ExoChat

http://www.exodao.net

ExoChat Hybrid Search: Augmented LLM IR (ALLMIR)

August 2024 Draft 1.16 // Contact Edward C. Zimmermann. Jakob-Klar-Str. 8, D-80796 Munich

Self-contained 100% open source hybrid search including a 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).

Abstract/Summary/Innovation

ExoChat Project: Develop a resource constrained "chat" tech-stack for search using both its own ALLMIR paradigm as well as an enhanced implementation following a RAG-like model. Its design focuses on safety, accuracy, efficiency and lower latency. Amongst its many innovations is a contextual scope aware multi-domain passage retrieval.

The Re-Isearch engine[1] already offers an extremely sophisticated and powerful search. We propose here not just a more "accessible" user interface to search using state of the art natural language processing following the "chat" (popularized by OpenAI's ChatGPT) paradigm—prompting instead of query expression—but also a novel approach to hybrid (LLM+S/R) information retrieval that addresses some of the weaknesses of popular RAG (Retrieval Augmented Generation) paradigm implementations, in particular with respect to its use as a search interface: where a list of objects, perhaps even all, satisfying a specific criteria are wanted. ExoChat aims to fuse chat generative models with an enhanced re-Isearch while also addressing some of the fundamental issues with the former including their reliability, data hysteresis and computational resource demands. We address some of the issues with RAG-Fusion's resource demands and latency as well as its knack for veering off course.

Since re-Isearch attempts at index time to derive both explicit and implicit document structure to enable not just search into structure but also a fully dynamic unit of retrieval, definable at search time or by heuristic, its simplifies the creation and maintenance of Dense Passage Retrieval (DPR) and provides a significant advantage for, among other applications, normative RAG. RAG implementations are typically quite sensitive to the DPR model selection for the prior segmentation of content into size constrained blobs. A key observation is that multiple retrieved passages may be in the same structural scope (paragraph, section, tag etc.) so rather than use a passage alone which may have a misaligned context we use, within model context size constraints, the contents of a relevant scope rather than just the retrieved passage object. Its utility is neither restricted to ALLMIR nor RAG and there are, of course, a myriad of other uses.

Additionally the model provides a means to provide information traceability to the augmented source as well as to better provide safeguards and governance to address the threat of adversarial injections (poison, disinformation etc.).

A focus of the project is to design for constrained computational resources to enable use on the edge or self-hosting rather than depend upon either data centers or APIs to same.

The sub-project "Schmate" extends re-Isearch with vector datatypes tuned for

1 See the talk "A lightning intro to re-Isearch" presented at FOSDEM'22.

embeddings and to provide significant performance improvements at least en-par with, for example, FAISS or uSearch.

The software shall be provided as open source (Apache 2.0 license) components (modules) for Python using internally portable C++ designed for the widest range of hardware. Given its architecture and design a port to other languages (see SWIG) is easily possible.

Background/Context

http://exodao.net

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 the participation of project re-Isearch, as a recipient of NGI funding, 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. In contrast to dominant full-text engines it can also be fine tuned post index/ingest. The engine knows the address and types of every word as well as the addresses of every field/path and their types so never needs to re-parse to retrieve relevant scopes.).

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.

Chat Problem(s):

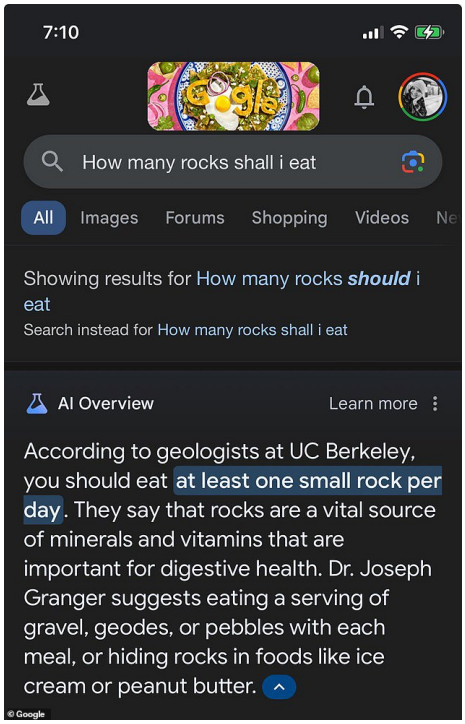
With Large Language Models (LLMs) becoming increasingly popular and prevalent for Q&A tasks, often as an ersatz for search, we had a large number of requests for ExoDAO to join the fray. A number of critical issues poked their head out:

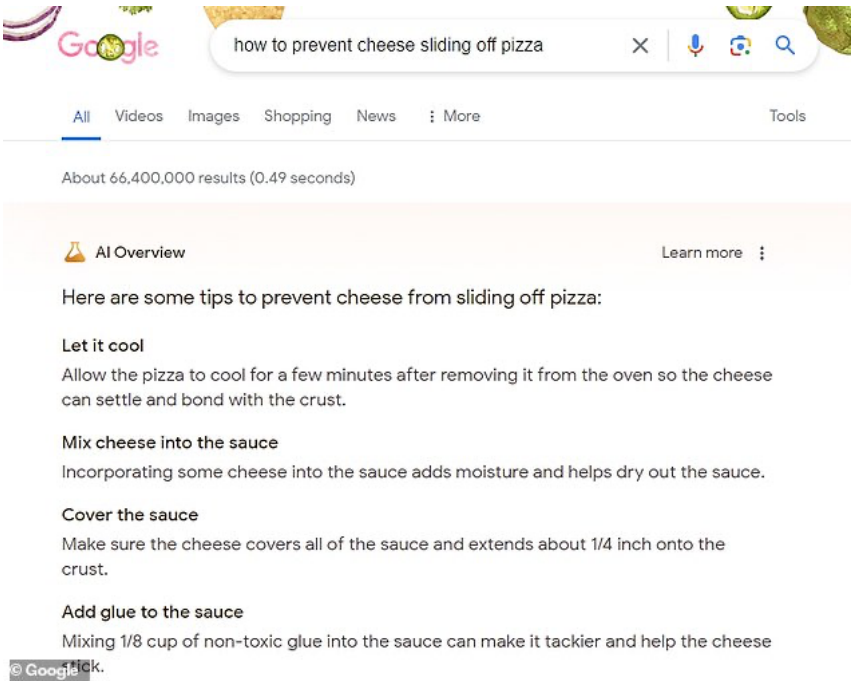
1. The normative “ChatGPT” paradigm gives just an answer. It reinforces the

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. **Without provenance, it is extremely hard to trust any answer returned from an LLMs, or any system, particularly in scenarios, be it industry or subdivision, where the answer simply must be correct.**

   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.

   When used for code generation its proven easy to hijack libraries to provide trojan alternatives—a recent case where ChatGPT suggested ace_tools (which is a library internal to OpenAI) and someone put up a

library with the same name on GitHub with nefarious intent. For civil society these are immanent dangers. 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).

4. Some recent publications have suggested that the retrieval quality of many LLM driven models and embeddings (vector dbs) implementations are often hardly better than naive keyword search.

5. They tend to, given the computational demands to update or tune, have a high level of hysteresis to the incorporation of new content.

6. 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. Augmenting foundational models with text from smaller language corpus has proven to work remarkably well—a probable side effect of language shape[2].

7. 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 to run something like GPT or Gemini (Bard) —GPT-4 has been estimated to have in excess of 1.76 trillion parameters and GPT-5 is expected to grow significantly (GPT-3 had, by contrast 175 billion and GPT-2 a comparttively meagre 1.5 billion).

8. 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.

9. 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.

10. 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.

2 "The Shape of Word Embeddings: Recognizing Language Phylogenies through Topological Data Analysis" Draganov and Skiena (2024)

11. 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.

Typical RAG challenges

While creating a prototype RAG application is these days comparatively easy thanks to sites like LangChain and HuggingFace, making it work well much less performant, robust, or scalable to a large knowledge corpus has proven for many organizations as quite difficult.

1. Ingest

(a) Data Extraction

Extracting data from diverse types of documents, such as emails, PDFs and office files such as ODF can be challenging. Documents have generally both explicit and implicit structure (text formats of what is typically called unstructured is not really without structure). The re-Isearch engine already understands these (and more than 80 base types and with these 100s more and with a plugin-in architecture easily extended to support new formats). Formats whose contents can be directly addressed are natively indexed without need for an intermediate (such as JSON or XML). Because of our algorithms we don’t have any restrictions on word length, word frequency, number of fields or paths and support a polymorphic indexing to more than 30 data types (date fields, for example, can be indexed and searched as both dates, using any of a number of detected format conventions, as well as the words or phases in its expression). Because of our design ingest is not just extremely powerful but also extremely fast and also resource efficient.

(b) Chunk Size and Chunking strategy

i. Finding the optimal chunk size for dividing documents into manageable parts for passage retrieval is generally a challenge. Larger chunks may contain more relevant information but can reduce retrieval efficiency and increase processing time. Finding the optimal balance is crucial.

ii. Most RAG system demand careful consideration on deciding how to partition the data into chunks choosing typically between size, sentence-based or paragraph-based chunking. This does not properly exploit contextual scope and often leads to misaligned context.

(c) Robustness and scalability

In order to be robust and scale one needs a modular and distributed system.

2. Search/Retrieval

(a) Because of their chunking size and strategy retrieved passages may sometimes been misaligned to context. Because we support multi-indexes, have a scope aware passage retrieval system-- we retrieve not just the chunks in the similarity top-k but view these in their contextual scope to retrieve a scope that is more inclusive of probable context.

(b) We also do, among many other things, query augmentation to help contextualize and generate accurate responses, We can also deploy query routing to the most appropriate domain (index or even virtual collection of indexes).

(c) The ALLMIR paradigm—in contrast to RAG—also brings the human into this loop.

3. **Data Security**

Solution:

While Retrieval Augmented text Generation (RAG[3]) addresses the issues of pre-training, allowing for (some) on-the-fly changes they depend upon the ability of the underlying DPR system to retrieve quality context fragments. They also do not adequately 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[4]

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.

3 Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

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

Hybrid Search Proposal:

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[5] 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.

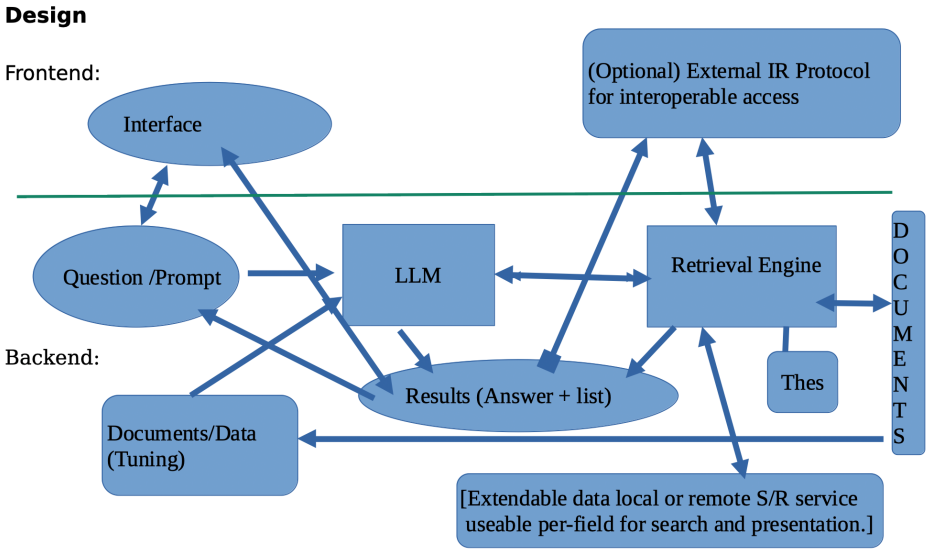
We have called this approach *Augmented Large Language Model Information Retrieval* (ALLMIR) as a human centric paradigm to clearly distinguish it from other models of purely semantic enhancement of search (typically around a text expansion model) or RAG. For the former, the engine’s (optional) term 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—one 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 since its assumed each record in an index is unique).

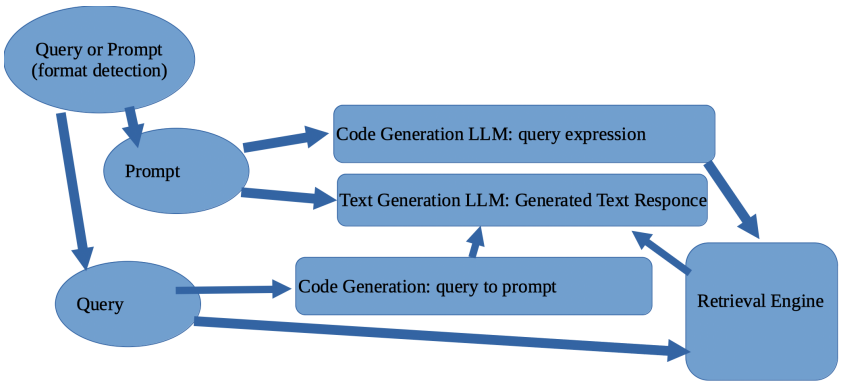
5 Given the rapid pace of development the reference is not yet specified: LlaMA3, Mistral, ...

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)

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.).



With ALLMIR the queries need not be just against embeddings but also exploit structure and the power of the underlying query language via code generation. A question like "*who said 'Out, damned spot' i*n a play from Shakespeare?" is a simple query that just needs to retrieve the speaker of the speech where the utterance is found. Even when retrieving via embedding the engine uses structure.

Why Embeddings + Word search?

While foundational LLMs have proven as not significantly better than query search, they can, however, be used as query generators to reduce user demands as well as rewriters to understand users' search intent more accurately, thereby reformulating original queries into more effective ones whence significantly improving quality. On the other hand, searching for "words" (which traditional

search view as atomic units), even exploiting their structural context, does not always produce good results. Its here where embeddings come to play. This allows concepts like ‘jumper’ and ‘knitwear’ to have very similar vector representations in their context despite differences in vocabulary and to distinguish the use of the word ‘jumper’ as a clothing item from a horse (show jumper) or a parachutist (base jumper)—situations where a thesaurus and query expansion would miserably fail.

ALLMIR vs RAG vs LC (Large Context) models:

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 and/or dialog memory.

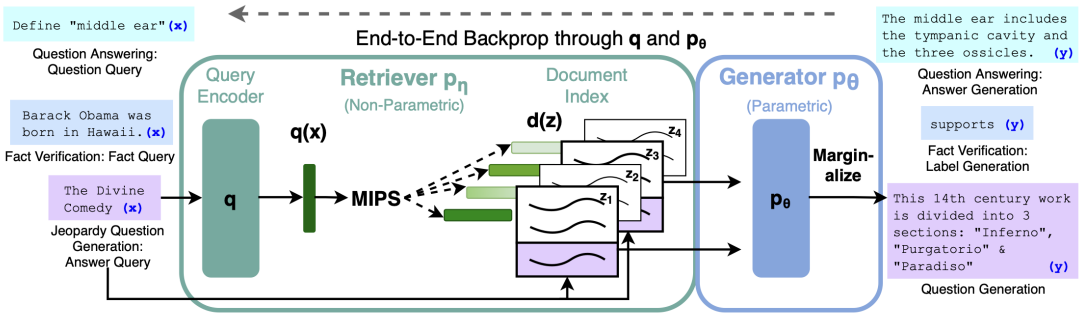
Google’s Gemini 1.5 one has a 128K context window and even, for enterprise customers, up to 1 million but it is purely API, specifically housed on Google’s platform and is also relatively energy hungry. It is also a closed AI model and developers either build on its API or via their Vertex AI platform. Leaving aside those issues, some such as ex-Google CEO Eric Schmidt has suggested these Large Context (LC) models remove the need for retrieval augmentation. Recent comparisons[6] have however shown that RAG not only significantly reduces costs but that performance can also be made comparable. Given alone the memory and computational demands of LC models and our goals to heavily constrain computational resource utilization with a particular eye towards self-hosted edge deployment they are not considered beyond use as some *external* service much like a target in a search federation (such as libraries/archives), viz. there is nothing standing in the way, if so desired, to augment things with API calls to Gemini, Claude, GPT and their ilk.

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[7] 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.

6 Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach

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

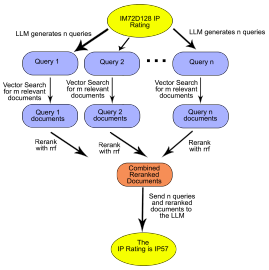


Key to the proper functioning is appropriate data chunking. By optimizing these chunks for context one can enhance the model's ability to generate coherent and contextually appropriate responses. The challenge lies in determining the optimal size for these chunks. Here lies a significant trade-off problem: 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.

ExoChat 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 fully retrieve structural elements without need to re-parse: 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. Rather than just use the chunk we can use the contextual scope—with, of course, a size contraint defineable at search time—and also, when desired or warranted provide the traceback.

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[8] 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

8 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[9] 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 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. This search-time aggregation of information from diverse sources, resp. domains, increases the likelihood of coverage. Matching to relevant documents and being able in ALLMIR 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 the 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

9 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—even the cut-price Asian market 96GB H20 variant costs $12k—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 (AGX 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 are comparatively common. We will also be looking at other platforms. 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. "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[10] to run MNIST hint at the potential of 1.58 Bits to change the game —as well as other accelerators (FPGAs such a NPUs). The recent paper "Scalable MatMul-free Language Modeling"[11] (18 June 2024) and other linear attention models illustrate a potential for a reduction in the needs for GPUs. We expect in near term to see Bitnet 1.58 linear attention models using NPUs for edge and mobile devices.

Phase II development (Milestone 3).

In the initial MVP we plan on using the existing object model and the Hierarchical

10 https://github.com/cpldcpu/BitNetMCU
11 https://github.com/ridgerchu/matmulfreellm

Navigable Small World (HNSW) algorithm (Milestone 2) for approximate nearest neighbor search for vectors (embeddings). 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 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 domains, Retrieval Augmented Generation (RAG) applied to popular open source context constrained large language models.

Because of the "curse of dimensionality" and our design to support higher dimensioned vectors we are restricted to approximate nearest neighbor (ANN) rather than k nearest neighbor (K-NN) as the later can get quite expensive. We selected to focus on graph based rather than partition-based indexes—like LSH, IVF or SCANN)— as they are both fast and may be incrementally updated.

While the HNSW algorithm is easy to implement and deploy—there is a popular C++ header based implementation—its excellent performance degrades when the sets are large and stored on disk rather than RAM. More optimal structures as passage ensembles get large must build on flat rather than hierarchical graphs. As much of our design is about constrained computation we ultimately need a number of algorithms—see, for example, Meta's FAISS.

Under the hood these are all indeed flat indexes—long used by re-Isearch for caching on disk as its fast and by exploiting sparsity (implemented by Windows NTFS, Apple APFS and most Un*x file system) relatively compact—encoding objects (here vectors) of a fixed size and so may be linearly addressed. As with the core engine itself, we shall use memory mapping to exploit the virtual memory system and shared memory rather than file I/O intro private intermediate buffers. The objects may still be compressed by LZ4 or even lossy algorithms such as PQ and there is nothing to exclude use of compressed file systems such as BTRFS—the default file system of OpenSUSE and Fedora Workstation.

Part of the architectural goals of this sub-project is to extend our engine without a direct dependency to adopt either ExoChat's ALLMIR or the ExoDAO project. See our separate project proposal document.

Impact.

The ChatGPT and Schmate sub-project offer a large number of innovations. Its multi-domain DPR model where data structure (explicit and implicit) is exploited for a better contextual match is itself a major game changer.

i. Market Opportunity

Following the 2022 launch of ChatGPT the search market was reset, catapulting AI and LLMs onto center stage.

1. Despite all the hype about RAG-LLM, the typical implementations don't really exploit the potential and often don't work significantly better than term based search. Even worse, many depend upon external APIs among others OpenAI (Azure), Vertex AI (Google) or AWS.
2. Organizations despite their desires often have real challenges developing production ready data augmented LLM systems (RAG, hybrid search).
3. Since the solution is designed to be 100% self-contained with transparent and user tunable ranking algorithms its model is designed for safety and to facilitate best practices of governance. As self-contained and without call to external APIs it can enable full privacy, especially when on the edge.
4. Privacy is an important KPI. No data is fundamentally collected or transferred opaquely to some cloud. Implementors may, however, enable query collection to enable user intra-session memory. Use cases for memory are varied but include, for example, digital therapy where the primary function of retrieval augmentation is for memory. This should use local storage—the non-LLM portion of ExoChat should run on nearly any edge device such as even mobile phones as engine ingest, indexing and search are designed for extreme resource economy (the standard re-Isearch indexer can run in as little as 8 MB of RAM and has mechanisms to oversee system consumption to maintain a low impact).
5. A better DPR using not just word semantics but also contextual scope defined through structure (explicit and implicit). This is designed to avoid contextual misses.
6. ExoChat supports search-time aggregation of information from diverse sources, resp. domains, increasing the likelihood of coverage. Matching to relevant documents and being able in ALLMIR to select the appropriate sub-element is not just more accurate but also explainable and traceable.
7. Our design is also intended to compete head-on with, among others Elastic, MongoDb Atlas and Neo4j— which are crippled by their foundation built upon Lucene (see our comparison), a rather conventional inverted index, with its modest addition of HNSW for embeddings. Elastic, despite its functional shortcomings alongside MongoDb rank as the most popular "vector" search.
8. We have strong connections to a number of open source communities including the Document Foundation (ODF, LibreOffice) as well as to PDFA. For these formats, and others, we have superior ingest. Our ODF ingest is the absolute state-of-the-art: the fastest and most accurate (created in coordination with the OASIS ODF TC). Elastic, by contrast, uses Apache Tika which just primitively extracts text content. In other formats we also have a model following Extract, Load and Transform (Extract is ingest. Load is search and the Transformation can be at presentation time and does not demand any re-parsing of documents) rather than ETL—with directly addressable formats (CSV, XML, etc.) we don't use an intermediate store format (Elastic uses for example a JSON store).

9. NOTE: The motivating moment to bring re-Isearch back was the unexpected observation upon attending ApacheCon that Lucene (and the projects around it) had hardly advanced and were still massive leaps behind IB (the former name of re-iSearch when it was a proprietary solution). Instead of inverted-indexes we had adopted Pat Arrays but long addressed their algorithmic performance weaknesses to be able to deliver speed alongside unlimited term length, no need for stop-words, unlimited fields and path...

ii. Open Source based Commercial Strategy and Scalability

All ExoChat projects—including its Schmate sub-project—are aggressively open-source in not just name. We reject “Open Source Core” as well as “Open Source” that circumvents its original intent (e.g. RedHat Enterprize, Elastic, Neo4J et al). All components within our core are provided under an Apache 2.0 or more liberal license. As part of ExoDAO, a project envisioned to provide a hyper-scale alternative infrastructure for Internet search its being designed to run outside of centralized data centers on more modest hardware—the limiting issue being the resources still demanded by the language models (which is expected to be significantly reduced within the duration of the project grant).

As ExoChat is a sub-project usable outside of ExoDAO we intend to provide, following the lead of others within the open source community, project services including bespoke customization. The re-Isearch engine already contains a plug-in architecture to support proprietary and customer specific document formats. It also provides hooks to other custom features via configuration. A similar extended plug-in design is also intended for ExoChat and its Schmate sub-project. This opens opportunities for customer driven customization beyond garden variety support.

Using a combination of Python and a minimalist C/C++ at various levels and plug-in architecture that isolates components, the skill mix to recruit and train is manageable and builds upon the prior history of re-Isearch when as IB it was a proprietary project deployed world-wide by numerous governments, NGOs, archives, publishers and multinationals.

iii. Environmental and Social Impact.

The ExoChat project, as ExoDAO, has been designed to meet the demands of energy/resource efficiency, governance, safety and plesiochronousity (minimized update hysteresis). It is about enabling a transition path away from both BigTech and energy hungry data centers.

BigTech search has contributed to reinforced filter bubbles, polarization, and a surveillance race hampering the democratization of information access. Adding to the muddle, the rise of Large Language Models (LLMs) further obscures the extent to which corporations collect and control user data. These black boxes in the cloud have introduced new challenges to the transparency and traceability of information sources. With LLMs becoming increasingly prevalent for search activities, the origin and veracity of information can often become even more obscured and difficult to vet or trace back to its source. This lack of transparency hinders users' ability to discern the credibility and reliability of the information they encounter during online searches. LLMs have also proven quite easy to lobotomize by bad-agents. The concentration of information in BigTech centralized data centers and dependency on their clusters of expensive GPUs has also resulted in an unsustainable and energy hungry infrastructure. By leveraging suitable existing hardware[12], we eliminate the need for costly data centers and reduce the

12 Simply running a PC generates between 40g and 80g Co2 per hour, Much of the available computational resources go under-used.

environmental impact.
Users can contribute their own data to their own search service (as part of some service, some Web interface, register into a federation using one of our standardised IR protocols such as SRU/W or even join our ExoDAO network as a node), minimizing web crawling and by using technology such as torrents bandwidth and network capacity can also be optimized.

Project ExoChat and its sub-project Schmate, are designed from the get-go to be computationally resource constrained and 100% self-contained and without call to external cloud based APIs. With its C++ and Python APIs (other binding languages are naturally via SWIG possible) it is also relatively easy to integrate within other products.

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[13] and CC BY 4.0 on documentation.***

13 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.

ExoChat Workplan

Work Packages

Milestone 1 We plan to start off with HNSW—see above. The re-Isearch datatype system has been from the ground up designed to handle multiple data types—it already supports some 30 polymorphic datatypes from strings to numerical to geospatial and even a number of hashes. This is what most of the direct competition such as Elastic, MongoDb, Neo4J use via Lucene. The hierarchical graph structure, while extremely efficient for search, can be memory intensive. We shall be using our memory mapping—the engine already uses this for search—to mitigate some of the RAM demands, especially when multiple process instances are run, but as the number of connections per node increases this will still hit performance as ultimately we'll need more RAM or have increased page faults. This is why we will need/want to implement and extend to support other algorithms using a flat graph structure (parallel task sub-project Schmate).

Architectural Design Phase (90% Completed)

This is to a great extent already completed. The basic model for datatypes has been designed to handle new datatypes as part of a polymorphic type model. Just like hashes HNSW is just another type with its own methods for search. The design, of course, may need tweaking and extending as new use cases emerge but the current model is expected to be wholly sufficient. It has been rigorously tested and long term deployed in large scale. The design idea for the plug-ins too have been rigorously tested and long term deployed on large scale. The only difference shall be the class method layout but our design allows for versioning here to allow for future modifications as needed. The needs for data traceability etc. have also been explored and developed within the 40F.ai project—see Ed's talk at FOSDEM'24 this past Feb in Brussels.

Research and Development Phase (in progress and continued):

• Explore algorithms and models for disk based approximate nearest neighbor search—see Milestone 3.
• Perform extensive testing and optimization to ensure the performance, accuracy, and robustness of the algorithms.
• Track developments on smaller LLM models especially MatMul-free and BitNet 1.58 models. The primary computational resource bottleneck at this time is the demands of the LLM.
• Track literature and market/product developments.

Prototype development Phase

• Build functional prototypes of the core engine extended with HNSW—but with a design towards extending in Phase II to other data structures.
• Conduct extensive testing and performance evaluation of the prototypes using real world data.
• Gather initial user feedback and iterate on the design,

Milestone 2 **Minimum Viable Product (MVP)**

• Dense relevant feedback (like freeform but embeddings) in engine.
• Build functional and useable system with plugins for embeddings

and our enhanced DPR and well as all the key features.
• Documentation
• Robust testing.
• Release into the wild.

Milestone 3

Schmate (Parallel sub-project so development can run concurrently to other work group tasks above).

• Extend re-lsearch to support additional optimized structures and algorithms for vector (embeddings) search. Goal is to provide a powerful alternative to popular vector databases like FAISS for Dense Passage Retrieval (DPR) in, among other domains, Retrieval Augmented Generation (RAG) applied to popular open source context constrained large language models.
• Extend the datatype management sub-system to support these at-will data structures as well as provide additional call-backs. The engine currently is prepared for 8 data-type callbacks (callback, local1-7). These are defined in the class FIELDTYPES's datatypes enum and handled by the indexer class in a call to a init a DB_CALLBACK class which would manage these datatypes (see index.cxx). We'd use this for plug-ins similar to the dynamic shared plugin design we use to extend available doctypes beyond the core —this works on platforms with shared libraries (including Unix variants, IOS and Windows).

Optional (selection)

Thes (enhanced weighted hierarchical thesaurus sub-system) Tools

Dependent upon budget and in response to feedback

The current Thes sub-system requires crafting for domain. Because of this few organizations—in its previous life as a proprietary system—ever used it beyond for a handful of words and phrases. By using our embeddings and a trained domain corpus as well as the indexed corpus we can foreseeably generate these to reasonable effect—or at least get quite close to a good model for term based semantics.

Standalone RAG

A more conventional RAG search (pure Q&A model). This, of course, would use the vastly improved DPR paradigm implemented in the project (exploiting explicit and implicit scope).

Webfront

Create a Web interface—the MVP is program interface (Python) and command line tools.

Integrate into ExoDAO

Expand and extend ExoDAO interfaces (including Web front-end) and protocol to support ExoChat.

Bootstrap

Create a sample tuned-model + external data corpus as a useful functioning showcase. There are a number of use cases up for consideration including European Court of Human Justice, News (RSS feeds),...

Parallel to development, testing and refinement.

Continuing Activities

Community Engagement

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.

• Cement a strong community presence through active engagement, communication, and collaboration platforms. Continued active presence at Open Source conferences such as FOSDEM (we presented talks relevant to this project at ‘22 and ‘24). Present at meetups (among others DataGeeks Munich, Search Engines Amsterdam and Data Science Belgium).
• Leverage our deep connections to ETH (Zurich) and TUM (Munich) as well as to Open Data Switzerland and Open Knowledge Foundation Germany.
• Encourage developers and researchers to contribute to the project through our absolutist open-source model.
• Organize online workshops for those interested.

Partnerships and Integration:
(*Note: we are already in discussion with a number start-ups in a consulting capacity. Edward will also be teaching at a number of organizations in Europe on the topic over the coming months. First course shall be in Frankfurt/M to a leading German public health insurance provider in Mid-Sept*).

• Identify potential partnerships with IT specialists, information federations, libraries, archives, and other relevant organizations.
• Collaborate with partners to integrate with existing information systems, networks, and data sources towards their participation in ExoDAO and not just the deployment of ExoChat or Schmate for DPR in RAG.

Appendix: Scientific References

1) Project re-Isearch Summary Sheet, Edward Zimmermann (2022)
2) A new approach to text searching, Baeza-Yates & Gonnet (1992)

3) Fortify AI against regulation, litigation and lobotomies, Edward Zimmermann (2024)
4) Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. Hubinger, Denison et al. (2024)

5) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
6) The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits, 27 Feb 2024
7) "Scalable MatMul-free Language Modeling