IdunAI: A Self-Learning, Real-Time Web3 Analytics Engine

IdunAI Team

Abstract

Generative AI systems like ChatGPT and emerging "operator-based" agents (e.g., Manus) have ignited mainstream interest in artificial intelligence, and the release of **DeepSeek R1** in early 2025 further renewed public enthusiasm[12]. However, these general-purpose assistants lack the specialized knowledge and real-time data access needed for the rapidly evolving Web3 domain. This paper introduces **IdunAI**, an AI-powered search and question-answering engine tailored to blockchain and Web3 knowledge. IdunAI aggregates decentralized on-chain and off-chain data in real time and leverages advanced AI techniques—including large language models, retrieval-augmented generation, and a domain-specific knowledge graph—to deliver accurate, context-rich answers for Web3 queries. Additionally, we integrate an operator-based approach (inspired by STRIPS-like planning paradigms) that enables IdunAI to acquire new knowledge automatically. For instance, through daily browser-driven crawls by an operator, IdunAI can fetch fresh Web3 data—even from websites lacking a formal API—thereby continuously "self-learning." We outline the system's design and implementation, demonstrating how it bridges the gap between generic AI and Web3's specialized needs. IdunAI is intended to empower developers, researchers, investors, and even autonomous agents with a comprehensive platform for Web3 intelligence. We also discuss current prototype results and a future roadmap, positioning IdunAI as a step toward a "Google of Web3" that keeps pace with this dynamic field.

Introduction

The rapid growth of Web3 – encompassing blockchain networks, cryptocurrencies, decentralized finance (DeFi), non-fungible tokens (NFTs), and decentralized applications (dApps) – has led to an explosion of data and information. Unlike the traditional web, Web3 data is highly decentralized and dynamic. On-chain transactions, smart contract events, and protocol updates stream continuously, while off-chain discussions on forums and social media (especially **Crypto Twitter**) provide real-time insights and sentiment (Crypto Twitter: Understanding the Online Crypto Community). Keeping up with this fragmented knowledge landscape presents a significant challenge for researchers, developers, and investors. A *Web3 AI search engine* is needed to intelligently aggregate and interpret these disparate sources of information in real-time.

Recent developments in operator-based planning [15, 16] and multi-step autonomous AI agents (such as the Manus system [17, 18]) highlight how tool-using AI can dynamically gather data, reason incrementally, and produce outputs in diverse formats. Building on these insights, we propose an operator-based extension to IdunAI: whenever a user query demands new or site-specific data, an operator automatically "visits" relevant websites or APIs, extracts the needed information, and updates the knowledge repository. This leads to a more robust, continuous learning cycle in the rapidly-evolving Web3 domain.

Existing search engines and general-purpose AI assistants struggle to meet this need. Traditional web search provides keyword matches but lacks contextual understanding of blockchain-specific terms and data. General large language model (LLM) assistants like ChatGPT, while powerful in open-domain conversation, have notable blind spots in the Web3 domain. For instance, ChatGPT's knowledge cutoff (September 2021 for GPT-4) means it cannot access or accurately reason about developments in the past few years – including recent smart contract exploits, emerging protocols, or regulatory changes (What is the Impact of ChatGPT on Blockchain and Web3 Space?). This limitation in up-to-date data can lead to **incomplete or incorrect answers** for Web3 queries. Moreover, generic LLMs were not trained specifically on blockchain transaction data or crypto jargon, which increases the risk of hallucinations (fabricated answers) when asked detailed Web3 questions (An in-depth explanation of OKX Jumpstart's new project QnA3.AI - the AI Agent essential for residents of the Web3 world. - BlockBeats). The gap between the *real-time, specialized knowledge* demanded by the Web3 space and the capabilities of current AI tools is increasingly evident.

In fact, the public launch of **DeepSeek R1**—a free ChatGPT alternative—in 2025 briefly captivated the AI community, underscoring the interest in more capable assistants. Yet even such breakthroughs do not address Web3-specific knowledge needs. There is a growing consensus that **domain-specific AI models** or augmented systems are required to bridge this gap (In-depth exploration of ChatGPT's competitor "DeepSeek" - ChainCatcher). Early so-

lutions have emerged: OpenAI-based assistants fine-tuned on crypto data, or community-driven knowledge bases for blockchain. However, these tend to address only parts of the problem. Some focus on developer needs (e.g., coding smart contracts), others on market analytics, but few offer a comprehensive platform that covers *the full spectrum of Web3 knowledge* in an integrated way. There remains a need for a **Web3-focused AI search engine** that can seamlessly combine on-chain data, off-chain news, and context from historical knowledge – all while leveraging advanced AI reasoning to provide direct answers and insights.

IdunAI is proposed as this missing piece: an AI-powered Web3 search and question-answering engine designed to empower users and AI agents with up-to-date, relevant Web3 knowledge. IdunAI aims to **bridge the gap** by ingesting real-time Web3 data, organizing it into a rich knowledge repository, and employing state-of-the-art AI (including LLMs, knowledge graphs, and retrieval techniques) to answer user queries. The goal is to offer the **best of both worlds**: the depth and precision of a domain-specific knowledge base, and the flexibility and intelligence of a modern AI assistant. In doing so, IdunAI would enable researchers to query complex on-chain data with natural language, help developers find solutions and best practices, assist investors in tracking news or on-chain metrics, and even allow autonomous AI agents to interface with the Web3 ecosystem for data-driven decision making. This white paper outlines the vision, situates it among related efforts, and delves into IdunAI's design, current progress, and future roadmap.

Related Work

General AI Assistants and Search Engines

Several general-purpose AI tools illustrate the current state of conversational search and question-answering, but each has notable limitations in the Web3 context:

- **ChatGPT (OpenAI):** ChatGPT is a leading LLM-based assistant with broad knowledge and fluent dialogue. However, its lack of real-time data access and Web3-specific training limits its usefulness in this domain. Its knowledge cutoff (2021) means it is unaware of recent blockchain developments[3], and it is not fine-tuned on blockchain data structures. As a result, ChatGPT often gives incomplete or outdated answers to complex Web3 queries.
- **Grok (xAI):** Grok is a newer general chatbot positioned as a competitor to ChatGPT. Its distinguishing feature is integration with X (formerly Twitter) for real-time information, allowing it to pull in the latest tweets[5]. This connectivity helps with timely Web3 news or trends emerging on crypto social media. However, Grok lacks a dedicated blockchain data pipeline or deeper analytical capabilities, so it still shares many of ChatGPT's limitations in understanding on-chain data.
- **DeepSeek:** DeepSeek is an emerging AI search engine touted as a domain-specific alternative to ChatGPT[4]. It emphasizes accuracy by leveraging specialized datasets and real-time learning for particular industries. DeepSeek illustrates the importance of context-driven responses using domain-specific knowledge. However, as of now it does not specifically target Web3, serving more as an example of the benefits of domain tuning in AI.
- **Perplexity AI:** Perplexity is a conversational search engine that integrates web search with an LLM to answer user questions, providing cited sources. It can retrieve information from the live web (e.g., news articles) to answer general queries. For Web3 questions, however, Perplexity faces two challenges: (1) much of the granular Web3 data (transactions, mempool, contract state) is not readily accessible via simple web search; (2) without blockchain-specific training, the LLM may misinterpret technical on-chain information it finds (for example, raw data from a block explorer). Thus, while Perplexity bridges some timeliness gaps, its understanding of Web3-specific content remains limited.

In summary, general AI tools offer useful features (fluent language understanding, broad knowledge, some web access), but **fall short in Web3** due to lack of up-to-date data integration, insufficient training on blockchain-specific knowledge, and inability to directly process on-chain datasets. This has motivated the development of Web3-specialized solutions.

Web3-Specific AI and Knowledge Tools

Recognizing the shortcomings of generic AI in the crypto domain, several projects have emerged that specifically target Web3 data and queries:

- **0xScope (ScopeChat and Web3 AI Data Layer):** 0xScope positions itself as the first *Web3 AI Data Layer*, establishing standardized methods for collecting and managing on-chain and off-chain data[6]. On top of this foundation, 0xScope offers products like *ScopeChat* (an AI assistant for crypto queries) and *ScopeScan* (an analytics tool), all leveraging a curated, structured data repository. Notably, 0xScope introduces **entity abstraction**, clustering blockchain addresses to identify the underlying entities (exchanges, whales, protocol contracts) behind them. This enables queries about an entity's activity across addresses and chains. 0xScope primarily focuses on trading and portfolio insights, showing the value of a purpose-built Web3 data pipeline, though its scope remains largely finance-centric.
- **Alva (Alva.xyz):** Alva, launched in late 2023, is described as an "AI copilot for Web3 research and exploration"[7]. It provides a Web3-focused conversational chatbot that users can query for project information, comparisons, real-time news updates, airdrop alerts, and token price analytics. Alva maintains an evolving knowledge base and even envisions becoming a decentralized community-driven knowledge protocol (an "AI-powered Web3 wiki + chatbot"). It excels at user-friendly research assistance and breadth of coverage, but as a proprietary platform with a closed database, its handling of deeply technical queries or direct on-chain data is unclear. In comparison, IdunAI aims to be an open and

extensible search engine, providing API access and transparent reasoning for complex queries, to serve a broader range of Web3 needs.

- **QnA3:** QnA3 is one of the most ambitious projects in the Web3 AI space, branding itself as a *"next-generation AI-driven Web3 knowledge engine."*[1]. Launched in early 2023, QnA3 quickly gained traction by deploying an AI Q&A chatbot (e.g., via Telegram) and offering on-chain data exploration features. It uses a *Retrieval-Augmented Generation (RAG)* approach to combine an LLM with a vast Web3 knowledge base[13], significantly improving the relevance and accuracy of answers by retrieving up-to-date information and reducing hallucinations. QnA3 also provides an **intent-centric trading assistant**, an AI agent that can help execute user-defined actions in the Web3 domain (such as trading or asset management based on natural language prompts). It integrates with decentralized identity solutions and has introduced a native token to unlock AI features and for governance[11]. In essence, QnA3 represents a comprehensive Web3 AI ecosystem blending knowledge retrieval with direct action. IdunAI shares some of QnA3's technical approaches (e.g., leveraging knowledge graphs and vector search) but is oriented toward open access and serving as a flexible Web3 knowledge service, rather than a closed trading platform.

Competitive Analysis: All these related works highlight a qualitative trend: **blending AI with Web3 data yields better outcomes** for users than either approach alone. Each has a unique angle – 0xScope leans into data infrastructure and trading signals, Alva focuses on research assistance with real-time news, and QnA3 builds an all-in-one knowledge and trading agent. Yet, none has become a ubiquitous "Google of Web3" yet, indicating that there is room for innovation and improvement. IdunAI distinguishes itself by aiming to combine the strengths of these approaches into a unified system: real-time data ingestion (like Grok or Alva) across diverse sources, rigorous data structuring and semantic search (like 0xScope's data layer or QnA3's RAG), and advanced reasoning for complex Q&A (with techniques such as chain-of-thought prompting [10], which none of the current products explicitly advertise). Moreover, IdunAI is envisioned as an **enabler for others** (AI agents, enterprise tools, research analysts), not just an end-user chatbot. By providing robust APIs and focusing on *trustworthy, transparent answers* (with citations and reasoning steps), it can fill an important niche in the Web3 landscape: a go-to intelligent search engine that stakeholders can rely on for accurate, timely Web3 information.

System Design

IdunAI's system design is centered around an **end-to-end pipeline** that spans data acquisition, knowledge base construction, query understanding, and answer generation. The architecture integrates multiple components – each optimized for handling the challenges of Web3 data and user queries – into a cohesive engine. Figure 1 illustrates the high-level workflow of IdunAI, from data ingestion to answer delivery.

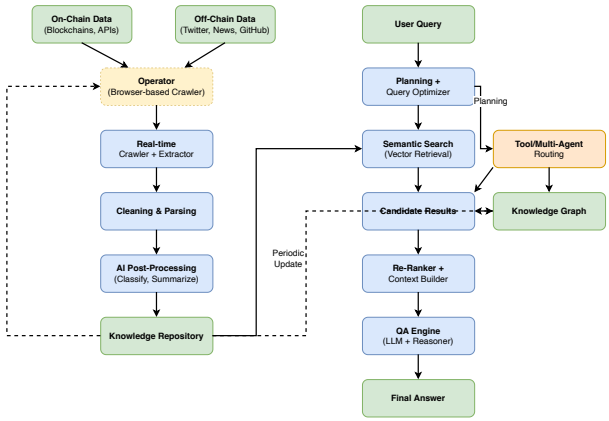

Figure 1: High-level system workflow in IdunAI, from data ingestion to answer delivery. On-chain and off-chain data feed the repository, which is then queried by users via the semantic search and QA engine.

Data Collection & Preparation

IdunAI continuously gathers data from a variety of **Web3 data sources**, ensuring that its knowledge repository stays current. The data collection strategy covers both **on-chain data** and **off-chain information**:

- **On-Chain Data:** This includes blockchain transaction data, smart contract events, and state information from multiple networks (e.g., Ethereum, BSC, Polygon, etc.). IdunAI can interface with blockchain nodes or third-party APIs to stream blocks and transactions in real-time. For efficiency, targeted data collection may be employed – for instance, focusing on certain contracts or topics of interest (like major DeFi protocols) for deeper monitoring. Over time, the system will aggregate a wealth of on-chain facts, such as token transfers, contract creations, governance votes, and more. These are typically stored in structured form (tables or JSON) within the knowledge repository, and also indexed for search. On-chain data often needs decoding (e.g., interpreting input data to function calls, labeling addresses with known entities) which is part of the preparation process.
- **Off-Chain Data:** Equally important is capturing the discourse and context around Web3. This includes social media (Crypto Twitter posts, influential threads), news articles, blog posts and whitepapers, developer documentation, Q&A forums (e.g., StackExchange for Ethereum), and research reports. IdunAI's current implementation focuses heavily on **X (Twitter)** given its role as a real-time news feed for crypto (Crypto Twitter: Understanding the Online Crypto Community), but will expand to other sources such as Reddit, Discord (for major project announcements), Medium blogs, and official project websites. Off-chain data gives necessary background – for example, a sudden on-chain spike in trans-

actions might be explained by a news announcement or a social media rumor. The system's **crawler** uses APIs or web scraping to fetch this content in real-time or at regular intervals.

Once raw data is collected, it undergoes a **cleaning and pre-processing** stage. This involves parsing the data into a standardized format, removing duplicates or irrelevant information, and enriching the data where possible. For text content (like tweets or news), cleaning might remove HTML tags, code blocks, or excessive formatting. For blockchain data, parsing includes decoding hexadecimal fields, identifying token symbols from addresses, etc. The cleaning step ensures that subsequent AI processing deals with normalized, readable data.

An **AI-based post-processing** then enhances the raw information. IdunAI leverages a specialized language model (e.g., a fine-tuned mini GPT-4 model, referred to as `gpt-4o-mini`) for tasks such as classification, summarization, and translation. For instance:

- **Classification:** Every piece of content can be tagged with relevant categories (e.g., "DeFi exploit", "NFT marketplace update", "regulation news", "technical Q&A"). This taxonomy helps in filtering and targeting search results. The classifier can also identify mentions of key entities like project names or token tickers, which will later feed into the knowledge graph.
- **Summarization:** Long-form texts (multi-thread tweetstorms, lengthy blog articles, forum discussions) are summarized to capture the main points. Summaries make it easier for the QA engine to digest content and also allow storing a concise version in the knowledge repository for quick reference. For example, a 20-tweet thread discussing a protocol upgrade can be reduced to a few sentence summary highlighting the essential details.
- **Translation:** The crypto community is global, with significant discussions in languages other than English (Chinese, Spanish, Russian, etc.). IdunAI can translate non-English content into English (or a common representation) for indexing, ensuring that valuable information isn't missed due to language barriers. Conversely, answers could be translated to the user's preferred language if needed.

All cleaned and enriched data is then stored in the **knowledge repository**. This repository acts as the central storage of IdunAI's world knowledge. It is not a simple text index, but a composite store consisting of: (a) a document database containing raw and summarized content with metadata (source, timestamp, tags), (b) a vector index for semantic search (described below), and (c) a growing **knowledge graph** of Web3 entities and their relationships. The knowledge repository is updated in near real-time – as soon as new data comes in and is processed, it's made available for queries. By design, the repository also retains historical data (e.g., past states, old news) which can be crucial for queries that ask about changes over time or past context (for example, *"What were the major exploits in DeFi in the last year?"*).

The result of this pipeline is a **rich, machine-understandable knowledge base** of Web3 information, continuously curated and ready to be queried by the IdunAI engine. By handling the heavy lifting of data crawling and cleaning, IdunAI ensures that users (or AI agents using it) can focus on asking questions and getting insights, rather than manually gathering data from dozens of disparate sources.

Query Optimization

When a user (or an AI agent) poses a question to IdunAI, the system first passes it through a **Query Optimizer** module. The goal of query optimization is to interpret the user's intent, refine the query for effective retrieval, and route it appropriately within the system. This step is especially important in the Web3 context due to the variety and complexity of queries possible.

Key functions of the Query Optimizer include:

- **Temporal Filtering:** Many Web3 questions have an implicit or explicit time dimension. For instance, *"What is the current TVL of Uniswap?"* implies a query for the latest data point, whereas *"What was the NFT trading volume in Q1 2023?"* has a historical range. IdunAI's optimizer can inject time constraints based on query wording. Currently, a partial implementation sets a default lookback window (e.g., 180 days) for queries that don't specify a timeframe, to prioritize recent data and improve performance. The system supports a minimum granularity of 0.1 days (≈ 2.4 hours) for time filtering in searches, allowing precise slicing of data if needed. This ensures that results are not polluted with outdated information when the user clearly cares about recent context (or vice versa).
- **Synonym & Terminology Handling:** The crypto domain is notorious for jargon and evolving terminology. The optimizer will leverage a dictionary of synonyms and equivalents to expand or translate the query. For example, a user might ask, "Has **Eth2** staking withdrawal been enabled?" The optimizer knows "Eth2" refers to the Ethereum 2.0 upgrade (now just part of Ethereum post-Merge) and that "staking withdrawal" relates to the Shanghai upgrade. It can reformulate or tag the query with these related terms to cast a wider net on retrieval (including "Shanghai upgrade" or "ETH staking unlock"). Similarly, ticker symbols (BTC, ETH) can be expanded to full names (Bitcoin, Ethereum) or vice versa, and slang like "rug pull" can be recognized as "scam exit". This kind of query expansion will improve recall of relevant results.
- **Intent Classification & Routing:** The module also classifies the type of query. Web3 questions can range from factual (e.g., *"What is the block time of Solana?"*), to analytical (*"Compare the decentralization of Binance Smart Chain vs Ethereum."*), to actionable (*"Show me arbitrage opportunities between DEXes now."*). By classifying intent, IdunAI can route the query to the appropriate internal handler or model. A factual query may trigger a straightforward search in the knowledge base,

whereas an analytical one might prompt the system to gather multiple pieces of information and then use a more powerful LLM reasoning step to synthesize an answer. In future iterations, some intents might be served by specialized subsystems (for example, a numerical data query could invoke a small algorithm or an on-chain query executor). Another exciting direction is the **integration of operator routines with Web3 wallets**. By pairing an operator-driven browser approach with on-chain wallet access, IdunAI could autonomously execute certain Web3 interactions (e.g., signing transactions, initiating governance proposals) under user-defined policies. This synergy would enable advanced "one-click" or even fully automated workflows: from knowledge retrieval to direct smart contract interactions.

- **Query Rewriting:** In some cases, the optimizer may rewrite the user's question for clarity or searchability, without changing its meaning. For instance, a user question: "What are people saying about $XYZ rug?" could be rewritten as "What are social media discussions regarding the XYZ project being a rug pull (scam)?" This formalizes slang and symbols into more standard language for the search phase. The original phrasing can still be preserved for answer generation to maintain conversational tone, but the internal retrieval might benefit from the rewrite.

Overall, query optimization acts as the **brainstorming and setup phase** before diving into the knowledge repository. It ensures that the subsequent semantic search is both efficient and comprehensive in fetching the pieces of information that will likely answer the question. This component is continually improved as the system learns from past queries (e.g., which rewrites led to good answers, which did not). As IdunAI's user base grows, the optimizer could also personalize itself – for example, recognizing if a certain user often inquires about developer documentation and tailoring the search accordingly.

Semantic Search

After the query is optimized and understood, IdunAI performs a **semantic search** through its knowledge repository to retrieve relevant information. Unlike traditional keyword search, semantic search uses the meaning of the query and documents, not just exact word matches, to find the most relevant pieces of content. This is crucial in IdunAI for dealing with the nuanced and varied expressions in the crypto domain.

The semantic search module is built on a combination of **vector-based retrieval** and knowledge graph filtering:

- **Vector-Based Retrieval:** Each document or data item in the knowledge repository (tweets, paragraphs from articles, transaction records, etc.) is converted into a high-dimensional vector embedding at the time of ingestion. This is done using a pre-trained embedding model. IdunAI currently utilizes OpenAI's `text-embedding-3-small` model to encode textual data into embeddings, which capture semantic similarity (so that two texts about the same topic end up near each other in vector space). The embeddings are stored and indexed in a vector database – in our implementation we use **Milvus**, a scalable open-source vector DB (Milvus | High-Performance Vector Database Built for Scale). Milvus allows fast nearest-neighbor search even over millions of vectors, making it ideal for retrieving relevant snippets given an embedding of the query. When a user query comes in, we embed the query in the same vector space and ask Milvus for, say, the top N most similar content pieces. This finds information that may be phrased very differently yet is conceptually related to the query. For example, if the query is "DAO treasury management best practices," the relevant content might not contain those exact words but might discuss "governance proposals for managing DAO funds" – the embedding similarity can capture this connection where keyword search might fail.

  To formally describe how we measure similarity between the query embedding $\mathbf{q}$ and a document embedding $\mathbf{d}_i$, we use the cosine similarity:

  $$\text{sim}(\mathbf{q}, \mathbf{d}_i) \;=\; \frac{\mathbf{q} \cdot \mathbf{d}_i}{\|\mathbf{q}\| \, \|\mathbf{d}_i\|}.$$

  The system retrieves top N documents by this similarity score. Then a neural re-ranker f_{rerank} produces a final relevance score r_i for each candidate:

  $$r_i \;=\; f_{\text{rerank}}\big(\mathbf{q}, \mathbf{d}_i\big).$$

  Here, f_{rerank} is a cross-encoder model that refines the ranking based on deeper query-document analysis.

- **Knowledge Graph Integration:** While vector search is powerful, purely similarity-based results can sometimes miss the mark if not guided, especially in a domain with many intersecting topics. This is where the **Web3 knowledge graph (KG)** comes into play. IdunAI's knowledge graph is a growing network of entities (projects, tokens, people, addresses, protocols) and their relationships (e.g., *Project A is built on Blockchain B*, *Address X belongs to Entity Y*, *Developer Z founded Project A*). By leveraging the KG, the search process can be made more precise. For instance, if a query specifically mentions an entity in the KG, we ensure that results involving that entity (or closely related ones) are prioritized. The system might first look up the entity in the KG to get context – say the query mentions *"Uniswap v3"*, the KG tells us this is a protocol on Ethereum, related to the *Uniswap* entity, and has concepts like *Liquidity Pools*. This knowledge can filter or re-rank the vector search results to those that are indeed about Uniswap (the protocol) as opposed to unrelated uses of the word. Moreover, if a query involves multiple entities or a relationship (e.g., "How does Compound interact with Uniswap?"), the KG can help identify that Compound and Uniswap are both DeFi protocols on Ethereum and perhaps find an intermediate link (maybe *Compound uses Uniswap's price oracles* – an actual relationship that could be in the KG). The KG thus injects a layer of **symbolic reasoning** into the otherwise neural retrieval, ensuring factual relevance and coverage of all facets of the query.

We model our knowledge graph as a directed labeled graph $\mathcal{G} = (V, E)$, where V is the set of entities (e.g., `Project`, `Token`, `Person`, `Address`), and E is the set of edges (relationships). Each edge (v_i, r, v_j) carries a relation type r (e.g., *deployed_on*, *founded_by*), linking entity v_i to v_j. For instance:

$$v_i = \text{Compound}, \quad r = \text{has\_token}, \quad v_j = \text{COMP}.$$

An optional binary adjacency matrix A can represent the presence of a relationship:

$$A_{ij} = \begin{cases} 1, & \text{if there is a relation from } v_i \text{ to } v_j, \\ 0, & \text{otherwise.} \end{cases}$$

Storing these relations allows us to perform entity-centric lookups and reason about connections across Web3 protocols, tokens, and addresses.

- **Hybrid Search Strategy:** In practice, IdunAI uses a hybrid of lexical and semantic search. While the primary method is semantic (vector-based), sometimes a quick keyword filter is applied to narrow down the candidate set for embedding search. This is useful for efficiency and precision. For example, if the query includes a rare term (like a contract address or a specific error message), we first retrieve documents containing that term, then rank them by semantic relevance. Conversely, for very broad queries, we rely more on semantic similarity to surface relevant sub-topics.

The outcome of the semantic search stage is a set of **candidate results** – these could be text snippets, data points, or entire short documents that potentially hold the answer or parts of the answer to the query. Typically, more results (e.g., top 10-20) are retrieved than will ultimately be used, because the next stage will perform a more fine-grained ranking and selection.

Notably, IdunAI also incorporates a **neural re-ranking** step after initial retrieval. Using a cross-encoder model, the candidate results are re-scored with respect to the query (jina-reranker-v2-base-multilingual - Search Foundation Models). This re-ranker model takes a query and a candidate text as input and yields a relevance score based on a deeper analysis (it can consider the context of the entire text, not just a vector similarity). The cross-encoder is more precise but too slow to run on every document in the database, which is why we apply it only on the top-N from the fast vector search. The effect of re-ranking is a cleaner, more relevant set of information for the answering phase – it helps filter out any tangential results that the vector search might have included, ensuring that the most on-point facts and statements are considered.

By combining embeddings, a knowledge graph, and re-ranking, IdunAI's semantic search strives to achieve both **breadth and depth**: capturing all relevant information even if phrased differently, while homing in on the specific answer-worthy pieces with high precision.

Question Answering Pipeline

The final stage is the **Question Answering (QA) pipeline**, where the system synthesizes an answer from the retrieved information. This stage is powered by one or more AI models and is where advanced reasoning techniques are applied to produce a useful and accurate response.

The QA pipeline in IdunAI involves several subcomponents working in concert:

- **Multi-Model Routing:** IdunAI adopts a flexible approach by utilizing multiple AI models optimized for different tasks. At the center is a powerful large language model (LLM) – such as GPT-4 or a fine-tuned variant – which acts as the primary answer generator. However, depending on the query, this LLM might call upon specialized tools or models. For instance, if the question is numerical or involves on-chain metrics (e.g., *"What is the percentage change in total market cap since last year?"*), a small Python tool or a statistical model might be invoked to calculate that number from data, rather than relying on the LLM's internal knowledge. Or, if the question is about code (like *"What does this smart contract function do?"*), the system might use a code interpreter or a static analysis tool on the provided code and then explain it. This is conceptually similar to the *tool use* in advanced AI agent frameworks (An in-depth explanation of OKX Jumpstart's new project QnA3.AI - the AI Agent essential for residents of the Web3 world. - BlockBeats), where the LLM can decide to delegate subtasks to dedicated modules (like calculators, code analyzers, etc.). IdunAI's architecture is being built with this modularity in mind, so that as new capabilities are added, the query can be dynamically routed to leverage them. In the current iteration, the pipeline routes queries through either a direct answer mode (LLM only for straightforward questions) or a retrieval mode (LLM with retrieved context for complex questions), but future expansion will increase this routing granularity.
- **Contextual Answer Synthesis:** When the relevant snippets have been fetched (and re-ranked), the QA engine compiles a **context** for the LLM. This typically consists of a prompt that includes the question and the top relevant information (facts, quotes, data points) from the search stage. The LLM then generates an answer using both its own learned knowledge and the provided context. This Retrieval-Augmented Generation approach ensures that the LLM stays grounded in actual data, reducing the chance of hallucination and allowing it to cite sources. For IdunAI, the answer is often expected to include references – for example, if the question was *"What caused the sudden drop in X token yesterday?"*, the answer might say, *"X token dropped 25% following a governance proposal hack (Crypto Twitter: Understanding the Online Crypto Community),"* including a citation to a source (like a tweet or article) that reported the hack. This not only provides transparency but also credibility, which is vital in an industry prone to misinformation.

  We can frame our Retrieval-Augmented Generation (RAG) approach as a probabilistic model:

$$p(A \mid Q) = \sum_{i=1}^{N} p\big(A \mid Q, D_i\big)\, p\big(D_i \mid Q\big),$$

where Q is the user query, D_i are the top retrieved documents, $p(D_i \mid Q)$ may be derived from the cosine similarity and re-ranker scores, and $p(A \mid Q, D_i)$ reflects the language model's likelihood of generating answer A when conditioned on Q plus the context from D_i.

In practice, we do not compute this sum explicitly for every possible A, but this formulation underscores how our system combines retrieved evidence to form a final answer with some confidence level.

Function Calling Integration. **IdunAI** also supports *function calling* mechanisms (inspired by recent LLM frameworks, wherein the LLM can dynamically invoke a registered "function" with structured arguments and receive structured outputs. For example, if the user asks, *"Please get the token price from on-chain data, and summarize in a table,"* the LLM can call an internal `get_token_price(tokenSymbol)` function, passing `tokenSymbol="XYZ"`, then automatically parse the JSON response to incorporate the price into the final textual answer. Function calling improves reliability (by ensuring correct data retrieval steps) and modularity (developers can register new functions/tools for the LLM to use).

In the current iteration, the pipeline routes queries through a direct answer mode (LLM only) or a retrieval-augmented mode (LLM with context), but **function calling** is emerging as the standardized method for bridging the LLM with specialized modules.

- **Step-by-Step Reasoning (Chain-of-Thought):** Complex queries may require the system to perform reasoning in multiple steps. For example, *"Will the recent Bitcoin ETF approval likely impact DeFi token prices?"* is a question that doesn't have a single factoid answer – it requires reasoning across events (Bitcoin ETF approval), understanding market dynamics, maybe comparing historical analogies. IdunAI's QA pipeline leverages **Chain-of-Thought (CoT) prompting** to handle such scenarios. Chain-of-Thought prompting involves asking the LLM to explicitly generate intermediate reasoning steps before finalizing an answer (Chain-of-Thought Prompting Elicits Reasoning in Large Language . . .). In practice, the system might prompt the model: *"Let's think step by step: 1) recall what happened with the Bitcoin ETF approval, 2) recall any historical precedent, 3) reason about DeFi tokens in relation. . . "* etc. By making the reasoning process explicit, we tap into the model's ability to handle more complex tasks systematically, as research has shown CoT improves performance on multi-step problems (Chain-of-Thought Prompting Elicits Reasoning in Large Language . . .). IdunAI can either show these reasoning steps to the user for transparency or keep them hidden and only show the final answer (depending on user preference). The chain-of-thought approach is akin to how a human analyst would break down a hard question, and it greatly enhances the reliability of the answer in intricate Web3 topics (like interpreting governance outcomes or technical differences between protocols).

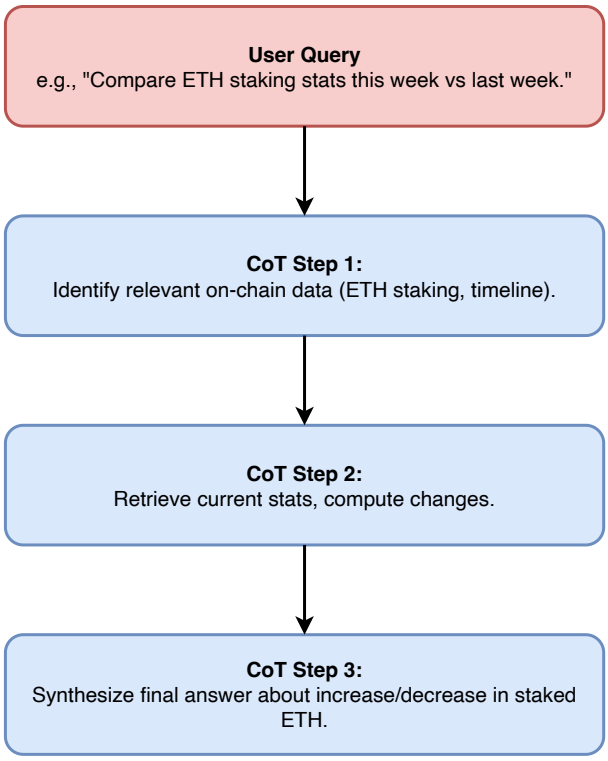

Figure 2: Illustration of chain-of-thought reasoning steps for a multi-step query. Each step refines the context and narrows down the needed data.

- **Answer Drafting and Self-Consistency:** The model drafts an answer using the context. To further boost reliability, IdunAI can employ a **self-consistency** decoding strategy[14]. This involves generating multiple answer variations by sampling different reasoning paths, and then choosing the answer that is most consistent among those attempts. This technique, per recent research, helps reduce random errors in reasoning tasks by aggregating the "wisdom of multiple attempts."
- **Answer Generation and Verification:** Once the LLM produces an answer, IdunAI can optionally have a verification stage. This could involve cross-checking key statements in the answer against the knowledge base or using another pass of the re-ranker to ensure the answer is supported by the retrieved evidence. In some cases, a second model (or the same model with a different prompt) might be used to assess the answer's correctness or completeness (a bit like how one might use ChatGPT to critique another ChatGPT's answer). For instance, an answer that includes numerical data or specific claims might be fed back into a verification prompt: *"Check if all claims in this answer are supported by the sources"*. This is an evolving area of the design aimed at maximizing answer accuracy, which is crucial for user trust.
- **Formatting and Delivery:** Finally, the answer is formatted for presentation. As this white paper is written

in Markdown, IdunAI is designed to output answers with proper formatting: e.g., lists for multi-part answers, tables for comparative analysis, and inclusion of citations in the specified format. If the user's environment supports it (like a certain chat UI or integration), IdunAI can also include links, or even simple charts if needed (though image generation is not a focus currently). The answer aims to be **concise yet comprehensive**, often a few paragraphs with references, unless the user requests more detail.

Importantly, the entire QA pipeline is built to be **interactive and iterative**. If the user finds the answer incomplete or wants more detail, they can ask follow-up questions, and the system will treat that in context (maintaining a conversation state). The multi-model architecture also means we can continually integrate improvements. For example, if tomorrow a better model for summarizing smart contract code emerges, we can plug it into the pipeline for queries that involve code, enhancing the overall capability of IdunAI without retraining the whole system.

Operator-Driven Self-Learning

While IdunAI's data ingestion relies heavily on APIs and direct blockchain connections, certain websites or platforms lack convenient data feeds. To address this, we incorporate an **operator** (an automated browser-based agent) capable of interacting with arbitrary websites. If an API is absent, IdunAI sends instructions to the operator, which navigates pages, fills forms, and collects relevant data. This data is then parsed and added to the repository, enabling continuous self-learning even in data-sparse corners of Web3.

Planning-Based Reasoning

In addition to chain-of-thought prompting, IdunAI uses **planning-based reasoning** inspired by classical operator-based planning [15, 16]. When a complex query arises—e.g., "Analyze recent DeFi exploits and compare them to NFT hacks"—the system breaks the request into sub-goals (finding exploit data, collecting NFT hack details) and may invoke the operator to fetch extra info if missing. This stepwise approach mimics how STRIPS or Manus-like agents plan actions, ensuring thorough coverage and context.

Flexible Answer Generation Formats

Previously, IdunAI returned mostly textual answers. Now, by leveraging operator-capable modules, we can generate various output formats:

- **Tables/Excel**: If numeric or tabular data is requested, IdunAI can produce a downloadable spreadsheet (e.g., CSV, XLS).
- **Images/Charts**: The system can render charts of token prices or produce a quick summary graphic and embed it in the response.
- **Web Pages or PDFs**: On demand, IdunAI's operator can compile a custom web page or PDF that aggregates curated data for more formal reporting.

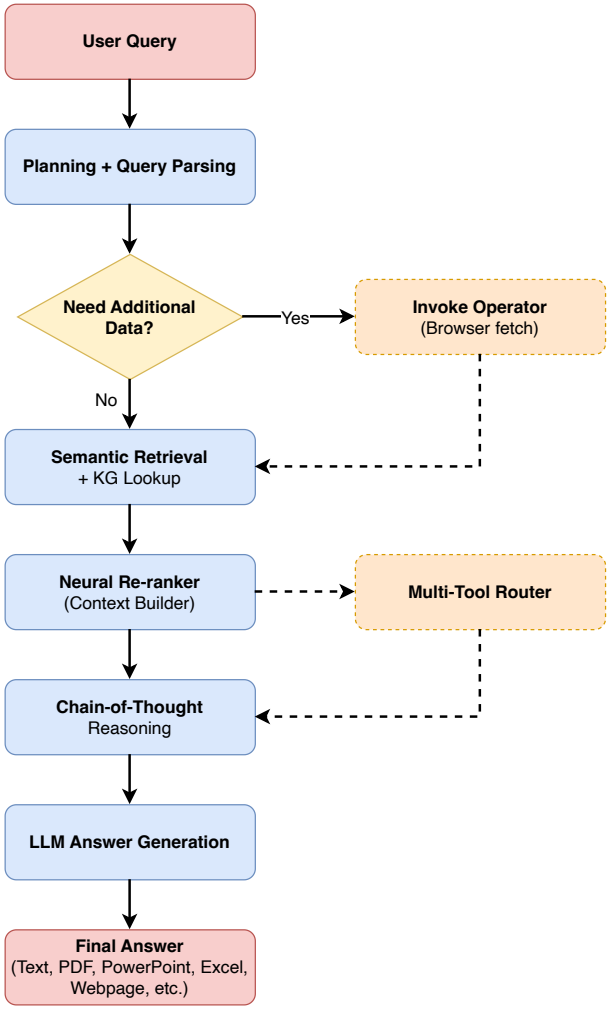

Figure 3: Overview of the query processing pipeline. After optimization and retrieval, the engine uses chain-of-thought reasoning and verification to produce a final answer.

This "answer in any modality" capability extends IdunAI's utility beyond simple chat replies, resembling advanced multi-format outputs seen in next-generation AI frameworks like Manus [18].

In summary, the question answering pipeline is where IdunAI brings together its **AI prowess and curated data** to deliver final value to the user: direct, accurate answers to complex Web3 questions, with the reasoning and evidence to back them up.

Current Work

IdunAI is under active development. As of now, several key components of the system have been implemented in prototype form, providing a solid foundation for the full vision. The focus has been on getting real-time data ingestion working and setting up the core search infrastructure. Below we summarize the current progress and capabilities:

- **Real-time Data Crawling from X (Twitter):** We have

built a custom TypeScript service that connects to X's data (via available APIs or direct fetch when necessary) to continuously collect crypto-related posts in real time. Using `node-fetch` for HTTP requests and `Prisma` as an ORM, the service stores tweets and thread data into a MySQL database. This allows IdunAI to have a live feed of what the crypto community is talking about. The crawler targets specific keywords, hashtags, and influential accounts to filter for Web3-relevant content. In practice, this means if a major announcement or breaking news hits Crypto Twitter, it enters IdunAI's repository within minutes, ready to be surfaced in answers. This real-time aspect addresses the timeliness gap observed in systems like ChatGPT (What is the Impact of ChatGPT on Blockchain and Web3 Space?), and leverages the fact that Twitter often serves as the **early warning system** for many crypto events (Crypto Twitter: Understanding the Online Crypto Community).

- **AI-based Post-Processing (`gpt-4o-mini`):** Incoming data from the crawler (tweets, etc.) are run through an automated post-processing pipeline powered by a lightweight language model dubbed `gpt-4o-mini`. This is essentially a smaller-scale GPT-4-like model fine-tuned for tasks such as summarization and tagging. It generates concise summaries of tweet threads (so a long discussion can be condensed), classifies the content (e.g., identifies if a tweet is about a hack, a new project launch, market analysis, or a meme), and translates non-English tweets to English. These steps enrich the raw data, making it far more useful for the QA system. For example, a tweet in Chinese about a partnership deal would be stored with an English translation and tags like #Partnership, #ProjectX, etc. This post-processing runs quickly to keep up with the stream of data, ensuring the knowledge base is not only up-to-date but also organized and comprehensible to the AI when searching.
- **Partial Query Optimization (Timeframe Filtering):** The query optimization module has an initial implementation that deals with temporal aspects. By default, queries are scoped to a **180-day timeframe** in the past unless specified otherwise. This means the system will prioritize recent information (about the last 6 months) on the assumption that for many questions, especially in crypto, fresher information is more relevant. It also avoids overwhelming the search with too much data (given the volume of historical data). Additionally, the system supports a minimum time granularity of **0.1 days** (˜2.4 hours) for queries that involve specific time windows or need time-series analysis. In practice, a user could ask, *"What happened in the crypto market in the first week of September 2024?"* and the system can slice that period and focus only on data from those 7 days. This partial implementation lays the groundwork for more advanced query understanding, demonstrating the system's capability to handle time-based queries and to ignore irrelevant older data when not needed.
- **Vector Search with Milvus and Re-Ranking:** The semantic search backbone is up and running. Using OpenAI's `text-embedding-3-small` model, we encode text snippets into embeddings. These are stored in a Milvus vector database instance. Milvus was chosen for its high-performance similarity search over large vector sets (Milvus | High-Performance Vector Database Built for Scale), which is essential as our data grows to millions of entries. We have integrated the search such that for any query, the system fetches the top candidates via Milvus in a few milliseconds. On top of this, we have implemented a **re-ranking** stage using the `jinaai/jina-reranker-v2-base-multilingual` model. After retrieving (for example) the top 20 candidate passages, each candidate is scored with the cross-encoder re-ranker relative to the query (jina-reranker-v2-base-multilingual - Search Foundation Models). This step significantly improves the quality of search results, as observed in testing – often eliminating off-topic results that made it through semantic search and elevating highly relevant ones even if they were slightly lower in the initial similarity ranking. The combination of **dense retrieval + neural re-ranking** is a state-of-the-art approach also seen in academic systems, validating our technical direction and already yielding accurate retrieval of information in prototype queries.
- **Knowledge Repository and Early Knowledge Graph:** All processed data resides in the knowledge repository (MySQL for structured data and Milvus for vectors). We have started constructing the **knowledge graph** in parallel. At present, the knowledge graph is rudimentary – it includes basic entity types like Token, Project, Person, and simple relations (e.g., "Token -> is part of -> Project", "Person -> founded -> Project"). This KG is being built by extracting relationships from data (for example, parsing "Alice, CEO of Acme DeFi, tweeted…" allows us to link Alice to Acme DeFi as a CEO). We are also integrating known datasets (like coin metadata, project databases) to seed the graph. While the KG is not yet fully integrated into the search pipeline, the back-end structure is being set up so that each new piece of data can update the graph (for example, if a tweet says "Project Y launched on Polygon," we add a link ProjectY -> deployed_on -> Polygon). This ongoing KG development will soon allow us to do things like entity disambiguation in queries and relationship queries (e.g., "show connections between X and Y").

In summary, the current state of IdunAI demonstrates the **feasibility and effectiveness** of the core components: real-time data capture, intelligent data processing, and semantic search. Even in this partial form, IdunAI can already answer straightforward questions about recent events (especially those discussed on Twitter or news) and provide sources. For example, a question like *"What did people say about the XYZ exploit yesterday?"* can be answered by pulling in tweets and summaries around that exploit, thanks to the live Twitter integration and embeddings search.

However, many features are still in development or not yet implemented, such as a user-facing interface, full multi-

source crawling (beyond Twitter), more complex reasoning in answers, and the richer knowledge graph integration. The progress so far provides confidence in the architectural choices, and we are iterating quickly to expand the system's capabilities.

Future Work

Moving forward, the IdunAI project has a comprehensive roadmap to evolve from the current prototype into a robust, full-featured platform. We outline the key areas of future work below, each of which addresses limitations of the current system or opens new capabilities. These enhancements will collectively advance IdunAI's mission of providing unparalleled Web3 intelligence to users and AI agents.

- **Expanding Data Sources & Reducing Latency:** A top priority is to broaden the range of data sources beyond Twitter. This involves integrating blockchain node data for multiple chains (to directly ingest on-chain events without relying on third-party APIs), adding news feeds and blogs (via RSS or web crawlers), monitoring developer platforms (GitHub commits, pull requests for major projects), and indexing Q&A forums or documentation sites. Each new source type will enrich the knowledge base and enable answering a wider array of questions (e.g., developer-centric queries from StackExchange or contract audits from GitHub). Along with source expansion, we aim to reduce the latency from data publication to ingestion. For critical feeds like security alerts or price feeds, sub-minute latency is ideal. Techniques like **webhooks and push notifications** from data providers, or running our own network nodes, can ensure IdunAI learns of new data almost instantaneously. Lower latency means users get truly real-time answers (for example, the moment a governance proposal executes on-chain, IdunAI could report its outcome).
- **Advanced Query Optimization:** We plan to significantly enhance the query understanding module. This includes building a more sophisticated **NLU (Natural Language Understanding)** component fine-tuned on Web3 query data. It would better handle complex question phrasing, multi-part queries, and colloquial language. Synonym expansion will be augmented with a comprehensive lexicon of crypto terms (constantly updated as new slang or acronyms emerge). We also intend to incorporate a **user intent learning** system that can personalize results: for instance, if a user frequently asks technical dev questions, the system will learn to prioritize developer docs in their results. Improved query embeddings or using dual-encoders (one for query, one for docs) that are fine-tuned on Q&A pairs in Web3 could further improve retrieval effectiveness by aligning the vector space with the kind of questions users ask. Additionally, dynamic **query clarification** might be introduced – if the system is unsure about ambiguous query terms, it could ask a follow-up from the user (e.g., "Did you mean X the protocol or X the token?"), rather than guessing incorrectly.
- **Full Integration of the Knowledge Graph:** The knowledge graph, currently under construction, will be fully integrated into both the retrieval and answering phases. In practice, this means for any entity mentioned in a query, the system will pull in associated information from the KG to enrich the context. For example, if one asks *"What's new with Compound?"*, the KG will tell us Compound is a DeFi lending protocol, its token is COMP, it's on Ethereum, etc., which helps ensure the search finds relevant updates (like protocol announcements, COMP token movements, related governance proposals). We will implement graph-based querying abilities, so users can ask things like, *"Which projects are backed by Alice (a certain investor)?"* or *"List lending protocols on Ethereum with TVL over $1B."* These are queries that combine structured data conditions with unstructured knowledge – a perfect use case for a knowledge graph. Integrating KG also boosts reasoning: the system can traverse connections to answer implicit questions (for instance, a query about one project's impact on another could be answered by discovering a partnership link in the graph). The KG will also be exposed via API for power users who want to run their own analysis on the data relationships.
- **Enhanced Re-ranking for Web3 Timeliness & Credibility:** While our current re-ranker improves relevance, we plan to train or incorporate a custom re-ranking model that also accounts for **timeliness and source credibility** – factors extremely important in Web3 information. For example, given two relevant pieces of content, one from today and one from a year ago, the system should usually favor the recent one (unless the query explicitly asks historically). Similarly, between a random forum comment and an official blog post, the latter might be deemed more credible. We may integrate signals such as author reputation (e.g., influencer vs new account on Twitter), engagement metrics (a tweet liked by many might be more significant), and known trustworthy sites vs. clickbait sites. A reranker that is fine-tuned on human feedback specific to "Was this answer timely and from a reliable source?" could be developed. This will ensure that not only are answers correct, but they are also *fresh and trustworthy*, addressing a common concern where outdated or dubious info can mislead users.
- **Improved Reasoning with Chain-of-Thought Prompting:** Although we have CoT prompting in the pipeline, we want to deepen the reasoning capabilities by leveraging the latest research. This may involve using **self-consistency** techniques (where the model generates multiple reasoning paths and the system chooses the most consistent answer among them) ([PDF] Self-consistency improves chain of thought reasoning in language ...), which has been shown to improve the accuracy of chain-of-thought. We might also explore **external reasoning modules**: for instance, a logic engine that the LLM can query for certain kinds of problems (like validating a sequence of transactions logically). Another angle is fine-tuning the LLM on a

dataset of Web3 reasoning tasks – essentially teaching it patterns of solving Web3-specific problems (e.g., economic reasoning in DeFi, or if-then analysis in smart contract logic). As the field of AI agents advances, we will align with best practices so that IdunAI's reasoning is not a black box but rather something that can be audited and even improved by community prompts (imagine users writing better prompts for certain types of questions which the system can then adopt).

- **Multi-Model Routing & Domain-Specific Models:** In the future, IdunAI will incorporate a broader collection of models for specialized tasks. For example, we might use a **Solidity code analysis model** for any query involving smart contract code, or a **time-series forecasting model** for questions about price trends. We will invest in fine-tuning smaller LMs on domain-specific subtasks: one for legal/regulatory questions (trained on compliance documents and law text), one for NFT metadata analysis, etc. The orchestrator will route parts of the query to these experts when appropriate. By doing this, we ensure that each aspect of a complex query is handled by the best possible model. This modular approach also future-proofs the system: as new open-source models or APIs become available (for instance, a specialized on-chain analytics AI), we can plug them in. The synergy of multiple models working together – sometimes termed an ensemble or an AI "committee" approach – can yield more accurate and robust results than any single model (In-depth exploration of ChatGPT's competitor "DeepSeek" - ChainCatcher). We will also explore **fine-tuning our main LLM** on a Web3 Q&A dataset we accumulate (user questions and our validated answers), to continually improve its baseline knowledge of the domain. In addition, we plan to standardize a **function-calling** interface where each domain-specific tool (e.g., a contract parser or an on-chain aggregator) can be invoked by the LLM through structured API calls. This framework ensures modularity—new "functions" can be seamlessly added and discovered by the main QA pipeline.
- **APIs and Agent Integration:** A core part of IdunAI's vision is to **serve as a knowledge backbone for AI agents** operating in the Web3 space. In practical terms, this means developing robust APIs and SDKs so that external applications and autonomous agents can query IdunAI programmatically. For example, a trading bot might query "what's the latest news on protocol X before executing trades," or a virtual assistant in a metaverse might use IdunAI to answer a user's question about NFT provenance. We will provide endpoints for both simple questions and more complex analytical queries, with options to retrieve raw data or a narrated answer. Security and rate-limiting will be crucial here, especially if some data or functionalities become premium. We might also explore a decentralized aspect for the API, like allowing nodes to run parts of IdunAI (for community-driven contributions, aligning with Web3 principles). In terms of agent integration, as standards like OpenAI's function calling or LangChain's agent tools evolve, we'll ensure IdunAI is accessible through those means. The ultimate goal is that any AI agent that needs knowledge of Web3 can tap into IdunAI as easily as a developer today uses an API like Etherscan – except with the added value of natural language understanding and reasoning on top of the raw data.

Conclusion

In this paper, we presented **IdunAI**, a domain-specific AI search and question-answering engine for the Web3 domain. IdunAI addresses the limitations of general-purpose assistants by integrating real-time on-chain and off-chain data with advanced AI techniques tailored to blockchain knowledge. We described how IdunAI's end-to-end pipeline (from data ingestion and knowledge base construction to query understanding and answer generation) is designed to provide accurate and timely information for Web3 queries. This approach bridges the gap between the rapidly evolving Web3 information landscape and the capabilities of current AI tools, empowering users and agents with a dedicated platform for blockchain intelligence.

Looking ahead, there are numerous opportunities to expand and refine IdunAI. Key directions include incorporating a broader array of data sources (additional blockchains, developer forums, news feeds) with minimal latency, improving natural language understanding for complex multi-part queries, and enhancing reasoning via techniques such as chain-of-thought prompting and self-consistency[14]. We also plan to integrate specialized models for tasks like smart contract analysis and time-series forecasting, and to provide robust APIs so that external applications and autonomous agents can leverage IdunAI's knowledge. These future enhancements will not only extend IdunAI's capabilities but also open avenues for research, such as developing standardized evaluation benchmarks for Web3 question-answering and exploring how knowledge graphs and neural reasoning can be combined for deeper insights.

By iterating along these lines, IdunAI aims to evolve into a comprehensive and trustworthy "Google of Web3." We hope this work inspires further exploration at the intersection of decentralized web data and AI, paving the way for intelligent systems that keep pace with the Web3 revolution.

References

[1] Gate.io Research. (2024). *What is QnA3.AI?* Describes QnA3 as an AI-enhanced search platform combining intelligent content generation with search for the Web3 era.

[2] PlasBit Blog. (2023). *Crypto Twitter: Understanding the Online Crypto Community.* Highlights Twitter's role as a real-time news feed for cryptocurrency updates.

[3] 101Blockchains. (2023). *Impact of ChatGPT on Blockchain and Web3 Space.* Notes ChatGPT's limitation of training data (up to Sep 2021), affecting Web3 applications.

[4] ChainCatcher. (2023). *In-depth Exploration of DeepSeek.* Discusses DeepSeek's domain-specific approach and real-time adaptability.

[5] SocialMediaToday. (2023). *X Updates on Grok AI Chatbot.* Points out that xAI's Grok has access to real-time posts for up-to-date information.

[6] 0xScope Team. (2023). *Web3 AI Data Layer Overview.* Introduces 0xScope as the first Web3 AI Data Layer with structured standards.

[7] Alva Team. (2023). *Introducing Alva – Your AI Companion for Web3.* Blog post on Alva's AI-powered Web3 chatbot.

[8] Jina AI. (2023). *Jina Reranker v2 Model Card.* Describes the cross-encoder re-ranking model for improved search relevance.

[9] Milvus Documentation. (2023). *High-Performance Vector Database Built for Scale.* Highlights Milvus' open-source vector database for GenAI applications.

[10] Wei, J. et al. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.* NeurIPS 2022 paper on prompting techniques.

[11] ICOHolder. (2024). *QnA3.AI Project Profile.* Notes that QnA3 introduced the QNA token for unlocking AI features and governance rights.

[12] Glover, E. (2025). *What Is DeepSeek-R1?* BuiltIn. Highlights the release of DeepSeek R1 as an open-source AI model that briefly overtook ChatGPT in popularity.

[13] Lewis, P. et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* NeurIPS 2020. Proposes RAG models that combine a parametric language model with a non-parametric memory (e.g., Wikipedia) to improve open-domain question answering.

[14] Wang, X. et al. (2023). *Self-Consistency Improves Chain-of-Thought Reasoning in Large Language Models.* ICLR 2023. Introduces a decoding strategy that boosts reasoning accuracy by sampling multiple reasoning paths and selecting the most consistent answer.

[15] Fikes, R. E., & Nilsson, N. J. (1971). *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving*. Artificial Intelligence, 2(3–4), 189–208.

[16] Hoffmann, J., & Nebel, B. (2001). *The FF Planning System: Fast Plan Generation Through Heuristic Search*. Journal of Artificial Intelligence Research, 14, 253–302.

[17] Sharwood, S. (2025). *Manus mania is here: Chinese 'general agent' is this week's 'future of AI' and OpenAI-killer.* The Register.

[18] JanusAI (2025). *Manus AI: The Best Autonomous AI Agent Redefining Automation and Productivity.* HuggingFace Community Blog.