series" is decontextualized to "Sabrina the Teenage Witch" in the proposition).

Question what is dj's boyfriends name on full house
Gold Answers Steve Hale, Steven "Steve" Hale, rich kid Nelson, or Viper

- **title + tf-idf**: Full House (season 8). The eighth and final season of the ABC sitcom Full House originally aired between September 27, 1994 and May 23, 1995.
- **Passage**: Full House (1987–1995) and its Netflix sequel Fuller House.
- **Sentence**: In the 1990s, she appeared in the films Blues Brothers 2000
- **Proposition**: ***Steve Hale*** was D.J.'s boyfriend in seasons six and seven.

For given input query, write 1-3 passages to answer the query. Write a hint keyword and a passage contained within « and ». A passage must be a complete sentence and not a phrase. It must contain complete context for answering the query and should not begin with it, he, they etc. Do not repeat any passages. Aim for new keywords.

question: who is the owner of phoenix mall pune?
passage: keyword: Phoenix Market City owner « Phoenix Market City is developed by Phoenix Mills Limited. »
answer: Phoenix Mills Limited

question: what brings in more money nba or nfl?
passage: keyword: NFL revenues « NFL revenues are well over $10 billion per season. » keyword: NBA revenue « NBA amasses about $6 billion annually. »
answer: NFL

question: when was the french national anthem adopted?
passage: keyword: French national anthem « La Marseillaise became the national anthem of France. » keyword: La Marseillaise adoption « La Marseillaise was adopted by France in 1795. »
answer: 1795

question: question
passage:

Table 11: Few-shot prompt used in RICHES for single-hop QA

- **Source text for proposition**: Full House, Production, Casting. As babies, the children were played by Daniel and Kevin Renteria, and in season six, the roles of the twins were succeeded by Blake and Dylan Tuomy-Wilhoit. The last main character added was Steve Hale, who was D.J. 's boyfriend in seasons six and seven. He was played by Scott Weinger.

Analysis The source sentence with the correct answer presents a challenge for auto-regressive decoding since the sentence prefix focuses on an aspect unrelated to the question ("The last main character added [...]"). With propositionalization, the sentence structure becomes aligned with the question, but requires that the model already knows the answer to the question, given that the first entity in the sentece is the answer.

Question who is the girl in green day 21 guns
Gold Answer Lisa Stelly

- **title + tf-idf**: Boulevard of Broken Dreams (Green Day song), Music video. The video won six awards at the MTV Video Music Awards in 2005, most notably for Video of the Year. It also won Best Group Video, Best Rock Video, Best Direction, Best Editing, and Best Cinematography.
- **Passage**: "21 Guns" is a song by American rock band Green Day. It was released as the second single from their eighth studio album, 21st Century Breakdown (2009), and serves as the sixteenth track from the album. The single was released through Reprise Records on May 25, 2009 as a digital download and July 14, 2009 as a CD single.
- **Sentence**: "21 Guns" is a song by American rock band Green Day.
- **Proposition**: The girl in the music video is Teresa Lourenco.
- **Source text for proposition**: The music video for Ägainfeatures Kravitz with his girlfriend in his apartment (Gershon), whom he does not seem to be interested in. Similar to the song's lyrical content, he meets a girl (Teresa Lourenco), who works as a waitress in a restaurant/diner.

Analysis In this case, all retrievals fail to retrieve the correct answer. In the case of the proposition-based representation, the model decodes a proposition where the subject is an ambiguous reference ("The girl") which has not been properly decontextualized (the source passage above makes it clear that the reference is not related to the question). Interestingly, the source passage with the correct answer requires an inferential step and its proposition representations are been decontextualized properly. **Source text with correct answer**: *21 Guns (song), Music video. The video takes place with the band and the album's two protagonists Christian (Josh Boswell) and Gloria (Lisa Stelly) taking refuge in a white room after robbing a bank..*
Relevant generated propositions:

- The video takes place with the band and the album's two protagonists Christian and Gloria.
- Gloria is played by Lisa Stelly.

To properly retrieve this passage using proposition-based representation we would need to properly disambiguate "The video" to "21 guns" and perform inference over these two propositions. Alternatively, proposition generation could generate more complex propositions containing both pieces of information, such as: **The "21 Guns" video takes place with the protagonist Gloria, played by Lisa Stelly**.

Question how many seasons of vampire diaries r there
Gold Answer eight, or 8

- **title + tf-idf**: The Vampire Diaries. The Vampire Diaries is an American supernatural teen drama television series developed by Kevin Williamson and Julie Plec, based on the book series of the same name written by L. J. Smith. The series premiered on The CW on September 10, 2009, and concluded on March 10, 2017, having aired 171 episodes

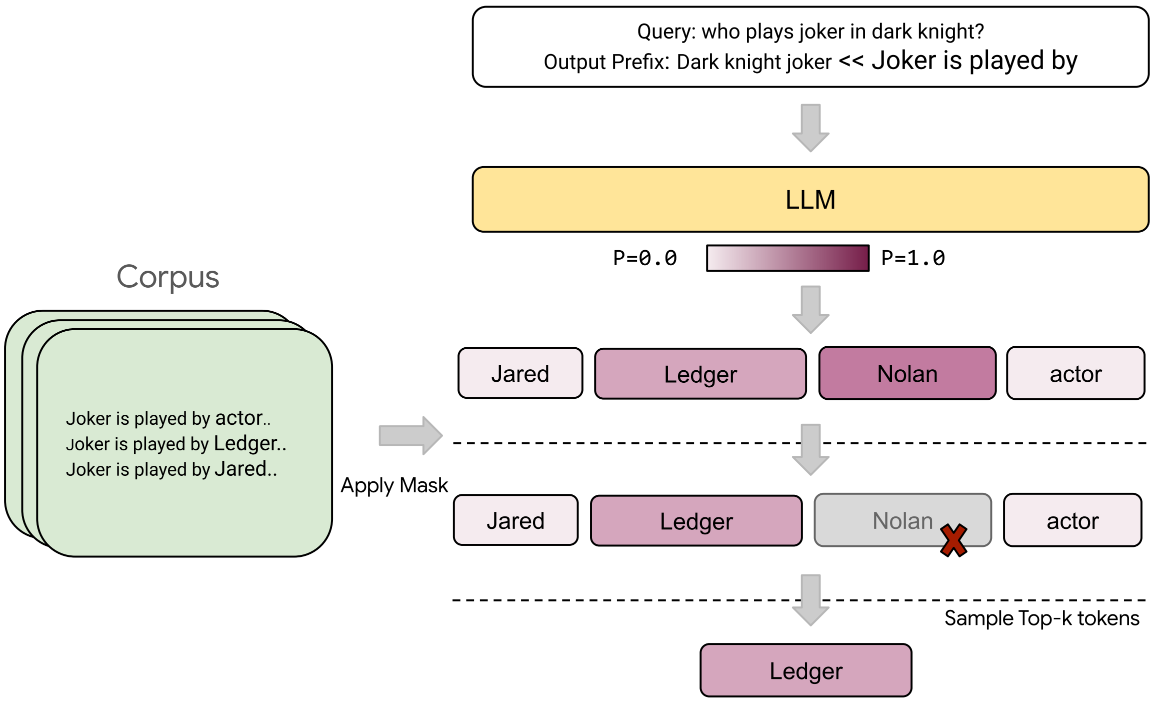

Figure 3: Illustration of the constrained decoding process. Given prefix, "Joker is played by", the continuation "Nolan" is not found in the corpus and therefore masked out.

over ***eight*** seasons.

- **Passage**: The Vampire Diaries is an American supernatural teen drama television series developed by Kevin Williamson and Julie Plec, based on the book series of the same name written by L. J. Smith. The series premiered on The CW on September 10, 2009, and concluded on March 10, 2017, having aired 171 episodes over ***eight*** seasons.
- **Sentence**: The series premiered on The CW on September 10, 2009, and concluded on March 10, 2017, having aired 171 episodes over ***eight*** seasons.
- **Proposition**: The Vampire Diaries is an American supernatural drama television series.
- **Source text for proposition**: The Vampire Diaries is an American supernatural drama television series that premiered on The CW on September 10, 2009, and concluded on March 10, 2017 after airing eight seasons.

Analysis In this case only the proposition-based representation retrieval is incorrect. We believe the retrieval fails here due to improper decontextualization of the correct answer passage. The sentence with the correct answer includes the proposition: *The series aired 171 episodes over eight seasons.*. Making it difficult for the model to

---

A.5 Computations involved

Evaluating the precise compute cost for RICHES depends on the specific implementations of the decoding algorithm, but we can sketch the key operations involved in retrieval: indexing and search. Indexing depends on the number of items in the corpus $|D|$. We use a model of size $\mathcal{M}$ to rewrite each passage (average length $|p|$) into propositions. The overall indexing cost is proportional to $O(D\mathcal{M}p^2)$, similar in magnitude to the cost for encoding the corpus in dense retrieval, differing only by a constant factor. Note that our experiments use a T5-large backbone (770M) for RICHES much smaller than T5-xxl (11B) used in the dense baselines.

Now let's look at the search operation. At each auto-regressive step, besides standard decoding, the only additional operation is computing FM-index constraints, which consumes CPU resources. However, while the index is efficient, communication between the index on the host and the Transformer model on the GPU/TPU adds latency to the decoding step. In contrast, RAG systems retrieve documents from index using nearest neighbor scan in a single go. But even there, the documents need to encoded as input to the language model.

Answer the ’question’ only based on the given ’passage’. If the ’passage’ lacks context or is not relevant, say ’Cannot answer’ else say generate a short answer. Do not answer the query from outside the scope of the passage. question: what brings in more money nba or nfl? passage: NFL revenues are well over $10 billion per season. NBA amasses about $6 billion annually. answer: NFL question: when did they put warnings on cigarette packs passage: Tobacco packaging 1978’s warning was not removed, so now every cigarette pack contains both warnings (one on each lateral). answer: Cannot Answer question: when was the french national anthem adopted? passage: La Marseillaise became the national anthem of France. La Marseillaise was adopted by France in 1795. answer: 1795 question: question passage: passage answer:

Table 12: Few-shot prompt for extracting answer from propositions

You are given a multi-hop ‘question‘. Decompose it into simple single-hop query, passage. And finally write the overall answer.

question: In what country was Lost Gravity manufactured?
query: Who manufactured The Lost Gravity (roller coaster)?
passage: Lost Gravity is a steel roller coaster at Walibi Holland manufactured by Mack Rides.
query: Mack Rides is from which country?
passage: Mack Rides is based in Germany.
answer: Germany

question: Do James Cameron and Christopher Nolan share their profession?
query: What is the profession of James Cameron?
passage: James Cameron is a Director.
query: What is the profession of Christopher Nolan?
passage: Christopher Nolan is a Director.
answer: Yes

question: The actor that stars as Joe Proctor on the series "Power" also played a character on "Entourage" that has what last name?
query: Who is the actor that stars as Joe Proctor on the series "Power"?
passage: Joe Proctor on the series "Power" was potrayed by Jerry Ferrara.
query: Jerry Ferrara played a character on Entourage named what?
passage: Jerry Ferrara played the character of Assante on Entourage.
answer: Assante

question: <question>
<sub-query steps so far>

Table 13: Few-shot prompt for Iterative baseline

Query	Retrievals	Comment
Index failure		
how many episodes of touching evil are there	A total of 35 episodes were produced.	Proposition lacks context
who is the coach for the ottawa senators	D. J. Smith is the head coach of the Ottawa Senators.	Incorrect Proposition generated
Search failure		
what age do you need to be to buy a bb gun	18 years of age or older.	partial phrase decoded
how many seasons of the bastard executioner are there	The Bastard Executioner is an American historical fiction drama television series. The Bastard Executioner is an American historical fiction drama television series.	repeated retrieval
who plays gram on the young and the restless	The Young and the Restless is an American television soap opera. The Young and the Restless was first broadcast on March 26, 1973.	irrelevant

Table 14: Example losses in RICHES

Unconstrained Generation	Constrained Generation
Q: who was the actor that played ben stone on law and order	
Ben Stone was played by actor Jerry Orbach.	Ben Stone was played by Michael Moriarty.
Q: how many pieces in a terry's chocolate orange	
Terry's Chocolate Orange is made with 32 segments	Terry's Chocolate Orange is divided into 20 segments
Q: who sings the song only in my dreams	
The song "Only in My Dreams" is sung by the band Air Supply.	Only in My Dreams is the debut single by Debbie Gibson.

Table 15: Unconstrained vs Constrained generation. Examples where unconstrained LLM emits incorrect answer but constraining on the corpus helps RICHES override this pre-existing knowledge to obtain the correct answer