HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

Bernal Jiménez Gutiérrez
The Ohio State University
jimenezgutierrez.1@osu.edu

Yiheng Shu
The Ohio State University
shu.251@osu.edu

Yu Gu
The Ohio State University
gu.826@osu.edu

Michihiro Yasunaga
Stanford University
myasu@cs.stanford.edu

Yu Su
The Ohio State University
su.809@osu.edu

Abstract

In order to thrive in hostile and ever-changing natural environments, mammalian brains evolved to store large amounts of knowledge about the world and continually integrate new information while avoiding catastrophic forgetting. Despite the impressive accomplishments, large language models (LLMs), even with retrieval-augmented generation (RAG), still struggle to efficiently and effectively integrate a large amount of new experiences after pre-training. In this work, we introduce HippoRAG, a novel retrieval framework inspired by the hippocampal indexing theory of human long-term memory to enable deeper and more efficient knowledge integration over new experiences. HippoRAG synergistically orchestrates LLMs, knowledge graphs, and the Personalized PageRank algorithm to mimic the different roles of neocortex and hippocampus in human memory. We compare HippoRAG with existing RAG methods on multi-hop question answering and show that our method outperforms the state-of-the-art methods remarkably, by up to 20%. Single-step retrieval with HippoRAG achieves comparable or better performance than iterative retrieval like IRCoT while being 10-30 times cheaper and 6-13 times faster, and integrating HippoRAG into IRCoT brings further substantial gains. Finally, we show that our method can tackle new types of scenarios that are out of reach of existing methods.[1]

1 Introduction

Millions of years of evolution have led mammalian brains to develop the crucial ability to store large amounts of world knowledge and continuously integrate new experiences without losing previous ones. This exceptional long-term memory system eventually allows us humans to keep vast stores of continuously updating knowledge that forms the basis of our reasoning and decision making [15].

Despite the progress of large language models (LLMs) in recent years, such a continuously updating long-term memory is still conspicuously absent from current AI systems. Due in part to its ease of use and the limitations of other techniques such as model editing [35], retrieval-augmented generation (RAG) has become the *de facto* solution for long-term memory in LLMs, allowing users to present new knowledge to a static model [28, 33, 50].

However, current RAG methods are still unable to help LLMs perform tasks that require integrating new knowledge across passage boundaries since each new passage is encoded in isolation. Many

[1] Code and data are available at `https://github.com/OSU-NLP-Group/HippoRAG`.

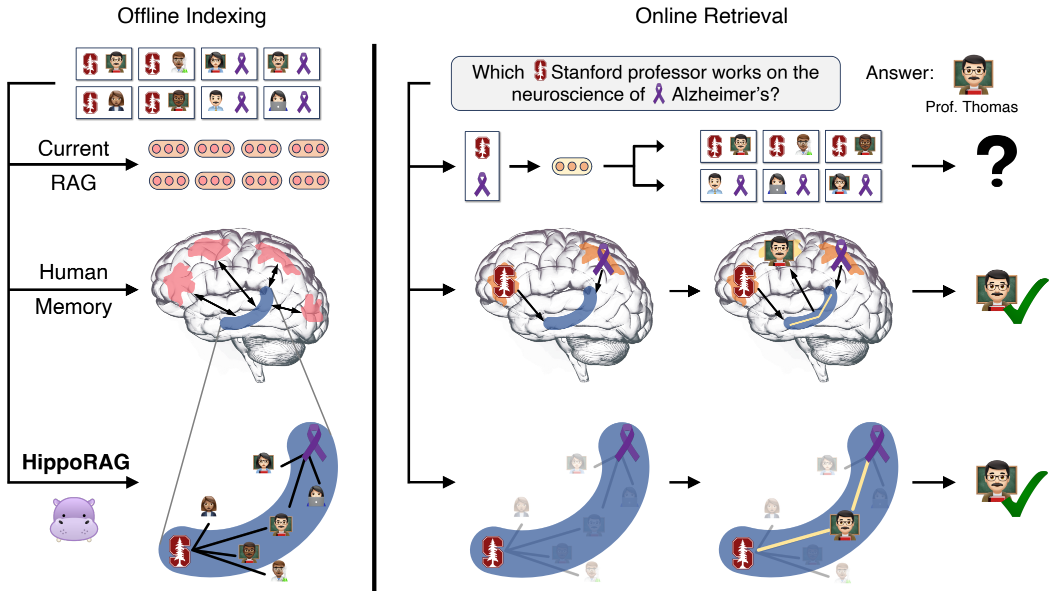


Figure 1: **Knowledge Integration & RAG.** Tasks that require knowledge integration are particularly challenging for current RAG systems. In the above example, we want to find a *Stanford* professor that does *Alzheimer's* research from a pool of passages describing potentially thousands *Stanford* professors and *Alzheimer's* researchers. Since current methods encode passages in isolation, they would struggle to identify *Prof. Thomas* unless a passage mentions both characteristics at once. In contrast, most people familiar with this professor would remember him quickly due to our brain's associative memory capabilities, thought to be driven by the index structure depicted in the C-shaped hippocampus above (in blue). Inspired by this mechanism, **HippoRAG** allows LLMs to build and leverage a similar graph of associations to tackle knowledge integration tasks.

important real-world tasks, such as scientific literature review, legal case briefing, and medical diagnosis, require knowledge integration across passages or documents. Although less complex, standard multi-hop question answering (QA) also requires integrating information between passages in a retrieval corpus. In order to solve such tasks, current RAG systems resort to using multiple retrieval and LLM generation steps iteratively to join disparate passages [49, 61]. Nevertheless, even perfectly executed multi-step RAG is still oftentimes insufficient to accomplish many scenarios of knowledge integration, as we illustrate in what we call *path-finding* multi-hop questions in Figure 1.

In contrast, our brains are capable of solving challenging knowledge integration tasks like these with relative ease. The hippocampal memory indexing theory [58], a well-established theory of human long-term memory, offers one plausible explanation for this remarkable ability. Teyler and Discenna [58] propose that our powerful context-based, continually updating memory relies on interactions between the neocortex, which processes and stores actual memory representations, and the C-shaped hippocampus, which holds the *hippocampal index*, a set of interconnected indices which point to memory units on the neocortex and stores associations between them [15, 59].

In this work, we propose HippoRAG, a RAG framework that serves as a long-term memory for LLMs by mimicking this model of human memory. Our novel design first models the neocortex's ability to process perceptual input by using an LLM to transform a corpus into a schemaless knowledge graph (KG) as our artificial hippocampal index. Given a new query, HippoRAG identifies the key concepts in the query and runs the Personalized PageRank (PPR) algorithm [23] on the KG, using the query concepts as the seeds, to integrate information across passages for retrieval. PPR enables HippoRAG to explore KG paths and identify relevant subgraphs, essentially performing multi-hop reasoning in a single retrieval step.

This capacity for *single-step multi-hop* retrieval yields strong performance improvements of around 3 and 20 points over current RAG methods [8, 27, 41, 53, 54] on two popular multi-hop QA benchmarks, MuSiQue [60] and 2WikiMultiHopQA [25]. Additionally, HippoRAG's online retrieval process is 10 to 30 times cheaper and 6 to 13 times faster than current iterative retrieval methods like IRCoT [61], while still achieving comparable performance. Furthermore, our approach can be combined with IRCoT to provide complementary gains of up to 4% and 20% on the same datasets and even obtain improvements on HotpotQA, a less challenging multi-hop QA dataset. Finally, we provide a case study illustrating the limitations of current methods as well as our method's potential on the previously discussed *path-finding* multi-hop QA setting.

2 HippoRAG

In this section, we first give a brief overview of the hippocampal memory indexing theory, followed by how HippoRAG's indexing and retrieval design was inspired by this theory, and finally offer a more detailed account of our methodology.

2.1 The Hippocampal Memory Indexing Theory

The hippocampal memory indexing theory [58] is a well-established theory that provides a functional description of the components and circuitry involved in human long-term memory. In this theory, Teyler and Discenna [58] propose that human long-term memory is composed of three components that work together to accomplish two main objectives: *pattern separation*, which ensures that the representations of distinct perceptual experiences are unique, and *pattern completion*, which enables the retrieval of complete memories from partial stimuli [15, 59].

The theory suggests that pattern separation is primarily accomplished in the memory encoding process, which starts with the **neocortex** receiving and processing perceptual stimuli into more easily manipulatable, likely higher-level, features, which are then routed through the **parahippocampal regions** (PHR) to be indexed by the hippocampus. When they reach the **hippocampus**, salient signals are included in the hippocampal index and associated with each other.

After the memory encoding process is completed, pattern completion drives the memory retrieval process whenever the hippocampus receives partial perceptual signals from the PHR pipeline. The hippocampus then leverages its context-dependent memory system, thought to be implemented through a densely connected network of neurons in the CA3 sub-region [59], to identify complete and relevant memories within the hippocampal index and route them back through the PHR for simulation in the neocortex. Thus, this complex process allows for new information to be integrated by changing only the hippocampal index instead of updating neocortical representations.

2.2 Overview

Our proposed approach, HippoRAG, is closely inspired by the process described above. As shown in Figure 2, each component of our method corresponds to one of the three components of human long-term memory. A detailed example of the HippoRAG process can be found in Appendix A.

Offline Indexing. Our offline indexing phase, analogous to memory encoding, starts by leveraging a strong instruction-tuned **LLM**, our artificial neocortex, to extract knowledge graph (KG) triples. The KG is schemaless and this process is known as open information extraction (OpenIE) [3, 4, 45, 79]. This process extracts salient signals from passages in a retrieval corpus as discrete noun phrases rather than dense vector representations, allowing for more fine-grained pattern separation. It is therefore natural to define our artificial hippocampal index as this open **KG**, which is built on the whole retrieval corpus passage-by-passage. Finally, to connect both components as is done by the parahippocampal regions, we use off-the-shelf dense encoders fine-tuned for retrieval (**retrieval encoders**). These retrieval encoders provide additional edges between similar but not identical noun phrases within this KG to aid in downstream pattern completion.

Online Retrieval. These same three components are then leveraged to perform online retrieval by mirroring the human brain's memory retrieval process. Just as the hippocampus receives input processed through the neocortex and PHR, our LLM-based neocortex extracts a set of salient named entities from a query which we call *query named entities*. These named entities are then linked to nodes in our KG based on the similarity determined by retrieval encoders; we refer to these selected nodes as *query nodes*. Once the query nodes are chosen, they become the partial cues from which our synthetic hippocampus performs pattern completion. In the hippocampus, neural pathways between elements of the hippocampal index enable relevant neighborhoods to become activated and recalled upstream. To imitate this efficient graph search process, we leverage the Personalized PageRank

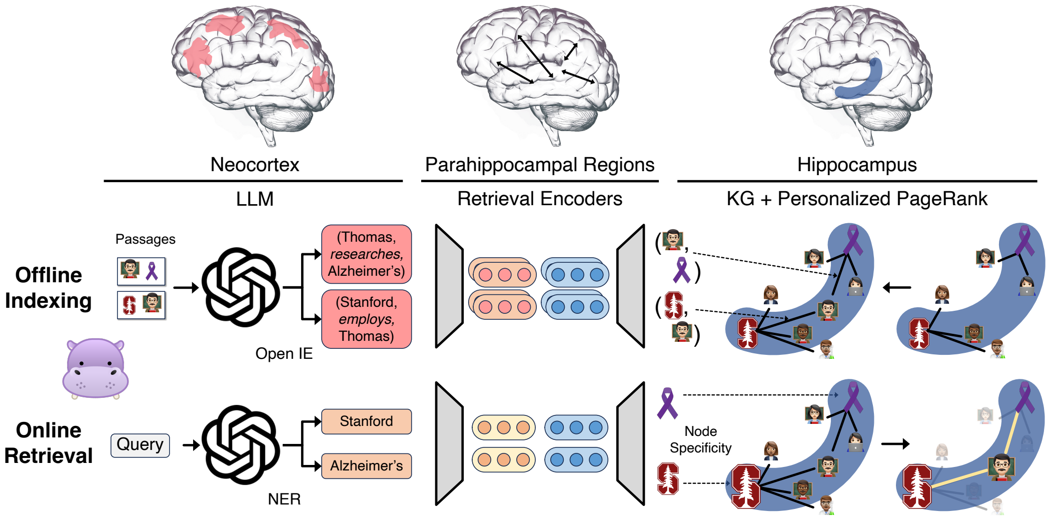


Figure 2: **Detailed HippoRAG Methodology.** We model the three components of human long-term memory to mimic its pattern separation and completion functions. For offline indexing **(Middle)**, we use an LLM to process passages into open KG triples, which are then added to our artificial hippocampal index, while our synthetic parahippocampal regions (PHR) detect synonymy. In the example above, triples involving Professor Thomas are extracted and integrated into the KG. For online retrieval **(Bottom)**, our LLM neocortex extracts named entities from a query while our parahippocampal retrieval encoders link them to our hippocampal index. We then leverage the Personalized PageRank algorithm to enable context-based retrieval and extract Professor Thomas.[4]

(PPR) algorithm [23], a version of PageRank that distributes probability across a graph only through a set of user-defined source nodes. This constraint allows us to bias the PPR output only towards the set of query nodes, just as the hippocampus extracts associated signals from specific partial cues.[2] Finally, as is done when the hippocampal signal is sent upstream, we aggregate the output PPR node probability over the previously indexed passages and use that to rank them for retrieval.

2.3 Detailed Methodology

Offline Indexing. Our indexing process involves processing a set of passages P using an instruction-tuned LLM L and a retrieval encoder M. As seen in Figure 2 we first use L to extract a set of noun phrase nodes N and relation edges E from each passage in P via OpenIE. This process is done via 1-shot prompting of the LLM with the prompts shown in Appendix I. Specifically, we first extract a set of named entities from each passage. We then add the named entities to the OpenIE prompt to extract the final triples, which also contain concepts (noun phrases) beyond named entities. We find that this two-step prompt configuration leads to an appropriate balance between generality and bias towards named entities. Finally, we use M to add the extra set of *synonymy* relations E' discussed above when the cosine similarity between two entity representations in N is above a threshold τ. As stated above, this introduces more edges to our hippocampal index and allows for more effective pattern completion. This indexing process defines a $|N| \times |P|$ matrix $\mathbf{P}$, which contains the number of times each noun phrase in the KG appears in each original passage.

Online Retrieval. During the retrieval process, we prompt L using a 1-shot prompt to extract a set of named entities from a query q, our previously defined query named entities $C_q = \{c_1, ..., c_n\}$ (*Stanford* and *Alzheimer's* in our Figure 2 example). These named entities C_q from the query are then encoded by the same retrieval encoder M. Then, the previously defined query nodes are chosen as the set of nodes in N with the highest cosine similarity to the query named entities C_q. More formally, query nodes are defined as $R_q = \{r_1, ..., r_n\}$ such that $r_i = e_k$ where $k = \arg\max_j \mathit{cosine\_similarity}(M(c_i), M(e_j))$, represented as the *Stanford* logo and the *Alzheimer's* purple ribbon symbol in Figure 2

[2] Intriguingly, some work in cognitive science has also found a correlation between human word recall and the output of the PageRank algorithm [19].

After the query nodes R_q are found, we run the PPR algorithm over the hippocampal index, i.e., a KG with $|N|$ nodes and $|E| + |E'|$ edges (triple-based and synonymy-based), using a personalized probability distribution $\vec{n}$ defined over N, in which each query node has equal probability and all other nodes have a probability of zero. This allows probability mass to be distributed to nodes that are primarily in the (joint) neighborhood of the query nodes, such as *Professor Thomas*, and contribute to eventual retrieval. After running the PPR algorithm, we obtain an updated probability distribution $\vec{n'}$ over N. Finally, in order to obtain passage scores, we multiply $\vec{n'}$ with the previously defined $\mathbf{P}$ matrix to obtain $\vec{p}$, a ranking score for each passage, which we use for retrieval.

Node Specificity. We introduce node specificity as a neurobiologically plausible way to further improve retrieval. It is well known that global signals for word importance, like inverse document frequency (IDF), can improve information retrieval. However, in order for our brain to leverage IDF for retrieval, the number of total "passages" encoded would need to be aggregated with all node activations before memory retrieval is complete. While simple for normal computers, this process would require activating connections between an aggregator neuron and all nodes in the hippocampal index every time retrieval occurs, likely introducing prohibitive computational overhead. Given these constraints, we propose *node specificity* as an alternative IDF signal which requires only local signals and is thus more neurobiologically plausible. We define the node specificity of node i as $s_i = |P_i|^{-1}$, where P_i is the set of passages in P from which node i was extracted, information that is already available at each node. Node specificity is used in retrieval by multiplying each query node probability $\vec{n}$ with s_i before PPR; this allows us to modulate each of their neighborhood's probability as well as their own. We illustrate node specificity in Figure 2 through relative symbol size: the *Stanford* logo grows larger than the *Alzheimer's* symbol since it appears in fewer documents.

3 Experimental Setup

3.1 Datasets

We evaluate our method's retrieval capabilities primarily on two challenging multi-hop QA benchmarks, **MuSiQue** (answerable) [60] and **2WikiMultiHopQA** [25]. For completeness, we also include the **HotpotQA** [70] dataset even though it has been found to be a much weaker test for multi-hop reasoning due to many spurious signals [60], as we also show in Appendix B. To limit the experimental cost, we extract 1,000 questions from each validation set as done in previous work [48, 61]. In order to create a more realistic retrieval setting, we follow IRCoT [61] and collect all candidate passages (including supporting and distractor passages) from our selected questions and form a retrieval corpus for each dataset. The details of these datasets are shown in Table 1.

Table 1: Retrieval corpora and extracted KG statistics for each of our $1{,}000$ question dev sets.

	MuSiQue	**2Wiki**	**HotpotQA**
# of Passages (P)	$11,656$	$6,119$	$9,221$
# of Unique Nodes (N)	$91,729$	$42,694$	$82,157$
# of Unique Edges (E)	$21,714$	$7,867$	$17,523$
# of Unique Triples	$107,448$	$50,671$	$98,709$
# of Contriever Synonym Edges (E')	$145,990$	$146,020$	$159,112$
# of ColBERTv2 Synonym Edges (E')	$191,636$	$82,526$	$171,856$

3.2 Baselines

We compare against several strong and widely used retrieval methods: **BM25** [52], **Contriever** [27], **GTR** [41] and **ColBERTv2** [53]. Additionally, we compare against two recent LLM-augmented baselines: **Propositionizer** [8], which rewrites passages into propositions, and **RAPTOR** [54], which constructs summary nodes to ease retrieval from long documents. In addition to the single-step retrieval methods above, we also include the multi-step retrieval method **IRCoT** [61] as a baseline.

[4] Many details around the hippocampal memory indexing theory are omitted from this study for simplicity. We encourage interested reader to follow the references in §2.1 for more.

Table 2: **Single-step retrieval performance.** HippoRAG outperforms all baselines on MuSiQue and 2WikiMultiHopQA and achieves comparable performance on the less challenging HotpotQA dataset.

	MuSiQue		**2Wiki**		**HotpotQA**		**Average**	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
BM25 [52]	32.3	41.2	51.8	61.9	55.4	72.2	46.5	58.4
Contriever [27]	34.8	46.6	46.6	57.5	57.2	75.5	46.2	59.9
GTR [41]	37.4	49.1	60.2	67.9	59.4	73.3	52.3	63.4
ColBERTv2 [53]	37.9	49.2	59.2	68.2	**64.7**	**79.3**	53.9	65.6
RAPTOR [54]	35.7	45.3	46.3	53.8	58.1	71.2	46.7	56.8
Proposition [8]	37.6	49.3	56.4	63.1	58.7	71.1	50.9	61.2
HippoRAG (Contriever)	**41.0**	**52.1**	**71.5**	**89.5**	59.0	76.2	57.2	72.6
HippoRAG (ColBERTv2)	40.9	51.9	70.7	89.1	60.5	77.7	**57.4**	**72.9**

Table 3: **Multi-step retrieval performance.** Combining HippoRAG with standard multi-step retrieval methods like IRCoT results in significant improvements on all three datasets.

	MuSiQue		**2Wiki**		**HotpotQA**		**Average**	
	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
IRCoT + BM25 (Default)	34.2	44.7	61.2	75.6	65.6	79.0	53.7	66.4
IRCoT + Contriever	39.1	52.2	51.6	63.8	65.9	81.6	52.2	65.9
IRCoT + ColBERTv2	41.7	53.7	64.1	74.4	**67.9**	82.0	57.9	70.0
IRCoT + HippoRAG (Contriever)	43.9	56.6	75.3	93.4	65.8	82.3	61.7	77.4
IRCoT + HippoRAG (ColBERTv2)	**45.3**	**57.6**	**75.8**	**93.9**	67.0	**83.0**	**62.7**	**78.2**

3.3 Metrics

We report retrieval and QA performance on the datasets above using recall@2 and recall@5 (R@2 and R@5 below) for retrieval and exact match (EM) and F1 scores for QA performance.

3.4 Implementation Details

By default, we use `GPT-3.5-turbo-1106` [42] with temperature of 0 as our LLM L and Contriever [27] or ColBERTv2 [53] as our retriever M. We use 100 examples from MuSiQue's training data to tune HippoRAG's two hyperparameters: the synonymy threshold τ at 0.8 and the PPR damping factor at 0.5, which determines the probability that PPR will restart a random walk from the query nodes instead of continuing to explore the graph. Generally, we find that HippoRAG's performance is rather robust to its hyperparameters. More implementation details can be found in Appendix H.

4 Results

We present our retrieval and QA experimental results below. Given that our method indirectly affects QA performance, we report QA results on our best-performing retrieval backbone ColBERTv2 [53]. However, we report retrieval results for several strong single-step and multi-step retrieval techniques.

Single-Step Retrieval Results. As seen in Table 2, HippoRAG outperforms all other methods, including recent LLM-augmented baselines such as Propositionizer and RAPTOR, on our main datasets, MuSiQue and 2WikiMultiHopQA, while achieving competitive performance on HotpotQA. We notice an impressive improvement of 11 and 20% for R@2 and R@5 on 2WikiMultiHopQA and around 3% on MuSiQue. This difference can be partially explained by 2WikiMultiHopQA's entity-centric design, which is particularly well-suited for HippoRAG. Our lower performance on HotpotQA is mainly due to its lower knowledge integration requirements, as explained in Appendix B, as well as a due to a concept-context tradeoff which we alleviate with an ensembling technique described in Appendix F.2.

Multi-Step Retrieval Results. For multi-step or iterative retrieval, our experiments in Table 3 demonstrate that IRCoT [61] and HippoRAG are complementary. Using HippoRAG as the re-

Table 4: **QA performance.** HippoRAG's QA improvements correlate with its retrieval improvements on single-step (rows 1-3) and multi-step retrieval (rows 4-5).

	MuSiQue		2Wiki		HotpotQA		Average	
Retriever	EM	F1	EM	F1	EM	F1	EM	F1
None	12.5	24.1	31.0	39.6	30.4	42.8	24.6	35.5
ColBERTv2	15.5	26.4	33.4	43.3	43.4	57.7	30.8	42.5
HippoRAG (ColBERTv2)	19.2	29.8	46.6	59.5	41.8	55.0	35.9	48.1
IRCoT (ColBERTv2)	19.1	30.5	35.4	45.1	45.5	58.4	33.3	44.7
IRCoT + HippoRAG (ColBERTv2)	**21.9**	**33.3**	**47.7**	**62.7**	**45.7**	**59.2**	**38.4**	**51.7**

triever for IRCoT continues to bring R@5 improvements of around 4% for MuSiQue, 18% for 2WikiMultiHopQA and an additional 1% on HotpotQA.

Question Answering Results. We report QA results for HippoRAG, the strongest retrieval baselines, ColBERTv2 and IRCoT, as well as IRCoT using HippoRAG as a retriever in Table 4. As expected, improved retrieval performance in both single and multi-step settings leads to strong overall improvements of up to 3%, 17% and 1% F1 scores on MuSiQue, 2WikiMultiHopQA and HotpotQA respectively using the same QA reader. Notably, single-step HippoRAG is on par or outperforms IRCoT while being 10-30 times cheaper and 6-13 times faster during online retrieval (Appendix G).

5 Discussions

5.1 What Makes HippoRAG Work?

OpenIE Alternatives. To determine if using GPT-3.5 is essential for building our KG, we replace it with an end-to-end OpenIE model REBEL [26] and an instruction-tuned LLM Llama-3 [1]. As shown in Table 5 row 2, building our KG using REBEL results in large performance drops, underscoring the importance of LLM flexibility. Specifically, GPT-3.5 produces twice as many triples as REBEL, indicating its bias against producing triples with general concepts and leaving many useful associations behind.

In terms of open-source LLMs, Table 5 (rows 3-4) shows that the performance of Llama-3 8B is comparable to GPT-3.5, although its 70B counterpart performs worse. This surprising behavior is due to this model's production of ill-formatted outputs that result in the loss of around 20% of the passages, compared to about 4% for the 8B model and less than 1% for GPT-3.5. The strong performance of Llama-3 8B is encouraging because that offers a cheaper alternative for indexing over large corpora. The statistics for these OpenIE alternatives can be found in Appendix C.

PPR Alternatives. As shown in Table 5 (rows 5-6), to examine how much of our results are due to the strength of PPR, we replace the PPR output with the query node probability $\vec{n}$ multiplied by node specificity values (row 5) and a version of this that also distributes a small amount of probability to the

Table 5: **Dissecting HippoRAG.** To understand what makes it work well, we replace its OpenIE module and PPR with plausible alternatives and ablate node specificity and synonymy-based edges.

		MuSiQue		2Wiki		HotpotQA		Average	
		R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
HippoRAG		**40.9**	**51.9**	70.7	**89.1**	**60.5**	**77.7**	**57.4**	**72.9**
OpenIE Alternatives	REBEL [26]	31.7	39.6	63.1	76.5	43.9	59.2	46.2	58.4
	Llama-3-8B-Instruct [1]	39.4	50.9	**71.1**	88.0	59.4	76.4	56.6	71.8
	Llama-3-70B-Instruct [1]	36.3	45.9	56.9	68.8	51.6	66.1	48.3	60.3
PPR Alternatives	R_q Nodes Only	37.1	41.0	59.1	61.4	55.9	66.2	50.7	56.2
	R_q Nodes & Neighbors	25.4	38.5	53.4	74.7	47.8	64.5	42.2	59.2
Ablations	w/o Node Specificity	37.6	50.2	70.1	88.8	56.3	73.7	54.7	70.9
	w/o Synonymy Edges	40.2	50.2	69.2	85.6	59.1	75.7	56.2	70.5

direct neighbors of each query node (row 6). First, we find that PPR is a much more effective method for including associations for retrieval on all three datasets compared to both simple baselines. It is interesting to note that adding the neighborhood of R_q nodes without PPR leads to worse performance than only using the query nodes themselves.

Ablations. As seen in Table 5 (rows 7-8), node specificity obtains considerable improvements on MuSiQue and HotpotQA and yields almost no change in 2WikiMultiHopQA. This is likely because 2WikiMultiHopQA relies on named entities with little differences in terms of term weighting. In contrast, synonymy edges have the largest effect on 2WikiMultiHopQA, suggesting that noisy entity standardization is useful when most relevant concepts are named entities, and improvements to synonymy detection could lead to stronger performance in other datasets.

5.2 HippoRAG's Advantage: Single-Step Multi-Hop Retrieval

A major advantage of HippoRAG over conventional RAG methods in multi-hop QA is its ability to *perform multi-hop retrieval in a single step*. We demonstrate this by measuring the percentage of queries where *all* the supporting passages are retrieved successfully, a feat that can only be accomplished through successful multi-hop reasoning. Table 8 in Appendix D shows that the gap between our method and ColBERTv2, using the top-5 passages, increases even more from 3% to 6% on MuSiQue and from 20% to 38% on 2WikiMultiHopQA, suggesting that large improvements come from obtaining all supporting documents rather than achieving partially retrieval on more questions.

We further illustrate HippoRAG's unique *single-step multi-hop retrieval* ability through the first example in Table 6. In this example, even though *Alhandra* was not mentioned in *Vila de Xira's* passage, HippoRAG can directly leverage Vila de Xira's connection to Alhandra as his place of birth to determine its importance, something that standard RAG methods would be unable to do directly. Additionally, even though IRCoT can also solve this multi-hop retrieval problem, as shown in Appendix G, it is 10-30 times more expensive and 6-13 times slower than ours in terms of online retrieval, arguably the most important factor when it comes to serving end users.

Table 6: **Multi-hop question types.** We show example results for different approaches on path-finding vs. path-following multi-hop questions.

	Question	HippoRAG	ColBERTv2	IRCoT
Path-Following	In which district was **Alhandra** born?	**1. Alhandra** **2. Vila de Xira** **3.** Portugal	**1. Alhandra** **2.** Dimuthu Abayakoon **3.** Ja'ar	**1. Alhandra** **2. Vila de Xira** **3.** Póvoa de Santa Iria
Path-Finding	Which **Stanford** professor works on the neuroscience of **Alzheimer's**?	**1. Thomas Südhof** **2.** Karl Deisseroth **3.** Robert Sapolsky	**1.** Brian Knutson **2.** Eric Knudsen **3.** Lisa Giocomo	**1.** Brian Knutson **2.** Eric Knudsen **3.** Lisa Giocomo

5.3 HippoRAG's Potential: Path-Finding Multi-Hop Retrieval

The second example in Table 6, also present in Figure 1, shows a type of questions that is trivial for informed humans but out of reach for current retrievers without further training. This type of questions, which we call *path-finding* multi-hop questions, requires identifying one path between a set of entities when many paths exist to explore instead of *following* a specific path, as in standard multi-hop questions.[5]

More specifically, a simple iterative process can retrieve the appropriate passages for the first question by following the one path set by *Alhandra's* one place of birth, as seen by IRCoT's perfect performance. However, an iterative process would struggle to answer the second question given the many possible paths to explore—either through professors at *Stanford University* or professors working on the neuroscience of *Alzheimer's*. It is only by associating disparate information about Thomas Südhof that someone who knows about this professor would be able to answer this question

[5] Path-finding questions require knowledge integration when search entities like *Stanford* and *Alzheimer's* do not happen to appear together in a passage, a condition which is often satisfied for new information.

easily. As seen in Table 6, both ColBERTv2 and IRCoT fail to extract the necessary passages since they cannot access these associations. On the other hand, HippoRAG leverages its web of associations in its hippocampal index and graph search algorithm to determine that Professor Thomas is relevant to this query and retrieves his passages appropriately. More examples of these path-finding multi-hop questions can be found in our case study in Appendix E.

6 Related Work

Parametric Long-Term Memory. It is well-accepted, even among skeptical researchers, that the parameters of modern LLMs encode a remarkable amount of world knowledge [2, 10, 17, 21, 24, 31, 47, 62], which can be leveraged by an LLM in flexible and robust ways [64, 65, 74]. Nevertheless, our ability to update this vast knowledge store, an essential part of any long-term memory system, is still surprisingly limited. Although many techniques to update LLMs exist, such as standard fine-tuning, model unlearning and model editing [12, 37, 38, 39, 40, 76], it is clear that no methodology has emerged as a robust solution for continual learning in LLMs [20, 35, 78].

RAG as Long-Term Memory. On the other hand, using RAG methods as a long-term memory system offers a simple way to update knowledge over time [28, 33, 50, 56]. More sophisticated RAG methods, which perform multiple steps of retrieval and generation from an LLM, are even able to integrate information across new or updated knowledge elements[30, 49, 55, 61, 69, 71, 73], another crucial aspect of long-term memory systems. As discussed above, however, this type of online information integration is unable to solve the more complex knowledge integration tasks that we illustrate with our *path-finding* multi-hop QA examples.

Some other methods, such as RAPTOR [54], MemWalker [7] and GraphRAG [14], integrate information during the offline indexing phase similarly to HippoRAG and might be able to handle these more complex tasks. However, these methods integrate information by summarizing knowledge elements, which means that the summarization process must be repeated any time new data is added. In contrast, HippoRAG can continuously integrate new knowledge by simply adding edges to its KG.

Long Context as Long-Term Memory. Context lengths for both open and closed source LLMs have increased dramatically in the past year [9, 13, 16, 46, 51]. This scaling trend seems to indicate that future LLMs could perform long-term memory storage within massive context windows. However, the viability of this future remains largely uncertain given the many engineering hurdles involved and the apparent limitations of long-context LLMs, even within current context lengths [32, 34, 77].

LLMs & KGs. Combining the strengths of language models and knowledge graphs has been an active research direction for many years, both for augmenting LLMs with a KG in different ways [36, 63, 66] or augmenting KGs by either distilling knowledge from an LLM's parametric knowledge [5, 67] or using them to parse text directly [6, 22, 75]. In an exceptionally comprehensive survey, Pan et al. [43] present a roadmap for this research direction and highlight the importance of work which *synergizes* these two important technologies [29, 57, 72, 80]. Like these works, HippoRAG is a strong and principled example of the synergy we must strike between these two technologies, combining the power of LLMs for knowledge graph construction with the strengths of structured knowledge and graph search for improved augmentation of an LLM's capacities.

7 Conclusions & Limitations

Our proposed neurobiologically principled methodology, although simple, already shows promise for overcoming the inherent limitations of standard RAG systems while retaining their advantages over parametric memory. HippoRAG's knowledge integration capabilities, demonstrated by its strong results on *path-following* multi-hop QA and promise on *path-finding* multi-hop QA, as well as its dramatic efficiency improvements and continuously updating nature, makes it a powerful middle-ground framework between standard RAG methods and parametric memory and offers a compelling solution for long-term memory in LLMs.

Nevertheless, several limitations can be addressed in future work to enable HippoRAG to achieve this goal better. First, we note that all components of HippoRAG are currently used off-the-shelf without any extra training. There is therefore much room to improve our method's practical viability by performing specific component fine-tuning. This is evident in the error analysis discussed in Appendix

F that shows most errors made by our system are due to NER and OpenIE, which could benefit from direct fine-tuning. Given that the rest of the errors are graph search errors, also in Appendix F, we note that several avenues for improvements over simple PPR exist, such as allowing relations to guide graph traversal directly. Finally, and perhaps most importantly, HippoRAG's scalability still calls for further validation. Although we show that Llama-3 could obtain similar performance to closed-source models and thus reduce costs considerably, we are yet to empirically prove the efficiency and efficacy of our synthetic hippocampal index as its size grows way beyond current benchmarks.

Acknowledgments

The authors would like to thank colleagues from the OSU NLP group and Percy Liang for their thoughtful comments. This research was supported in part by NSF OAC 2112606, NIH R01LM014199, ARL W911NF2220144, and Cisco. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

References

[1] AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

[2] B. AlKhamissi, M. Li, A. Celikyilmaz, M. T. Diab, and M. Ghazvininejad. A review on language models as knowledge bases. *ArXiv*, abs/2204.06031, 2022. URL `https://arxiv.org/abs/2204.06031`.

[3] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In C. Zong and M. Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL `https://aclanthology.org/P15-1034`.

[4] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[5] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1470. URL `https://aclanthology.org/P19-1470`.

[6] B. Chen and A. L. Bertozzi. AutoKG: Efficient Automated Knowledge Graph Generation for Language Models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 3117–3126, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society. doi: 10.1109/BigData59044.2023.10386454. URL `https://doi.ieeecomputersociety.org/10.1109/BigData59044.2023.10386454`.

[7] H. Chen, R. Pasunuru, J. Weston, and A. Celikyilmaz. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. *CoRR*, abs/2310.05029, 2023. doi: 10.48550/ARXIV.2310.05029. URL `https://doi.org/10.48550/arXiv.2310.05029`.

[8] T. Chen, H. Wang, S. Chen, W. Yu, K. Ma, X. Zhao, H. Zhang, and D. Yu. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*, 2023. URL `https://arxiv.org/abs/2312.06648`.

[9] Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, S. Han, and J. Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv:2309.12307*, 2023.

[10] Y. Chen, P. Cao, Y. Chen, K. Liu, and J. Zhao. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17817–17825, Mar. 2024. doi: 10.1609/aaai.v38i16.29735. URL `https://ojs.aaai.org/index.php/AAAI/article/view/29735`.

[11] G. Csárdi and T. Nepusz. The igraph software package for complex network research. 2006. URL `https://igraph.org/`.

[12] N. De Cao, W. Aziz, and I. Titov. Editing factual knowledge in language models. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL `https://aclanthology.org/2021.emnlp-main.522`.

[13] Y. Ding, L. L. Zhang, C. Zhang, Y. Xu, N. Shang, J. Xu, F. Yang, and M. Yang. Longrope: Extending llm context window beyond 2 million tokens. *ArXiv*, abs/2402.13753, 2024. URL `https://api.semanticscholar.org/CorpusID:267770308`.

[14] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson. From local to global: A graph rag approach to query-focused summarization. 2024. URL `https://arxiv.org/abs/2404.16130`.

[15] H. Eichenbaum. A cortical–hippocampal system for declarative memory. *Nature Reviews Neuroscience*, 1:41–50, 2000. URL `https://www.nature.com/articles/35036213`.

[16] Y. Fu, R. Panda, X. Niu, X. Yue, H. Hajishirzi, Y. Kim, and H. Peng. Data engineering for scaling language models to 128k context, 2024.

[17] M. Geva, J. Bastings, K. Filippova, and A. Globerson. Dissecting recall of factual associations in auto-regressive language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.751. URL `https://doi.org/10.18653/v1/2023.emnlp-main.751`.

[18] C. Gormley and Z. J. Tong. Elasticsearch: The definitive guide. 2015. URL `https://www.elastic.co/guide/en/elasticsearch/guide/master/index.html`.

[19] T. L. Griffiths, M. Steyvers, and A. J. Firl. Google and the mind. *Psychological Science*, 18: 1069 – 1076, 2007. URL `https://cocosci.princeton.edu/tom/papers/google.pdf`.

[20] J.-C. Gu, H.-X. Xu, J.-Y. Ma, P. Lu, Z.-H. Ling, K.-W. Chang, and N. Peng. Model Editing Can Hurt General Abilities of Large Language Models, 2024.

[21] W. Gurnee and M. Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=jE8xbmvFin`.

[22] J. Han, N. Collier, W. Buntine, and E. Shareghi. PiVe: Prompting with Iterative Verification Improving Graph-based Generative Capability of LLMs, 2023.

[23] T. H. Haveliwala. Topic-sensitive pagerank. In D. Lassner, D. D. Roure, and A. Iyengar, editors, *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA*, pages 517–526. ACM, 2002. doi: 10.1145/511446.511513. URL `https://dl.acm.org/doi/10.1145/511446.511513`.

[24] Q. He, Y. Wang, and W. Wang. Can language models act as knowledge bases at scale?, 2024.

[25] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In D. Scott, N. Bel, and C. Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL `https://aclanthology.org/2020.coling-main.580`.

[26] P.-L. Huguet Cabot and R. Navigli. REBEL: Relation extraction by end-to-end language generation. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.204. URL `https://aclanthology.org/2021.findings-emnlp.204`.

[27] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning, 2021. URL `https://arxiv.org/abs/2112.09118`.

[28] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. A. Yu, A. Joulin, S. Riedel, and E. Grave. Few-shot learning with retrieval augmented language models. *ArXiv*, abs/2208.03299, 2022. URL `https://arxiv.org/abs/2208.03299`.

[29] J. Jiang, K. Zhou, Z. Dong, K. Ye, X. Zhao, and J.-R. Wen. StructGPT: A general framework for large language model to reason over structured data. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.574. URL `https://aclanthology.org/2023.emnlp-main.574`.

[30] Z. Jiang, F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig. Active retrieval augmented generation. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL `https://aclanthology.org/2023.emnlp-main.495`.

[31] S. Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 2024. URL `https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1111/nyas.15125`.

[32] M. Levy, A. Jacoby, and Y. Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024.

[33] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. URL `https://dl.acm.org/doi/abs/10.5555/3495724.3496517`.

[34] T. Li, G. Zhang, Q. D. Do, X. Yue, and W. Chen. Long-context LLMs Struggle with Long In-context Learning, 2024.

[35] Z. Li, N. Zhang, Y. Yao, M. Wang, X. Chen, and H. Chen. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=fNktD3ib16`.

[36] L. LUO, Y.-F. Li, R. Haf, and S. Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ZGNWW7xZ6Q`.

[37] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*, 2022.

[38] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. *ArXiv*, abs/2110.11309, 2021.

[39] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. *ArXiv*, abs/2206.06520, 2022.

[40] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[41] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, and Y. Yang. Large dual encoders are generalizable retrievers. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.669. URL `https://aclanthology.org/2022.emnlp-main.669`.

[42] OpenAI. GPT-3.5 Turbo, 2024. URL `https://platform.openai.com/docs/models/gpt-3-5-turbo`.

[43] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2024. doi: 10.1109/TKDE.2024.3352100.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL `https://dl.acm.org/doi/10.5555/3454287.3455008`.

[45] K. Pei, I. Jindal, K. C.-C. Chang, C. Zhai, and Y. Li. When to use what: An in-depth comparative empirical analysis of OpenIE systems for downstream applications. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–949, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.53. URL `https://aclanthology.org/2023.acl-long.53`.

[46] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models, 2023.

[47] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. Language models as knowledge bases? In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL `https://aclanthology.org/D19-1250`.

[48] O. Press, M. Zhang, S. Min, L. Schmidt, N. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL `https://aclanthology.org/2023.findings-emnlp.378`.

[49] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models, 2023. URL `https://openreview.net/forum?id=PUwbwZJz9dO`.

[50] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tacl_a_00605. URL `https://aclanthology.org/2023.tacl-1.75`.

[51] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. URL `https://arxiv.org/abs/2403.05530`.

[52] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings*

of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum), pages 232–241. ACM/Springer, 1994. doi: 10.1007/978-1-4471-2099-5_24. URL `https://link.springer.com/chapter/10.1007/978-1-4471-2099-5_24`.

[53] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL `https://aclanthology.org/2022.naacl-main.272`.

[54] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning. RAPTOR: recursive abstractive processing for tree-organized retrieval. *CoRR*, abs/2401.18059, 2024. doi: 10.48550/ARXIV.2401.18059. URL `https://arxiv.org/abs/2401.18059`.

[55] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.620. URL `https://aclanthology.org/2023.findings-emnlp.620`.

[56] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W. tau Yih. Replug: Retrieval-augmented black-box language models. *ArXiv*, abs/2301.12652, 2023. URL `https://api.semanticscholar.org/CorpusID:256389797`.

[57] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. Ni, H.-Y. Shum, and J. Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=nnVO1PvbTv`.

[58] T. J. Teyler and P. Discenna. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100 2:147–54, 1986. URL `https://pubmed.ncbi.nlm.nih.gov/3008780/`.

[59] T. J. Teyler and J. W. Rudy. The hippocampal indexing theory and episodic memory: Updating the index. *Hippocampus*, 17, 2007. URL `https://pubmed.ncbi.nlm.nih.gov/17696170/`.

[60] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL `https://aclanthology.org/2022.tacl-1.31/`.

[61] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL `https://aclanthology.org/2023.acl-long.557`.

[62] C. Wang, X. Liu, Y. Yue, X. Tang, T. Zhang, C. Jiayang, Y. Yao, W. Gao, X. Hu, Z. Qi, Y. Wang, L. Yang, J. Wang, X. Xie, Z. Zhang, and Y. Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity, 2023.

[63] J. Wang, Q. Sun, N. Chen, X. Li, and M. Gao. Boosting language models reasoning with chain-of-knowledge prompting, 2023.

[64] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

[65] J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=_VjQlMeSB_J`.

[66] Y. Wen, Z. Wang, and J. Sun. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*, 2023.

[67] P. West, C. Bhagavatula, J. Hessel, J. Hwang, L. Jiang, R. Le Bras, X. Lu, S. Welleck, and Y. Choi. Symbolic knowledge distillation: from general language models to commonsense models. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.341. URL `https://aclanthology.org/2022.naacl-main.341`.

[68] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. URL `https://arxiv.org/abs/1910.03771`.

[69] W. Xiong, X. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, S. Yih, S. Riedel, D. Kiela, and B. Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=EMHoBG0avc1`.

[70] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL `https://aclanthology.org/D18-1259/`.

[71] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[72] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. Liang, and J. Leskovec. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://arxiv.org/abs/2210.09338`.

[73] O. Yoran, T. Wolfson, B. Bogin, U. Katz, D. Deutch, and J. Berant. Answering questions by meta-reasoning over multiple chains of thought. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://openreview.net/forum?id=ebSOK1nV2r`.

[74] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=fB0hRu9GZUS`.

[75] K. Zhang, B. Jimenez Gutierrez, and Y. Su. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.50. URL `https://aclanthology.org/2023.findings-acl.50`.

[76] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.

[77] X. Zhang, Y. Chen, S. Hu, Z. Xu, J. Chen, M. K. Hao, X. Han, Z. L. Thai, S. Wang, Z. Liu, and M. Sun. ∞bench: Extending long context evaluation beyond 100k tokens, 2024.

[78] Z. Zhong, Z. Wu, C. D. Manning, C. Potts, and D. Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://aclanthology.org/2023.emnlp-main.971.pdf`.

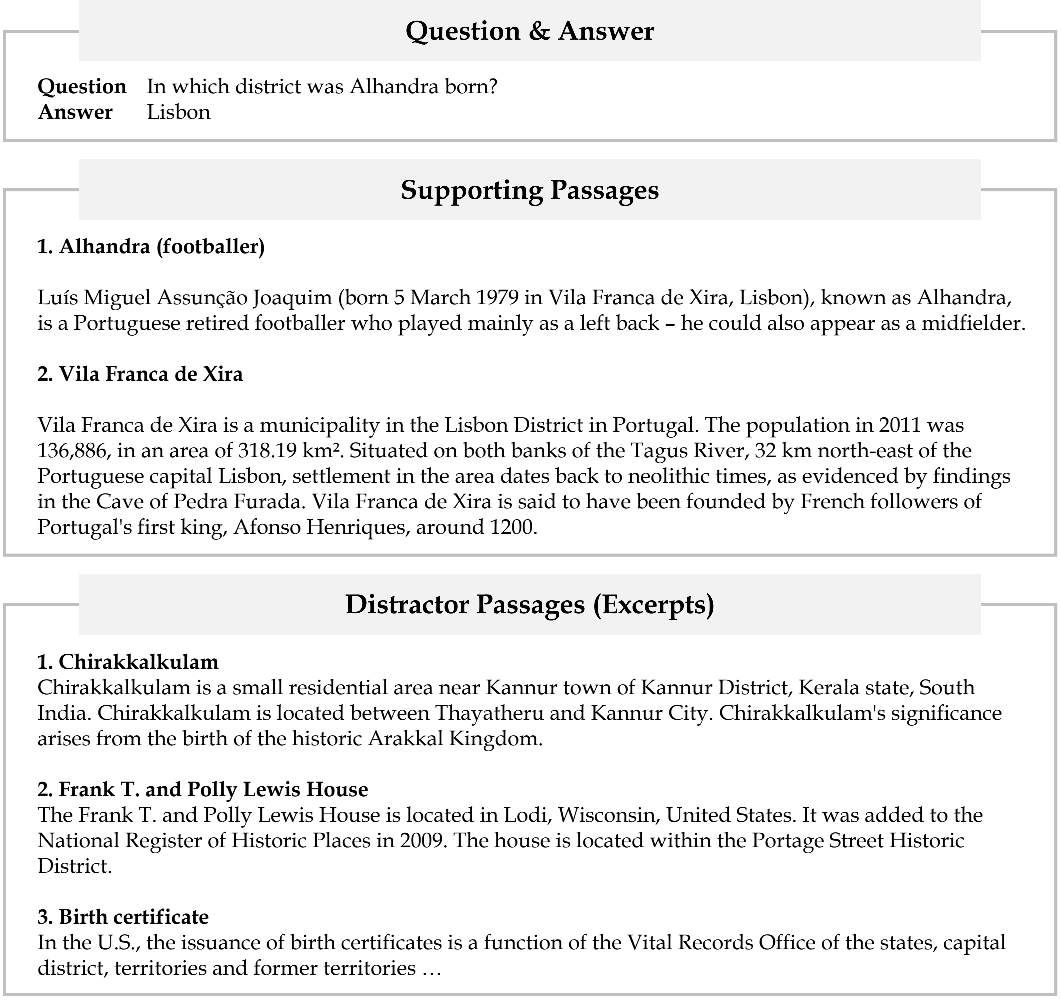
[79] S. Zhou, B. Yu, A. Sun, C. Long, J. Li, and J. Sun. A survey on neural open information extraction: Current status and future directions. In L. D. Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5694–5701. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/793. URL `https://doi.org/10.24963/ijcai.2022/793`. Survey Track.

[80] H. Zhu, H. Peng, Z. Lyu, L. Hou, J. Li, and J. Xiao. Pre-training language model incorporating domain-specific heterogeneous knowledge into a unified representation. *Expert Systems with Applications*, 215:119369, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.119369. URL `https://www.sciencedirect.com/science/article/pii/S0957417422023879`.

Appendices

Within this supplementary material, we elaborate on the following aspects:

- Appendix A: HippoRAG Pipeline Example
- Appendix B: Dataset Comparison
- Appendix C: Ablation Statistics
- Appendix D: Single-Step Multi-Hop: All-Recall Metric
- Appendix E: Path-Finding Multi-Hop Case Study
- Appendix F: Error Analysis on MuSiQue
- Appendix G: Cost and Efficiency Comparison
- Appendix H: Implementation Details & Compute Requirements
- Appendix I: LLM Prompts



Question & Answer

Question In which district was Alhandra born?
Answer Lisbon

Supporting Passages

1. Alhandra (footballer)

Luís Miguel Assunção Joaquim (born 5 March 1979 in Vila Franca de Xira, Lisbon), known as Alhandra, is a Portuguese retired footballer who played mainly as a left back – he could also appear as a midfielder.

2. Vila Franca de Xira

Vila Franca de Xira is a municipality in the Lisbon District in Portugal. The population in 2011 was 136,886, in an area of 318.19 km². Situated on both banks of the Tagus River, 32 km north-east of the Portuguese capital Lisbon, settlement in the area dates back to neolithic times, as evidenced by findings in the Cave of Pedra Furada. Vila Franca de Xira is said to have been founded by French followers of Portugal's first king, Afonso Henriques, around 1200.

Distractor Passages (Excerpts)

1. Chirakkalkulam
Chirakkalkulam is a small residential area near Kannur town of Kannur District, Kerala state, South India. Chirakkalkulam is located between Thayatheru and Kannur City. Chirakkalkulam's significance arises from the birth of the historic Arakkal Kingdom.

2. Frank T. and Polly Lewis House
The Frank T. and Polly Lewis House is located in Lodi, Wisconsin, United States. It was added to the National Register of Historic Places in 2009. The house is located within the Portage Street Historic District.

3. Birth certificate
In the U.S., the issuance of birth certificates is a function of the Vital Records Office of the states, capital district, territories and former territories …

Figure 3: **HippoRAG Pipeline Example (Question and Annotations). (Top)** We provide an example question and its answer. **(Middle & Bottom)** The supporting and distractor passages for this question. Two supporting passages are needed to solve this question. The excerpts of the distractor passages are related to the "district" mentioned in the question.

A HippoRAG Pipeline Example

To better demonstrate how our HippoRAG pipeline works, we use the *path-following* example from the MuSiQue dataset shown in Table 6. We use HippoRAG's indexing and retrieval processes to follow this question and a subset of the associated corpus. The question, its answer, and its supporting and distractor passages are as shown in Figure 3. The indexing stage is shown in Figure 4, showing both the OpenIE procedure as well as the relevant subgraph of our KG. Finally, we illustrate the retrieval stage in Figure 5, including query NER, query node retrieval, how the PPR algorithm changes node probabilities, and how the top retrieval results are calculated.

Indexing: Passage NER and OpenIE for Supporting Passages

1. Alhandra (footballer)
NER:
["5 March 1979", "Alhandra", "Lisbon", "Luís Miguel Assunção Joaquim", "Portuguese", "Vila Franca de Xira"]

OpenIE:
[("Alhandra", "is a", "footballer"),
("Alhandra", "born in", "Vila Franca de Xira"),
("Alhandra", "born in", "Lisbon"),
("Alhandra", "born on", "5 March 1979"),
("Alhandra", "is", "Portuguese"),
("Luís Miguel Assunção Joaquim", "is also known as", "Alhandra")]

2. Vila Franca de Xira
NER:
["2011", "Afonso Henriques", "Cave of Pedra Furada", "French", "Lisbon", "Lisbon District", "Portugal", "Tagus River", "Vila Franca de Xira"]

OpenIE:
[("Vila Franca de Xira", "is a municipality in", "Lisbon District"),
("Vila Franca de Xira", "located in", "Portugal"),
("Vila Franca de Xira", "situated on", "Tagus River"),
("Vila Franca de Xira", "is", "founded by French followers of Afonso Henriques"),
("Tagus River", "located near", "Lisbon"),
("Cave of Pedra Furada", "evidenced settlement in", "neolithic times"),
("Afonso Henriques", "was Portugal's first king in", "1200"),
("Vila Franca de Xira", "had population of", "136,886 in 2011"),
("Vila Franca de Xira", "has area of", "318.19 km^2")]

Indexing: Subgraph Related to the Question

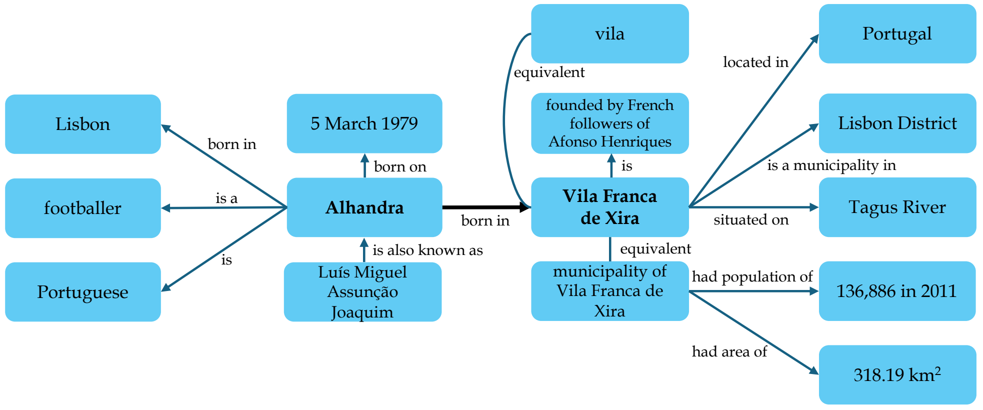


Figure 4: **HippoRAG Pipeline Example (Indexing).** NER and OpenIE are sequentially conducted on each passage of the corpus. Thus, an open knowledge graph is formed for the entire corpus. We only show the relevant subgraph from the KG.

Retrieval: Query NER & Node Retrieval

Question	In which district was Alhandra born?
NER	["Alhandra"]
Node Retrieval	{"Alhandra": "Alhandra"}

Retrieval: PPR

Node Probabilities Changes by PPR

Alhandra	1.000 ⇒ **0.533**	5 March 1979	0.000 ⇒ 0.045
Vila Franca de Xira	0.000 ⇒ **0.054**	Luís Miguel Assunção Joaquim	0.000 ⇒ 0.044
Lisbon	0.000 ⇒ 0.049	Portugal	0.000 ⇒ 0.009
footballer	0.000 ⇒ 0.047	Tagus River	0.000 ⇒ 0.007
Portuguese	0.000 ⇒ 0.046	José Pinto Coelho	0.000 ⇒ 0.004
…			

Retrieval: Top Results

*Top-ranked nodes from PPR are highlighted.

1. Alhandra (footballer)
Luís Miguel Assunção Joaquim (born 5 March 1979 in Vila Franca de Xira, Lisbon), known as Alhandra, is a Portuguese retired footballer who played mainly as a left back – he could also appear as a midfielder.

2. Vila Franca de Xira
Vila Franca de Xira is a municipality in the Lisbon District in Portugal. The population in 2011 was 136,886, in an area of 318.19 km². Situated on both banks of the Tagus River, 32 km north-east of the Portuguese capital Lisbon, settlement in the area dates back to neolithic times, as evidenced by findings in the Cave of Pedra Furada. Vila Franca de Xira is said to have been founded by French followers of Portugal's first king, Afonso Henriques, around 1200.

3. Portugal
Portuguese is the official language of Portugal. Portuguese is a Romance language that originated in what is now Galicia and Northern Portugal, originating from Galician-Portuguese, which was the common language of the Galician and Portuguese people until the independence of Portugal. Particularly in the North of Portugal, there are still many similarities between the Galician culture and the Portuguese culture. Galicia is a consultative observer of the Community of Portuguese Language Countries. According to the Ethnologue of Languages, Portuguese and Spanish have a lexical similarity of 89% - educated speakers of each language can communicate easily with one another.

4. Huguenots
The first Huguenots to leave France sought freedom from persecution in Switzerland and the Netherlands … A fort, named Fort Coligny, was built to protect them from attack from the Portuguese troops and Brazilian Native Americans. It was an attempt to establish a French colony in South America. The fort was destroyed in 1560 by the Portuguese, who captured part of the Huguenots. The Portuguese threatened the prisoners with death if they did not convert to Catholicism …

5. East Timor
Democratic Republic of Timor - Leste Repúblika Demokrátika Timór Lorosa'e (Tetum) República Democrática de Timor - Leste (Portuguese) Flag Coat of arms Motto: Unidade, Acção, Progresso (Portuguese) Unidade, Asaun, Progresu (Tetum) (English: ``Unity, Action, Progress '') Anthem: Pátria (Portuguese) (English:`` Fatherland'') Capital and largest city Dili 8 ° 20 ′ S 125 ° 20 ′ E / 8.34 ° S 125.34 ° E / - 8.34; 125.34 Coordinates: 8 ° 20 ′ S 125 ° 20 ′ E / 8.34 ° S 125.34 ° E / - 8.34; 125.34 …

Figure 5: **HippoRAG Pipeline Example (Retrieval).** For retrieval, the named entities in the query are extracted from the question **(Top)**, after which the query nodes are chosen using a retrieval encoder. In this case, the name of the query named entity, "Alhandra", is equivalent to its KG node. **(Middle)** We then set the personalized probabilities for PPR based on the retrieved query nodes. After PPR, the query node probability is distributed according to the subgraph in Figure 4, leading to some probability mass on the node "Vila France de Xira". **(Bottom)** These node probabilities are then summed over the passages they appear in to obtain the passage-level ranking. The top-ranked nodes after PPR are highlighted in the top-ranked passages.

B Dataset Comparison

To analyze the differences between the three datasets we use, we pay special attention to the quality of the distractor passages, i.e., whether they can be effectively confounded with the supporting passages. We use Contriever [27] to calculate the match score between questions and candidate passages and show their densities in Figure 6. In an ideal case, the distribution of distractor scores should be close to the mean of the support passage scores. However, it can be seen that the distribution of the distractor scores in HotpotQA is much closer to the lower bound of the support passage scores compared to the other two datasets.

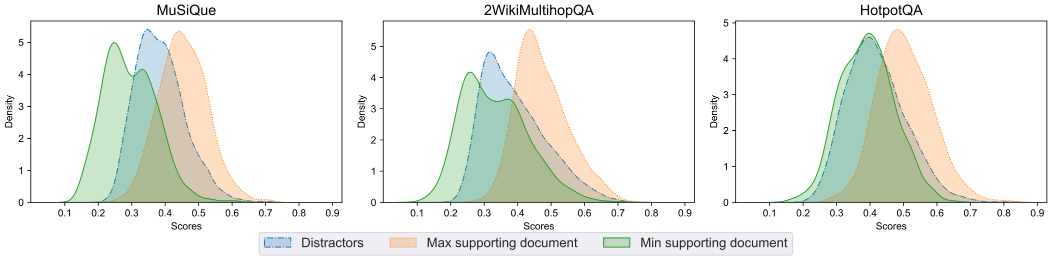


Figure 6: Density of similarity scores of candidate passages (distractors and supporting passages) obtained by Contriever. The similarity score of HotpotQA distractors is not significantly larger than that of the least similar supporting passages, meaning that these distractors are not very effective.

C Ablation Statistics

We use GPT-3.5 Turbo, REBEL [26] and Llama-3 (8B and 70B) [1] for OpenIE ablation experiments. As shown in Table 7, compared to both GPT-3.5 Turbo and both Llama models, REBEL generates around half the number of nodes and edges. This illustrates REBEL's lack of flexibility in open information extraction when compared to using both open and closed-source LLMs. Meanwhile, both Llama-3 versions produce a similar amount of OpenIE triples than GPT-3.5 Turbo.

Table 7: Knowledge graph statistics using different OpenIE methods.

Model	**Count**	**MuSiQue**	**2Wiki**	**HotpotQA**
GPT-3.5 Turbo [42] (Default)	# of Unique Nodes (N)	91,729	42,694	82,157
	# of Unique Edges (E)	21,714	7,867	17,523
	# of Unique Triples	107,448	50,671	98,709
	# of ColBERTv2 Synonym Edges (E')	191,636	82,526	171,856
REBEL [26]	# of Unique Nodes (N)	36,653	22,146	30,426
	# of Unique Edges (E)	269	211	262
	# of Unique Triples	52,102	30,428	42,038
	# of ColBERTv2 Synonym Edges (E')	48,213	33,072	39,053
Llama-3-8B-Instruct [1]	# of Unique Nodes (N)	89,759	44,251	78,959
	# of Unique Edges (E)	25,868	8,497	20,325
	# of Unique Triples	117,127	54,347	101,822
	# of ColBERTv2 Synonym Edges (E')	186,568	92,913	157,446
Llama-3-70B-Instruct [1]	# of Unique Nodes (N)	83,647	38,800	73,958
	# of Unique Edges (E)	20,694	6,917	15,663
	# of Unique Triples	111,996	51,379	100,372
	# of ColBERTv2 Synonym Edges (E')	177,979	75,463	150,291

D Single-Step Multi-Hop: All-Recall Metric

HippoRAG performs better on multi-hop QA benchmarks than standard RAG mainly because of its ability to perform multi-hop retrieval in a single step. Table 8 demonstrates this by measuring the percentage of queries where all supporting passages are retrieved successfully, something that can only be achieved through successful multi-hop reasoning. The fact that the gaps shown in this table are much larger than the gaps in Table 2 indicates that much of HippoRAG's improvements are coming from the ability to retrieve all supporting documents not by performing better partial retrieval.

Table 8: **All-Recall metric.** We measure the percentage of queries for which all supporting passages are successfully retrieved (all-recall, denoted as AR@2 or AR@5) and find even larger performance improvements for HippoRAG.

	MuSiQue		**2Wiki**		**HotpotQA**		**Average**	
	AR@2	AR@5	AR@2	AR@5	AR@2	AR@5	AR@2	AR@5
ColBERTv2 [53]	6.8	16.1	25.1	37.1	33.3	59.0	21.7	37.4
HippoRAG	10.2	22.4	45.4	75.7	33.8	57.9	29.8	52.0

E Case Study on Path-Finding Multi-Hop QA

As discussed above, path-finding multi-hop questions across passages are exceedingly challenging for single-step and multi-step RAG methods such as ColBERTv2 and IRCoT. These questions require integrating information across multiple passages to find relevant entities among many possible candidates, such as finding all Stanford professors who work on the neuroscience of Alzheimer's.

E.1 Path-Finding Multi-Hop Question Construction Process

These questions and the curated corpora around them were built through the following procedure. The first two questions follow a slightly separate process as the third one as well as the motivating example in the main paper. For the first two, we first identify a book or movie and then found the book's author or the movie's director. We would then find 1) a trait for either the book/movie and 2) another trait for the author/director. These two traits would then be used to extract distractors from Wikipedia for each question.

For the third question and our motivating example, we first choose a professor or a drug at random as the answer for each question. We then obtain the university the professor works at or the disease the drug treats as well as one other trait for the professor or drug (in these questions research topic and mechanism of action were chosen). In these questions, distractors were extracted from Wikipedia using the University or disease on the one hand and the research topic or mechanism of action on the other. This process, although quite tedious, allowed us to curate these challenging but realistic path-finding multi-hop questions.

E.2 Qualitative Analysis

In Table 9, we show three more examples from three different domains that illustrate HippoRAG's potential for solving retrieval tasks that require such cross-passage knowledge integration.

In the first question of Table 9, we want to find a book published in **2012** by an English author who won a specific award. In contrast to HippoRAG, ColBERTv2 and IRCoT are unable to identify **Mark Haddon** as such an author. ColBERTv2 focuses on passages related to awards while IRCoT mistakenly decides that Kate Atkinson is the answer to such question since she won the same award for a book published in 1995. For the second question, we wanted to find a war film based on a non-fiction book directed by someone famous for sci-fi and crime movies. HippoRAG is able to find our answer **Black Hawk Down** by **Ridley Scott** within the first four passages, while ColBERTv2 misses the answer completely and retrieves other films and film collections. In this instance, even though IRCoT is able to retrieve Ridley Scott, it does so mainly through parametric knowledge. The chain-of-thought output discusses his and Denis Villeneuve fame as well as their sci-fi and crime experience. Given the three-step iteration restriction used here and the need to explore two directors, the specific war film **Black Hawk Down** was not identified. Although a bit convoluted, people often ask these first two questions to remember a specific movie or book they watched or heard about from only a handful of disjointed details.

Finally, the third question is more similar to the motivating example in the main paper and shows the importance of this type of question in real-world domains. In this question, we ask for a drug used to treat lymphocytic leukemia through a specific mechanism (cytosolic p53 interaction). While HippoRAG is able to leverage the associations within the supporting passages to identify the **Chlorambucil** passage as the most important, ColBERTv2 and IRCoT are only able to extract passages associated with lymphocytic leukemia. Interestingly enough, IRCoT uses its parametric knowledge to guess that Venetoclax, which also treats leukemia, would do so through the relevant mechanism even though no passage in the curated dataset explicitly stated this.

Table 9: Ranking result examples for different approaches on several path-finding multi-hop questions.

Question	HippoRAG	ColBERTv2	IRCoT
Which book was published in **2012** by an **English** author who is a **Whitbread Award** winner?	**1.** Oranges Are Not the Only Fruit **2.** William Trevor Legacies **3. Mark Haddon**	**1.** World Book Club Prize winners **2.** Leon Garfield Awards **3.** Twelve Bar Blues (novel)	**1.** Kate Atkinson **2.** Leon Garfield Awards **3.** Twelve Bar Blues (novel)
Which **war film** based on a **non fiction book** was directed by someone famous in the **science fiction** and **crime genres**?	**1.** War Film **2.** Time de Zarn **3.** Outline of Sci-Fi **4. Black Hawk Down**	**1.** Paul Greengrass **2.** List of book-based war films **3.** Korean War Films **4.** All the King's Men Book	**1. Ridley Scott** **2.** Peter Hyams **3.** Paul Greengrass **4.** List of book-based war films
What drug is used to treat **chronic lymphocytic leukemia** by interacting with **cytosolic p53**?	**1. Chlorambucil** **2. Lymphocytic leukemia** **3.** Mosquito bite allergy	**1. Lymphocytic leukemia** **2.** Obinutuzumab **3.** Venetoclax	**1.** Venetoclax **2. Lymphocytic leukemia** **3.** Idelalisib

F Error Analysis

F.1 Overview

We mainly analyze the errors by HippoRAG on the MuSiQue dataset. As shown in Table 10, these errors can be attributed to three types. The NER limitation is the main error type (nearly half), as NER may not extract enough information from the query for retrieval, e.g., for the question "When was one internet browser's version of Windows 8 made accessible?", only the phrase "Windows 8" is extracted. Nothing about "browser" or "accessibility" is extracted for subsequent graph search.

For OpenIE errors, we discuss some examples in §F.3. Despite the correct functioning of NER and OpenIE, the PPR algorithm sometimes fails to identify relevant passages in the graph correctly. For instance, consider the question "How many refugees emigrated to the European country where Huguenots felt a kinship for emigration?". Despite the supporting passages being titled "Huguenots" as shown in the question, the PPR algorithm struggles due to multiple passages with similar topics within the corpus. Even when the term "Huguenots" is accurately extracted from both the question and the supporting passages, and the PPR algorithm initiates searches with the nodes labeled "European" and "Huguenots", it incorrectly prioritizes other passages containing "European" and "Huguenots" that do not actually support the question. This misdirection occurs because the algorithm does not effectively discriminate between passages with similar topics but different relevance to the question's context.

Table 10: Error analysis on MuSiQue.

Error Type	**Error Percentage** (%)
NER Limitation	48
Incorrect/Missing OpenIE	28
PPR	24

F.2 Concepts vs. Context Tradeoff

Given our method's entity-centric nature in extraction and indexing, it has a strong bias towards concepts that leaves many contextual signals unused. This design enables single-step multi-hop retrieval while also enabling contextual cues to avoid distracting from more salient entities. As seen in the first example in Table 11, ColBERTv2 uses the context to retrieve passages that are related to famous Spanish navigators but not "Sergio Villanueva", who is a boxer. In contrast, HippoRAG is able to hone in on "Sergio" and retrieve one relevant passage.

Unfortunately, this design is also one of our method's greatest limitations since ignoring contextual cues accounts for around 48% of errors in a small-scale error analysis; more details can be found in Appendix F. This problem is more apparent in the second example since the concepts are general, making the context more important. Since the only concept tagged by HippoRAG is "protons", it extracts passages related to "Uranium" and "nuclear weapons" while ColBERTv2 uses the context to extract more relevant passages associated with the discovery of atomic numbers.

Table 11: Examples showing the concept-context tradeoff on MuSiQue.

Question	**HippoRAG**	**ColBERTv2**
Whose father was a navigator who explored the east coast of the continental region where **Sergio Villanueva** would later be born?	**Sergio Villanueva** César Gaytan Faustino Reyes	Francisco de Eliza (navigator) Exploration of N. America Vicente Pinzón (navigator)
What undertaking included the person who discovered that the number of **protons** in each element's atoms is unique?	Uranium Chemical element History of nuclear weapons	**Atomic number** Atomic theory Atomic nucleus

Table 12: **Single-step retrieval performance.** HippoRAG performs significantly better on MuSiQue and 2WikiMultiHopQA than all baselines and achieves comparable performance on the less challenging HotpotQA dataset.

Model	**Retriever**	**MuSiQue**		**2Wiki**		**HotpotQA**		**Average**	
		R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
Baseline	Contriever	34.8	46.6	46.6	57.5	57.2	75.5	46.2	59.9
	ColBERTv2	37.9	49.2	59.2	68.2	**64.7**	79.3	53.9	65.6
HippoRAG	Contriever	41.0	52.1	71.5	**89.5**	59.0	76.2	57.2	72.6
	ColBERTv2	40.9	51.9	70.7	89.1	60.5	77.7	57.4	72.9
HippoRAG w/ Uncertainty-based Ensemble	Contriever	42.3	54.5	71.3	87.2	60.6	79.1	58.1	73.6
	ColBERTv2	**42.5**	**54.8**	**71.9**	89.0	62.5	**80.0**	**59.0**	**74.6**

To get a better trade-off between concepts and context, we introduce an ensembling setting where HippoRAG scores are ensembled with dense retrievers when our parahippocampal region shows uncertainty regarding the link between query and KG entities. This process represents instances when no hippocampal index was fully activated by the upstream parahippocampal signal and thus the neocortex must be relied on more strongly. We only use uncertainty-based ensembling if one of the query-KG entity scores $cosine\_similarity(M(c_i), M(e_j))$ is lower than a threshold θ, for example, if there was no *Stanford* node in the KG and the closest node in the KG is something that has a cosine similarity lower than θ such as *Stanford Medical Center*. The final passage score for uncertainty-based ensembling is the average of the HippoRAG scores and standard passage retrieval using model M, both of which are first normalized into the 0 to 1 over all passages.

When HippoRAG is ensembled with M under *"Uncertainty-based Ensembling"*, it further improves on MuSiQue and outperforms our baselines in R@5 for HotpotQA, as shown in Table 12. When used in combination with IRCoT, as shown in Table 13, the ColBERTv2 ensemble outperforms all previous baselines in both R@2 and R@5 on HotpotQA. Although the simplicity of this approach is promising, more work needs to be done to solve this context-context tradeoff since simple ensembling does lower performance in some cases, especially for the 2WikiMultiHopQA dataset.

Table 13: **Multi-step retrieval performance.** Combining HippoRAG with standard multi-step retrieval methods like IRCoT results in significant improvements on all three datasets.

Model	**Retriever**	**MuSiQue**		**2Wiki**		**HotpotQA**		**Average**	
		R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
IRCoT	Contriever	39.1	52.2	51.6	63.8	65.9	81.6	52.2	65.9
	ColBERTv2	41.7	53.7	64.1	74.4	67.9	82.0	57.9	70.0
IRCoT + HippoRAG	Contriever	43.9	56.6	75.3	93.4	65.8	82.3	61.7	77.4
	ColBERTv2	**45.3**	57.6	**75.8**	**93.9**	67.0	83.0	**62.7**	78.2
IRCoT + HippoRAG w/ Uncertainty-based Ensemble	Contriever	44.4	**58.5**	75.3	91.5	66.9	85.0	62.2	**78.3**
	ColBERTv2	40.2	53.4	74.5	91.2	**68.2**	**85.3**	61.0	76.6

F.3 OpenIE Limitations

OpenIE is a critical step in extracting structured knowledge from unstructured text. Nonetheless, its shortcomings can result in gaps in knowledge that may impair retrieval and QA capabilities. As shown in Table 14, GPT-3.5 Turbo overlooks the crucial song title "Don't Let Me Wait Too Long" during the OpenIE process. This title represents the most significant element within the passage. A probable reason is that the model is insensitive to such a long entity. Besides, the model does not accurately capture the beginning and ending years of the war, which are essential for the query. This is an example of how models routinely ignore temporal properties. Overall, these failures highlight the need to improve the extraction of critical information.

Table 14: Open information extraction error examples on MuSiQue.

Question	**Passage**	**Missed Triples**
What company is the label responsible for "Don't Let Me Wait Too Long" a part of?	"Don't Let Me Wait Too Long" was sequenced on side one of the LP, between the ballads "The Light That Has Lighted the World" and "Who Can See It" ...	(Don't Let Me Wait Too Long, sequenced on, side one of the LP)
When did the president of the Confederate States of America end his fight in the Mexican-American war?	Jefferson Davis fought in the Mexican–American War (1846–1848), as the colonel of a volunteer regiment ...	(Mexican-American War, starts, 1846), (Mexican-American War, ends, 1848)

G Cost and Efficiency Comparison

One of HippoRAG's main advantages against iterative retrieval methods is the dramatic online retrieval efficiency gains brought on by its single-step multi-hop retrieval ability in terms of both cost and time. Specifically, as seen in Table 15, retrieval costs for IRCoT are 10 to 30 times higher than HippoRAG since it only requires extracting relevant named entities from the query instead of processing all of the retrieved documents. In systems with extremely high usage, a cost difference of an order of magnitude such as this one could be extremely important. The difference with IRCoT in terms of latency is also substantial, although more challenging to measure exactly. Also as seen in Table 15, HippoRAG can be 6 to 13 times faster than IRCoT, depending on the number of retrieval rounds that need to be executed (2-4 in our experiments).[6]

[6] We use a single thread to query the OpenAI API for online retrieval in both IRCoT and HippoRAG. Since IRCoT is an iterative process and each of the iterations must be done sequentially, these speed comparisons are appropriate.

Table 15: Average cost and efficiency measurements for online retrieval using GPT-3.5 Turbo on 1,000 queries.

	ColBERTv2	IRCoT	HippoRAG
API Cost ($)	0	1-3	0.1
Time (minutes)	1	20-40	3

Although offline indexing time and costs are higher for HippoRAG than IRCoT—around 10 times slower and $15 more expensive for every 10,000 passages [7], these costs can be dramatically reduced by leveraging open source LLMs. As shown in our ablation study in Table 5 Llama-3-8B-Instruct [1] performs similarly to GPT-3.5 Turbo without any specific prompt tuning. As seen in Table 16, this 8B Llama-3 model is about half the cost as GPT-3.5 Turbo when using an LLM deploying service like Together AI [8]. Additionally, since these costs could be even further reduced by locally deploying this model, the barriers for using HippoRAG at scale could be well within the computational budget of many organizations. Finally, we note that even if LLM generation cost drops significantly, the online retrieval efficiency gains discussed above remain intact given that the number of tokens required for IRCoT vs. HippoRAG stay constant and LLM use is likely to also remain the system's main computational bottleneck.

Table 16: Average cost and latency measurements for offline indexing using GPT-3.5 Turbo and Llama-3 (8B and 70B) through the TogetherAI API on 10,000 passages.

Model	Metric	ColBERTv2	IRCoT	HippoRAG
GPT-3.5 Turbo-1106 (Main Results)	API Cost ($)	0	0	15
	Time (minutes)	7	7	60
GPT-3.5 Turbo	API Cost ($)	0	0	8
	Time (minutes)	7	7	60
Llama-3-8B-Instruct	API Cost ($)	0	0	3
	Time (minutes)	7	7	45
Llama-3-70B-Instruct	API Cost ($)	0	0	7
	Time (minutes)	7	7	100

H Implementation Details & Compute Requirements

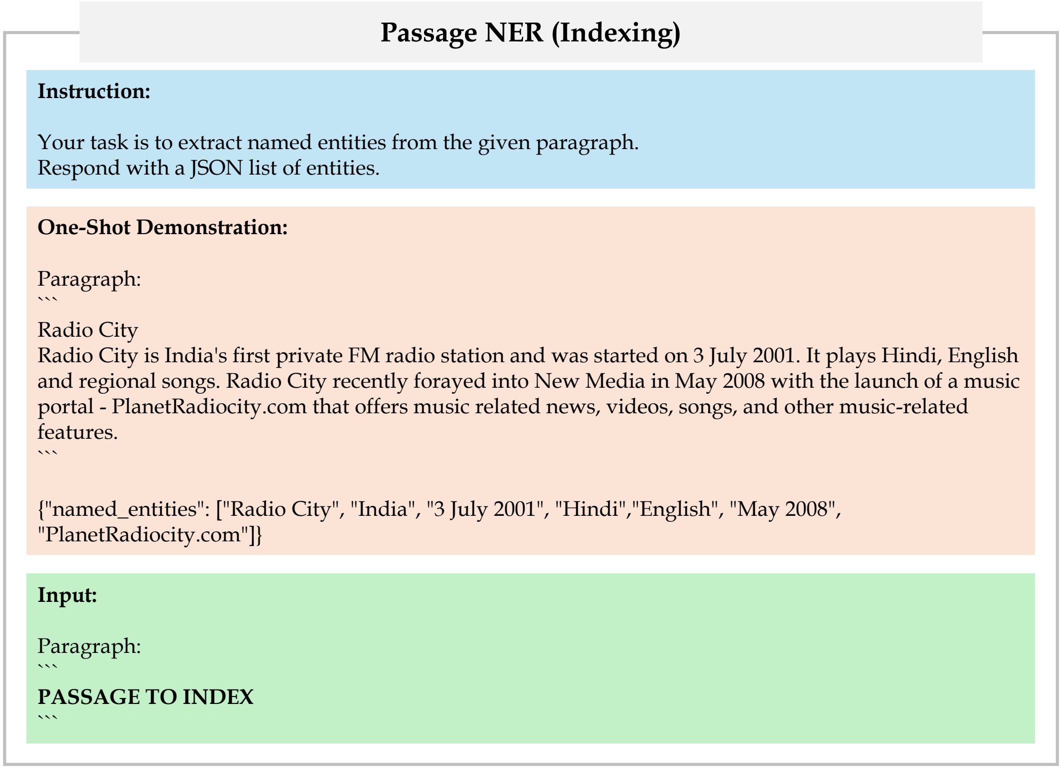
Apart from the details included in §3.4, we use implementations based on PyTorch [44] and HuggingFace [68] for both Contriever [27] and ColBERTv2 [53]. We use the python-igraph [11] implementation of the PPR algorithm. For BM25, we employ Elastic Search [18]. For multi-step retrieval, we use the same prompt implementation as IRCoT [61] and retrieve the top-10 passages at each step. We set the maximum number of reasoning steps to 2 for HotpotQA and 2WikiMultiHopQA and 4 for MuSiQue due to their maximum reasoning chain length. We combine IRCoT with different retrievers by replacing its base retriever BM25 with each retrieval method, including HippoRAG, noted as "IRCoT + HippoRAG" below.[9] For the QA reader, we use top-5 retrieved passages as the context and 1-shot QA demonstration with CoT prompting strategy [61].

In terms of compute requirements, most of our compute requirements are unfortunately not disclosed by the OpenAI and TogetherAI APIs. We run ColBERTv2 and Contriever for indexing and retrieval we use 4 NVIDIA RTX A6000 GPUs with 48GB of memory. Finally, we used 2 AMD EPYC 7513 32-Core Processors to run the Personalized PageRank algorithm.

[7] To speed up indexing, we use 10 threads querying *gpt-3.5-turbo-1106* through the OpenAI API in parallel. At the time of writing, the cost of the API is $1 for a million input tokens and $2 for a million output tokens.

[8] `https://www.together.ai/products#inference`

[9] Since the original IRCoT does not provide a score for each retrieved passage, we employ beam search for the iterative retrieval process. Each candidate passage maintains the highest historical score during beam search.



Passage NER (Indexing)

Instruction:

Your task is to extract named entities from the given paragraph.
Respond with a JSON list of entities.

One-Shot Demonstration:

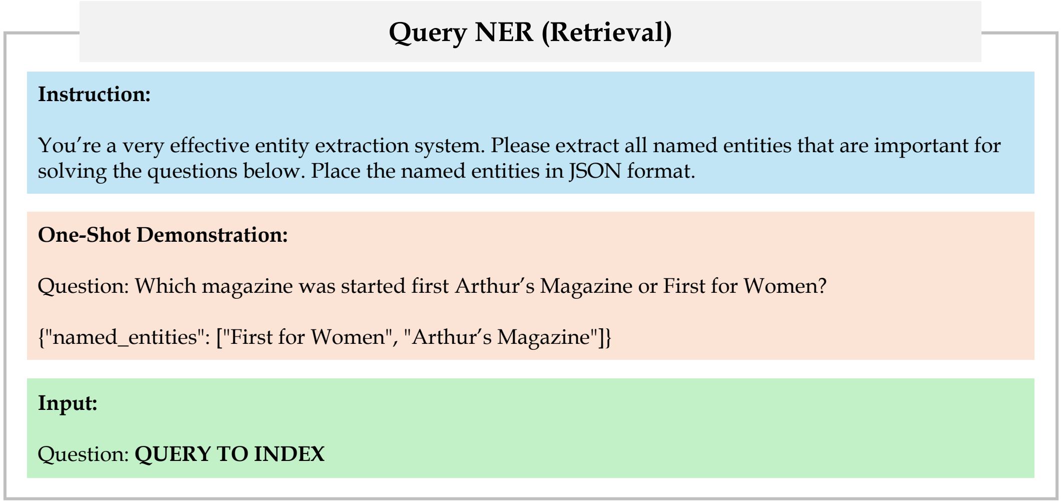
Paragraph:
```
Radio City
Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.
```

{"named_entities": ["Radio City", "India", "3 July 2001", "Hindi","English", "May 2008", "PlanetRadiocity.com"]}

Input:

Paragraph:
```
**PASSAGE TO INDEX**
```

Figure 7: Prompt for passage NER during indexing.



Query NER (Retrieval)

Instruction:

You're a very effective entity extraction system. Please extract all named entities that are important for solving the questions below. Place the named entities in JSON format.

One-Shot Demonstration:

Question: Which magazine was started first Arthur's Magazine or First for Women?

{"named_entities": ["First for Women", "Arthur's Magazine"]}

Input:

Question: **QUERY TO INDEX**

Figure 8: Prompt for query NER during retrieval.

I LLM Prompts

The prompts we used for indexing and query NER are shown in Figure 7 and Figure 8, while the OpenIE prompt is shown in Figure 9.

Open Information Extraction

Instruction:

Your task is to construct an RDF (Resource Description Framework) graph from the given passages and named entity lists.
Respond with a JSON list of triples, with each triple representing a relationship in the RDF graph.
Pay attention to the following requirements:
- Each triple should contain at least one, but preferably two, of the named entities in the list for each passage.
- Clearly resolve pronouns to their specific names to maintain clarity.

Convert the paragraph into a JSON dict, it has a named entity list and a triple list.

One-Shot Demonstration:

Paragraph:

~~~~
```
Radio City
Radio City is India’s first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.
```
{"named_entities": ["Radio City", "India", "3 July 2001", "Hindi","English", "May 2008", "PlanetRadiocity.com"]}

{"triples":
  [
    ["Radio City", "located in", "India"],
    ["Radio City", "is", "private FM radio station"],
    ["Radio City", "started on", "3 July 2001"],
    ["Radio City", "plays songs in", "Hindi"],
    ["Radio City", "plays songs in", "English"],
    ["Radio City", "forayed into", "New Media"],
    ["Radio City", "launched", "PlanetRadiocity.com"],
    ["PlanetRadiocity.com", "launched in", "May 2008"],
    ["PlanetRadiocity.com", "is", "music portal"],
    ["PlanetRadiocity.com", "offers", "news"],
    ["PlanetRadiocity.com", "offers", "videos"],
    ["PlanetRadiocity.com", "offers", "songs"]
  ]
}
~~~~

Input:

Convert the paragraph into a JSON dict, it has a named entity list and a triple list.
Paragraph:

~~~~
```
PASSAGE TO INDEX
```
{"named_entities": [NER LIST]}
~~~~

Figure 9: Prompt for OpenIE during indexing.