ExoChat

http://www.exodao.net

ExoChat Commons: Augmented LLM IR (ALLMIR)

June 2024 Draft 1.10

Self-contained 100% open source reference implementation of a human centric paradigm for search enhanced generative Q&A: Augmented Large Language Model Information Retrieval (ALLMIR) designed to meet the demands of energy/resource efficiency, governance, safety and plesiochronousity (minimized update hysteresis).

Background/Context

The ExoDAO Network Association, is tasked to develop a decentralized concept to catalyze the digital commons, prioritizing user privacy, transparency, inclusion, energy efficiency and information discovery. It was conceived as the result of our participation as re-Isearch in the NGI TETRA program in 2021 and established as an association in Zurich, Switzerland in 2022. It is housed both within the SPH of the Swiss Federal Institute of Technology (ETH Zurich) as well as in Munich.

Each self-contained node (software) currently includes a sophisticated, low footprint, high performance engine, re-Isearch (a kind of hybrid between full-text, XML, object and graph noSQL search engine that natively ingests a wide range of document types and formats). The stack includes alongside its own protocols other standards such as OASIS SRU/W and NISO-Z39.50/ISO-23950 widely used in the library and archive community. While the re-Isearch native query language is extremely powerful offering search into tree nodes, paths, proximity, a large wealth of binary and unary operators, polymorphic objects (more than 30 object types including numerical, dates, geospatial, numerous hashes that can be searched as both text and their encoded value) etc. joins between indexes and one can even use glob (wildcard expressions) on terms; it is also by its nature complex. Its query paradigm can prove daunting for those without a background in predicate logic and graphs to use its power to the fullest. While the query execution is based around an RPN based stack language, alternative expressive languages are already provided such as algebraic (infix) expressions (its own and hooks for OASIS CQL) and there is a smart mode using only terms for neophytes (see the extensive re-Isearch handbook) that tries to guess an optimal search intent but it is by its very nature, limited.

Problem(s):

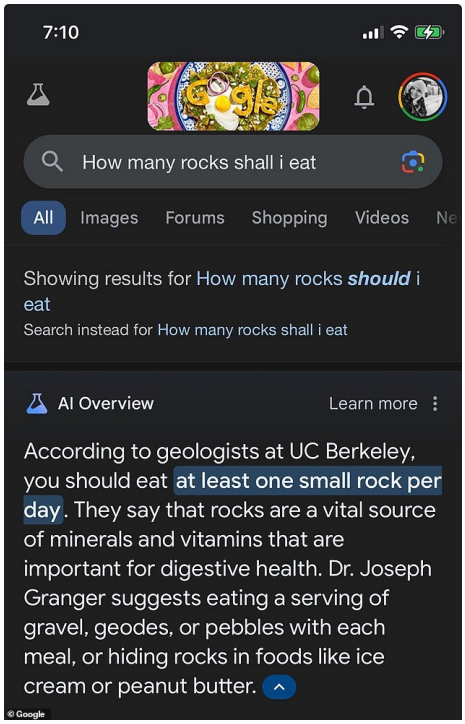
With Large Language Models (LLMs) becoming increasingly popular and prevalent for Q&A tasks, often as an ersatz for search, we had a large number of requests for ExoDAO to join the fray. A number of critical issues poked their head out:

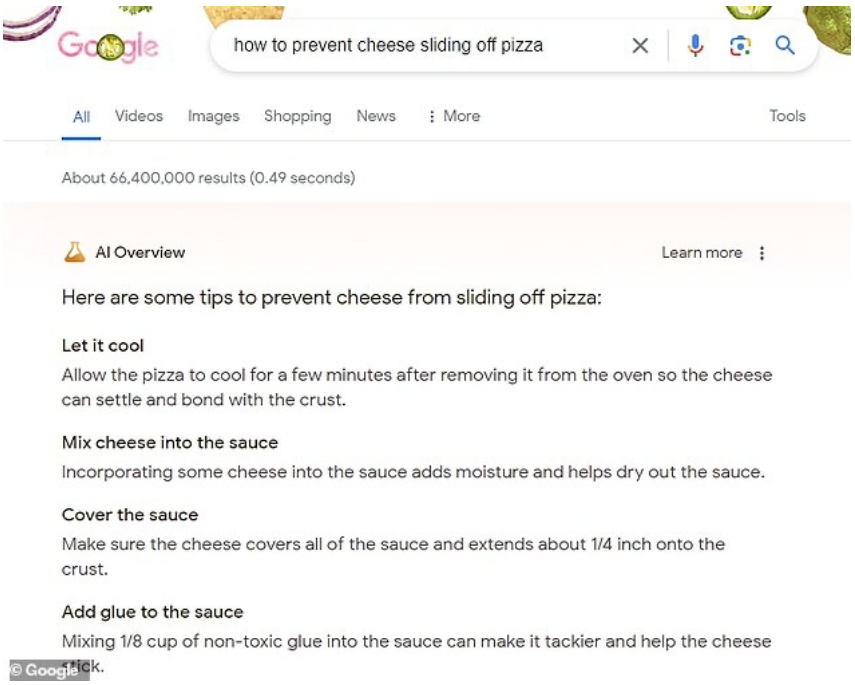
1. The normative ChatGPT paradigm gives just an answer. It reinforces the content “winner take all” feature—a significant flaw from the perspective of information discovery—of mainline search engines. In these search engines its a direct product of the information dominance metrics. In LLM its really no different but even more opaque. It can potentially constrain the already limited exposure to diverse perspectives and ideas. As a plank of the attention economy it opens also new markets for “product placement” sans the legal framework for labeling such as “sponsored”.

2. As an ersatz for search, the origin and veracity of information can often

become obscured and difficult to vet as they typically don’t provide a traceback to their information sources. This lack of transparency hinders users' ability to discern the credibility and reliability of the information they encounter during searches.

The training of the models can’t distinguish between fact, disinformation, jokes and irony. Training, for example, on Onion (a satirical publication) or Redit has resulted in such dialogs:





3. They have not just a tendency to to hallucinate and generate fictitious answers but have also proven quite easy to lobotomize or poison by bad-agents to propagate disinformation. For civil society it is an immanent danger. Without traceability for data with clear provenance, models are also easy targets (see our 40F.ai talk at FOSDEM ‘24 where we talked about risks and our approach to mitigate the threats). They also tend to, given the computational demands to update or tune, have a high level of hysteresis to the incorporation of new content.

4. For many, low resource, less common languages— more than 50% of the top 10 million sites in the Web are in English followed by Spanish, the 2nd most common spoken language on the planet, at under 6%—especially those found in less developed economies this is not just fiscally prohibitive but not even possible. There needs to be a way to reduce the size of model training and tuning as well as somehow cheaply augment by domain and language specific additional local text. Even within the European Union this is highly relevant with some of its languages with content less frequent than in 0.1% of sites.

5. In many countries there is insufficient computational resources (data centers) to follow the dominant paradigm. In all of Africa there is perhaps just one adequate data center. Even in developed nations the cost and computational resources to operate LLMs places too much power and control in the hands of the owners of these resources. They are also neither environmentally nor politically sustainable.

6. Data ownership. Putting local data into someone's cloud/LLM may not be desired. A solution that optionally enables the technology to operate on own consumer hardware or at the edge brings control closer to owner. Cloud is also expensive for non-elastic computational demands.

7. A fundamental feature of our core moon-shot project ExoDAO of which ExoChat is a subproject is the need to operate within a large hyper-distributed network to shift control from centralized and energy hungry data-centers towards using existing underutilized computation.

8. Another goal of ExoDAO is make search ranking and sorting transparent and put it into the hands of the searcher. Following this, we need to bring the human into the LLM loop to make Q&A human-centric.

Solution:

While Retrieval Augmented text Generation (RAG[1]) addresses the issues of pre-training, allowing for (some) on-the-fly changes it still does not address the difficulty of LLMs to put in safeguards and guardrails to block inappropriate content. Despite even significant "safety" training they can still be exploited by so-called "zero-day" trigger exploits easily embedded especially in the augmented data[2]

Instead of purely using the retrieved passages and feeding them into the LLM we instead envision a concurrent "search-like" human interface. Here we can readily filter responses at search time as needed.

The re-Isearch engine can at search-time respond to geo-location, user rights or other issues which may define what constitutes as "inappropriate" as it is just toggling a single bit. Through its virtual concept even the presentation of results can be fully controlled according to business logic rules without changing the index—even interfacing with an external database such as a RDBMS, for example, to fetch things like realtime inventory, pricing, exchange rates and other volatile data. Its modular design allows even for queries at real-time into specific paths/fields to feed into other pipelines for absolute up-to-date result presentations (for example, in the past, to list product availability, pricing across different currencies etc.)—even into other generative systems.

Hybrid Search Proposal:

1 Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

2 Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training

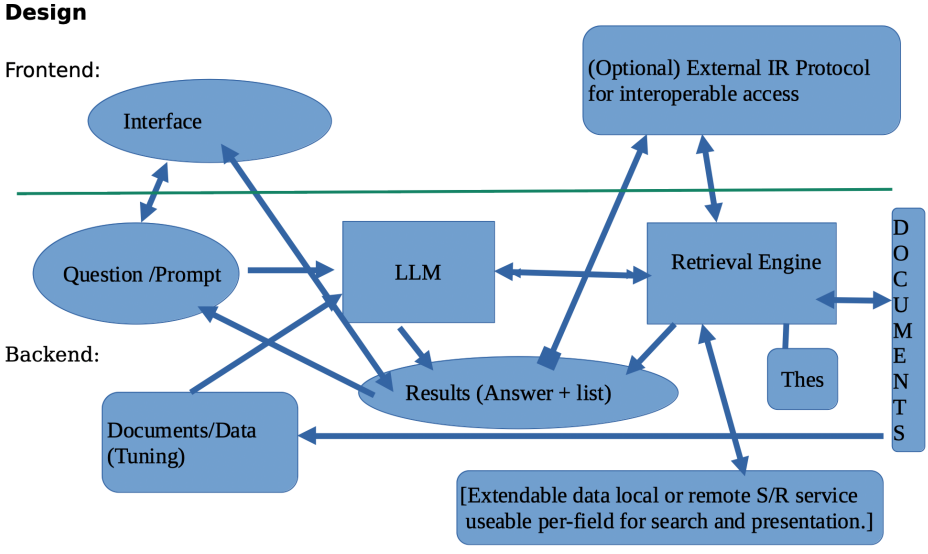
We propose here not just a more “accessible” user interface using state of the art natural language processing following the “chat” (popularized by OpenAI’s ChatGPT) paradigm to bring down its barriers—prompting instead of query expression—but also a significant enhancement to the Q&A “chat” model in several significant ways. The ExoChat sub-project aims to fuse chat[3] generative models with re-Isearch while also addressing some of the fundamental issues with the former including their reliability, data hysteresis and computational resource demands. Its design is intended for self-hosting but not exclude cloud.

We have called this approach *Augmented Large Language Model Information Retrieval* (ALLMIR) as a human centric paradigm to clearly distinguish it from other models of purely semantic enhancement of search (typically around a text expansion model) or RAG. For the former, the engine’s (optional) term normalization mode (Thes)—its enhanced weighted hierarchical thesaurus—provides the hooks out of the box. Another feature is performance: people expect quick and timely responses—this is in strong contrast to RAG-fusion.

The documents may be constantly updated, revised, appended. The re-Isearch engine can ingest these, by design, continuously without impacting search—no need to re-index. Changed documents are automatically updated—the previous version gets its versioning updated and, of so desired, marked internally as deleted (its a single bit, address masked at the sign bit) to not appear in searches—once can even “undelete”. Garbage may be collected to, if needed, remove these. There is even “trash compaction” as a means to completely remove all traces of the data in the index without the need to re-index. This is motivated by the observation that there are use cases where re-indexing may not be possible.

What is particularly interesting for some searches is the ability to have things joined. In an RDMS a join statement is mainly used to combine two tables based on a specified common field between them. If we talk in terms of Relational algebra, it is the cartesian product of two tables followed by the selection operation. In a Graph Database we traverse and don’t join. We build and extend a graph. There is the concept of graph and not document (or record). While re-Isearch can traverse the implicit graph of a record it can do more. It is designed and optimized for a generic and flexible search and provides a mechanism to support searching for virtually joined (or related) records in completely different indexes via a shared common index key. Our joins can apply across results from search in different indexes (normal conjugation like AND, OR can only apply within an index).

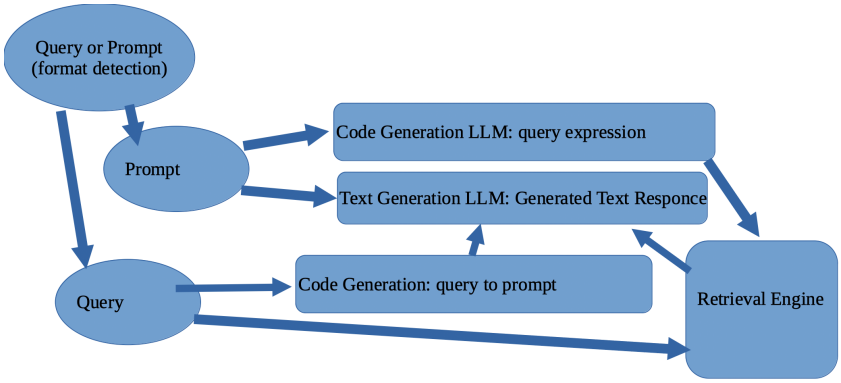
Design



(the optional external IR protocols include, but are not limited to, the lingua franca of libraries: Z39.50/ISO23950 and its Web counterpart OASIS SRU/W)

3

Our approach exploits tuned LLMs for two main tasks: code generation (query) and parallel Q&A text generation and augmentation—they can but need not be the same LLM. Input can be natural language prompts or in one of the supported query formats (such as RPN, Infix, Smart etc.).



ALLMIR vs RAG:

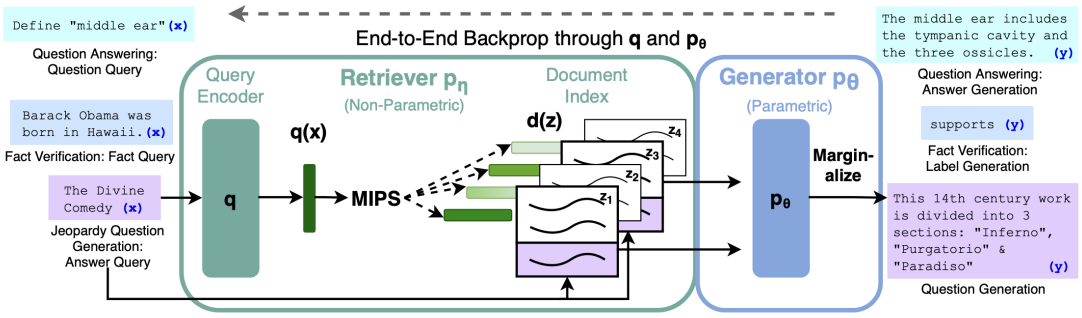
One of the typical methods to improve generated text—to produce more “up to to date” responses, control knowledge sources or to counter hallucinations caused by information gaps—is to fortify the prompt with supplemental text

Google’s Gemini 1.5 one has a 128K context window and even, for enterprise customers, up to 1 million but it is purely API, specifically housed on Google’s platform and is also relatively energy hungry. It is also a closed AI model and developers either build on its API or via their Vertex AI platform.

Common models suitable to self-hosting such as LLaMa2/LLaMa3l 4k, resp. 8k, ChaGLM, 8k context sizes. Mistral-7B has a context length of 32k. Via sliding window attention[4] the theoretical (attention) span be extended. This is too small to be able to include the whole local or updated corpus but only some bits.

RAG is a popular means to try to maneuver out of this constraint but because of the fixed unit of retrieval in typical vector databases used for Dense Passage Retrieval (DPR) they demand a prior segmentation of content into size constrained blobs or passages.

4 Longformer: The Long-Document Transformer: https://arxiv.org/pdf/2004.05150v2.pdf

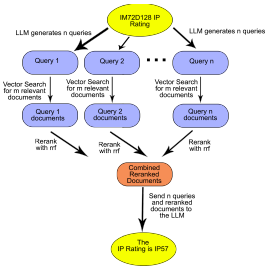


Key to the proper functioning is appropriate data chunking. By optimizing these chunks for context one can enhance the model's ability to generate coherent and contextually appropriate responses. The challenge lies in determining the optimal size for these chunks. Here lies a significant trade-off problem: granularity and context. If the segments are too large or too small, it can lead to issues such as incomplete information, fragmented sentences, or even a loss of semantic coherence.

The ALLMIR paradigm takes an approach to exploit data (document) structure to define the chunks. On the search retrieval side (full-text retrieval) alongside the search for similar chunks we have a retrieval of its own model of relevant bits using its ability to retrieve structural elements: dynamic unit of retrieval. This approach maintains the original author's organization of content and helps keep the text coherent. It builds on the idea that documents are implicitly structured to be understood by humans using either explicit markup or implicit structure such as lines, sentences or paragraphs. It also builds on the notion that meaning is communicated through also structure so needs to be viewed in the context of structure.

Versus RAG-Fusion:

Another shortcoming is the typical information retrieval use case where a list of objects, perhaps even all, satisfying a specific criteria are wanted. This is a key requirement of, for example, legal and patent search, literature review or even the typical chatbot questions in the ilk of “Give me a list of projects we did using foo”. To get around this some people have started to develop RAG-Fusion[5] models. Basically RAG-fusion uses a LLM to generate multiple queries for (vector) search and (reciprocal) ranks them, merges them into a final ranked list. They can be designed to iterate until all relevant passages are retrieved making them more information retrieval system like.



While this strategy can work—when it does not veer off course with irrelevant queries and fail—it is more expensive than pure RAG in both computational resource demands and in response latency. The information retrieval UX demand for a “fast” response may not be fiscally realizable as the only means to address the increased

5 See https://arxiv.org/pdf/2312.10997 and https://arxiv.org/html/2402.03367v2

inference time due to all the additional retrieval steps is to up the concurrent computation from its already ravenous demands, e.g. more accelerators (GPUs) and more, faster, data lanes (bus, switches).

Our approach, by contrast, is not only significantly more prudent with resources but it addresses things more like the way a human would try to research questions. In contrast to pure RAG implementations we don't just look to enhance the generated text response with some small text fragments (passages) found via (vector) search but view search, or more properly re-search (a recursive exploration) as the cornerstone.

The advantage of this approach is that the results can be better vetted and controlled then the text generated—removing a document in the index using RAG does not refute an assertion but removing it in search, by contrast, does not underpin the assertion (counterfactuals). RAG takes the human out of the loop and whatever biases exist in the training is propagated. The ALLMIR model puts the human back in. Another key feature is that because re-Isearch does not demand a fixed unit of retrieval but affords a dynamical, query or heuristic[6] driven, retrieval of relevant structural elements—defined during ingest either explicitly specified or implicitly derived—specified at search time, it can better retrieve relevant passages.

Since organizations might have extensive and disparate information sources; ExoChat following the ALLMIR model can also retrieve data more accurately as it can search different physical indexes defined at search time by demand on-the-fly as virtual indexes—a feature of re-iSearch—to find the relevant text. Matching to relevant documents and being able to select the appropriate sub-element is not just more accurate but also explainable and traceable.

The result of a search is also envisioned as not just the results but also a prompt for the LLM. This extends, among other features, the concept of "relevant feedback"—give me more like this— as currently implemented (sparse vector similarity) to latent dense space. This is a kind-of human centric fusion.

Since underlying engine already implements a weighted hierarchical term model (Thes) it is already enabled for domain specific semantic enhancements. At current these models should ideally be hand-crafted for each domain (which has proven a high hurdle for nearly all users) but we see it as feasible to perhaps generate these (or a good starting point) using a domain training corpus. Since a single index can support multiple search time user selected models it is not just more human centered but also more flexible then current typical semantic enhancements representative by, among others, Elastic on Lucene. A search can at runtime select suitable Thes—even on the level of individual indexes that combine to create virtual indexes (which can also be from from defined groups of indexes).

It's one thing to have a good search but it must also be economically

6 The heuristic can also impose, as needed, size constrains to allow for their back propagation into the model.

feasible, efficient and sustainable. While the standard re-Isearch powered node can run on pretty much any platform, even small embedded systems, LLM inferencing (currently) demands a bit more resources. But instead of costly GPU data-servers (where popular the NVIDIA H100 costs $30k each and consumes alone 700w and are often deployed for inferencing in 8x constellations) we aim to target accessible platforms. As initial target we're looking at Apple silicon (MacBook Pros, max 80w), NVIDIA embedded (Orin 64, max 50w) and gaming PCs (RTX-3090/RTX-4090, less efficient but still typically under 500w). These with the exception perhaps of the Orin ($2000) are comparatively common. For model tuning we plan on using QLoRA-PEFT (experimenting also with QALoRa) tuning given its smaller memory footprint.

While QLoRA is comparatively efficient in the output inferencing model it is relatively slow. The model will always have a built-in data lag (hysteresis). The seq2seq typically used in RAG too has a data hysteresis given its performance. The re-Isearch engine, by contrast, indexes extremely fast and yet be searched while data is still being added, allowing the ALLMIR combination to be both comparatively computationally efficient but also plesichronous (near synchronous) with the corpus.

By constraining computational requirement we can leverage existing hardware and adhere to our aim to eliminate the need for costly data centers and reduce the environmental impact for search. Just as with the standard version, it is intended that users will be able to host their own data under their own well defined conditions.

Much of our current low power design experimentation has been on an AGX Orin64 with LLaMa family based variants but the proposed model is not specified. A recent paper "The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits" (27 Feb 2024) has some quite promising results. We have, given its newness and their need for training (versus post-training quantization), we have not yet been able to replicate and test what might enable a reduction in the hardware requirements towards even, as the non LLM nodes, more common edge and mobile devices. We have also been exploring non-GPU approaches such as Ampere Altra SOCs (massive muti-core CPUs)—recent implementations of BitNet to even 25 cent RISC-V MCU[7] to run MNIST hint at the potential of 1.58 Bits to change the game—as well as other accelerators (FPGAs such a NPUs).

Future development roadmap: Phase II development.

In the initial MVP we plan on using the existing object model and HNSW. Something like RAG can be implemented in re-Isearch using a bi-encoder to encode queries and nodes into the desired dense vectors. This is from a performance perspective sub-optimal as vector search can benefit from a task optimized data structures and loadable modules for text2vector—much like what goes on under the hood in date, numerical and geospatial objects. We call this parallel or sub project "שמאטע" (Schmate)—pronounced SHMAH-teh which means rag in Yiddish. Goal is to provide a powerful alternative to popular outboard vector databases like FAISS or uSearch for Dense Passage Retrieval (DPR) in, among other

7 https://github.com/cpldcpu/BitNetMCU

domains, Retrieval Augmented Generation (RAG) applied to popular open source context constrained large language models. Part of the design of this project is to extend our engine without a direct dependency to adopt either ExoChat or the ExoDAO project. See our separate proposal document.

Community.

While we would delight in developers joining our ExoDAO moon-shot, our intent for ExoChat is to provide a complete solution de-coupled from a dependence upon participation and 100% useable in and of itself.

And, as always, under ExoDAO: ***100% self-contained (no hidden API demands to some cloud service) true open source with Apache 2.0 license on our software components[8] and CC BY 4.0 on documentation.***

Contact Edward C. Zimmermann. Jakob-Klar-Str. 8, D-80796 Munich

8 The underlying LLM might have different licensing. While Mistral 7B is under Apache 2.0, LLaMA 2 and 3, for example, are covered by Meta's LLAMA 2, respectively 3, Community Licenses.

Appendix: Scientific Background