AXIOLOGIC

LLM-Based Domain-Specific Inference Engines

Author: Sînică Alboaie, PhD, Axiologic Research
Purpose: Bring Thinking Databases to TRL 2 with ThinkyDB MVP implementation
Visibility: Public
Date: August 2024
Version: 0.3

Abstract

This report presents a pragmatic plan to validate the usefulness of the Thinking Database concept before progressing to more formalized inference engines. It provides insights into Domain-Specific Inference Engines and defines a Knowledge Domain as a distinct field of study with unique terminologies, theories, practices, and specific datasets, such as biology, economics, or computational sciences.

General-purpose Large Language Models (LLMs) appear to be reaching their limitations in the depth of inferences they can perform, primarily due to the lack of domain-specific training data for each application area and the absence of step-by-step thinking tailored to various domains. These models tend to perform inference at a fuzzy, high-level abstraction, making it challenging to achieve precise and explainable reasoning. This underscores the need for research that enables a granular approach tailored to each specific scientific, technical, or social domain. While LLMs have demonstrated an ability to perform common-sense inference, they can be more effectively trained on domain-specific tasks, incorporating customizable and explicit heuristics and modes of thinking for each field. This insight aligns with neuro-symbolic approaches, and the current report aims to concretize and elucidate various potential strategies to achieve this goal.

This report presents an initial implementation plan for ThinkyDB, the first-ever Thinking Database. Although it may not be optimal for all knowledge domains, it is designed to be effective for use cases involving text-based knowledge.

Think Databases and Domain-Specific Inference Engines

The core idea behind the "Thinking Database" concept is that it can serve as a more elegant alternative to multi-agent systems, which is one of our recent insights. The key insight is that the focus on agent “personality” and treating them as employees in a company makes sense until a point and could actually have a negative effect on the coherence of the “agents” swarm. The real issue is to focus on knowledge management, and the “Thinking Database” introduces the idea of “Knowledge Shard” as the key metaphor. In a way, by introducing the idea of a "Knowledge shard," we are mirroring the concept of a table in a traditional database—a collection of entries relevant to a specific concept.

The "Thinking Database" concept also draws inspiration from the principles in Jeff Hawkins' "A Thousand Brains: A New Theory of Intelligence," particularly the idea of using multiple models, or columns, to manage and propagate knowledge, aligning with our approach of organizing "Knowledge Shards" to function as interconnected units of understanding within a broader cognitive architecture.

Depending on the objective and the existing knowledge within the database, additional relevant knowledge can be inferred, much like how knowledge propagates in the brain based on our current understanding and goals. While we are currently at a Technology Readiness Level (TRL) 1, approaching TRL 2. There are still enough rough concepts of the concept in these initial reports, but I hope these ideas make enough intuitive sense and other research teams will start contributing or independently investigating these ideas. The Thinking Database solution emerged from real challenges we face with AssistOS in managing knowledge for agents and generating more complex scripts and documents. Our investigation indicated clearly that the multi-agent systems and chain of thought, along with various prompt management approaches, lack a generic framework for structuring both expert insights and domain knowledge, and we believe that the Thinking Database is a step in the right direction to fill this gap.

The "Thinking Database" report [TDB] introduced a new approach to knowledge management and retrieval, transcending but combining ideas from traditional databases and Embedding Databases It leverages advanced semantic search and inference techniques to unearth implicit knowledge embedded within explicit data. The volumes and complexities of data necessitate a shift from conventional keyword-based retrieval methods to semantic search strategies that interpret the intent and context of queries. This capability enhances the precision and relevance of search results, proving indispensable in fields such as natural language processing, artificial intelligence, and information retrieval.

Traditional databases often rely on direct data retrieval, primarily keyword-driven, which limits their efficacy in complex query situations where contextual understanding is paramount. In contrast, the Thinking Databases aim to integrate embeddings and inference mechanisms, allowing them to perform nuanced interpretations and generate contextually relevant insights as new implicit knowledge based on the explicit knowledge and inference rules prepared by their advanced users that prepare new AI-based systems. The proposed approach distinguishes itself by using small, energy-efficient Large Language Models (LLMs) tailored for specific domains, optimizing both performance and resource utilization. Unlike large-scale LLMs, which attempt to process extensive knowledge bases, the "Thinking Database" focuses on domain-specific knowledge, facilitating more accurate and manageable operations.

At its foundation, the "Thinking Database" is designed to perform logic-based and probabilistic inferences, integrating any type of inferring engine but looking especially at ways to use lightweight LLMs. It supports a dual-layered architecture where traditional and machine learning-based inference techniques coexist and complement each other, enabling the system to handle varied and complex data relationships efficiently. This innovative approach addresses the challenge of the combinatorial explosion in data complexity, which traditional databases and LLMs struggle to manage effectively. By focusing on domain-specific knowledge and employing manageable LLMs, the "Thinking Database" stands out as

scalable and efficient solution capable of supporting advanced decision-making processes across diverse fields. This report presents an architecture that will be used to bring the thinking concept towards TRL 2 and TRL 3. This report also delves into the concept of specialized inference engines, termed "Domain-Specific Inference Engines," designed for use in "Thinking Databases." These engines are uniquely tailored to perform inferences within narrowly defined knowledge domains, be they scientific, technical, or even artistic. The rationale behind these engines is to provide precise inferential capabilities that are directly relevant to specific fields, enhancing both the accuracy and applicability of the insights derived.

The concept of domain-specific inference is inspired by the advances in theorem proving and the modeling of formal semantics in programming languages. Theorem provers have been pivotal in mathematics and computer science, enabling the formal verification of proofs and algorithms. Similarly, formal semantics provides a rigorous foundation for understanding and designing programming languages, ensuring that language constructs have clear and predictable behaviors.

However, many knowledge domains useful for humans resist full formalization due to their inherent complexity and the nuanced nature of knowledge creation within those fields. For instance, while mathematics can be tightly formalized, disciplines such as biology or the social sciences involve variables and interactions that are less predictable and more sensitive to context.

Envisioned Techniques for DSIEs (Research Paths)

There is a growing acknowledgment of the challenges inherent in fully formalizing scientific disciplines solely on a mathematical basis. As a response, the DSIEs we envision must be designed not just to utilize formal methods such as theorem proving but also to incorporate empirical methods, including scientific experimentation and statistical analysis. This integrative, hybrid approach facilitates the validation of theoretical models with real-world data, which in turn bolsters the robustness and relevance of the inferences drawn.

In this chapter, we present several techniques that are envisioned as crucial pathways for the development of Domain-Specific Inference Engines (DSIEs). These methods are not solely theoretical constructs but are practical steps that we are currently exploring. There exists a clear degree of uncertainty regarding the computational feasibility of these approaches, reflecting the intrinsic challenges of such advanced technological endeavors. However, these ideas are promising and have the potential to significantly advance the field of domain-specific inference by integrating diverse computational methods and innovative frameworks. Acknowledging the uncertainties and embracing these exploratory paths, we aim to pioneer the creation of DSIEs that can significantly enhance the accuracy and application of domain-specific knowledge across various fields.

One family of such methods we can call the Neural-Symbolic Approach. These methods aim to leverage the strengths of deep learning for pattern recognition and symbolic AI for rule-based reasoning, creating a robust framework that supports both data-driven insights and logical deduction. This approach allows a DSIE to utilize classical logical inferences as well as multi-modal logics or fuzzy logics. It combines symbolic reasoning with probabilistic models to adeptly manage both deterministic processes—where rules are clearly defined—and stochastic processes—where uncertainty is intrinsic to the domain.

To further enhance the DSIE's capabilities, we aim to integrate both rule-based and case-based reasoning. This integration allows the engine not only to apply general domain rules but also to adapt insights from specific past cases, thereby enriching the DSIE’s contextual understanding. Such a setup is crucial for systems that need to apply learned knowledge dynamically to new but similar situations, thus improving their operational relevance and efficiency.

Another critical development is the creation of domain-specific ontologies. These ontologies define sets of terms and concepts that are relevant to particular fields and are used to enhance the semantic processing capabilities of the DSIE. By establishing a rich semantic framework, these ontologies facilitate more precise and meaningful interactions with the data, aligning domain knowledge directly with inferential processes.

We also plan to explore the fine-tuning of generic models. This technique involves using manuals, tutorials, and other domain-specific sources to fine-tune models to be more attuned to the subtleties of a specific knowledge domain. These finely tuned models can then serve as the basis for a Domain-Specific Inference Engine. While such engines strive to push the boundaries of what large language models can achieve, it is understood that there are no guarantees of flawless logic.

Lastly, a novel approach we plan to research is the use of LLMs to formalize theories by automating the detection of “facts” and “theorems” specific to a knowledge domain. These formalized facts and theorems are treated akin to axioms for ad-hoc theories, which are then tested through experiments to check if the theory is consistent or at least useful and to understand the limits of its usefulness. This method aims to elevate the status of these theorems and facts, thereby providing a foundational basis for theoretical exploration and application in practical scenarios.

Architecture of the initial implementation using an on-the-shelf LLMs

To provide more concreteness to the ideas presented in the reports on the Thinking Database and DSIEs, we have begun implementing an MVP for a Thinking Database, which we have named ThinkyDB [TKDB]. Below, we summarize its components and implementation approach.

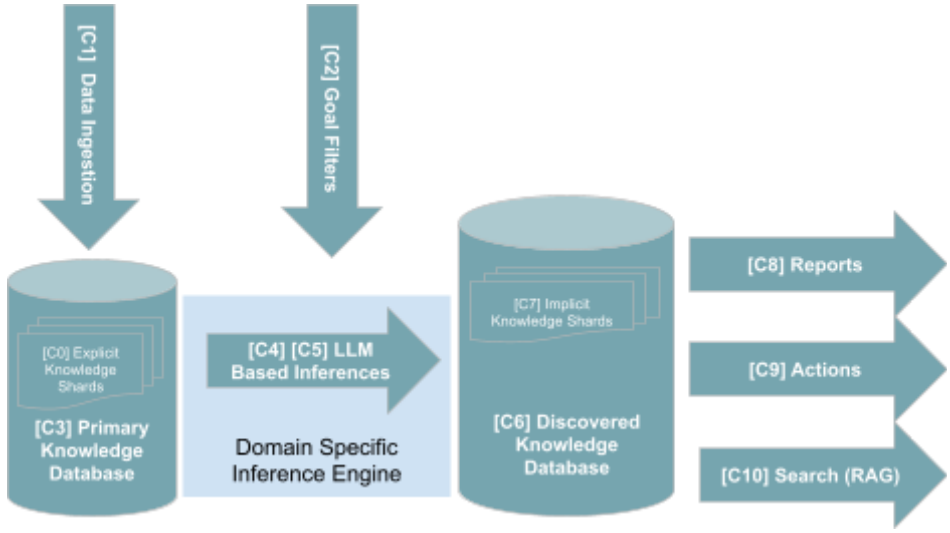

Diagram 1: Initial Components of ThinkyDB

The above diagram illustrates the key components and interactions within a knowledge processing framework designed to manage explicit and implicit knowledge shards, reasoning rules, and goal-oriented operations. Each component plays a distinct role in the ingestion, storage, processing, and retrieval of knowledge to fulfill specific goals or tasks. Below is a detailed explanation of each component and its function within the system.

The Explicit Knowledge Shard [C0] is responsible for storing “trusted” knowledge within defined knowledge domains. This knowledge is represented as a set of phrases in natural language, explicitly codified to facilitate retrieval and application. Explicit Knowledge Shards serve as the foundational repository for readily available information that can be directly referenced and utilized by the system.

These shards must contain metadata about the information source, such as the ingestion date into the system, the source URL, and other relevant details.

The Implicit Knowledge Shard [C7] contains knowledge that has been discovered or inferred rather than explicitly stated. Similarly with the Explicit Knowledge Shards, these knowledge shards also consist of natural language phrases but are derived through reasoning processes. They represent knowledge that is not directly provided but discovered through analysis, deduction, or induction. The implicit knowledge shard will contain references to knowledge from other shards and support better explanability.

Reasoning rules required for implementing a Domain-Specific Inference Engine are implemented using a combination of Large Language Models (such as “GPT-4o mini”) and custom prompts designed for various Knowledge Shards. These rules enable the system to perform complex reasoning tasks, including deduction, induction, and abduction, allowing the system to generate new knowledge and insights. These heuristics guide the ingestion process and support the creation of new explicit knowledge shards from existing implicit knowledge. The prompts for each shard (knowledge domain) will be initially generated by the LLM, but the human experts could modify and tune the rules.

The Goal Filter component [C2] acts as a decision-making component that determines whether a goal can be met using existing explicit or implicit knowledge shards. It evaluates the current state of knowledge and decides if additional explicit shards need to be created to fulfill the specified goal. The Goal Filter uses a set of predefined prompts to facilitate this evaluation process.

The Ingestion component [C1] is responsible for integrating new knowledge into the system. It can ingest information into one or multiple explicit knowledge shards and, if necessary, decide to create new knowledge shards to accommodate the incoming data. The ingestion process uses a set of prompts to ensure that the knowledge is appropriately categorized and stored.

Reporting Templates [C8] are predefined structures that organize and present information extracted from knowledge shards. They are utilized to generate reports that summarize the knowledge relevant to specific queries or tasks. These templates are based on a set of prompts that guide the selection and formatting of information for reporting purposes.

Action Instructions [C9] provide a set of guidelines or steps that outline how to perform specific actions based on the knowledge stored within the shards. These instructions include both prompts and action configurations, ensuring that the actions are aligned with the system's goals and knowledge base.

The Search APIs provide the necessary interface for interacting with the knowledge base, enabling efficient access and retrieval of relevant information. These APIs facilitate several key functionalities:

- Setting Report Templates: APIs allow for the configuration and customization of report templates, ensuring that information is presented in a structured and user-friendly manner.
- Goal Setting: APIs support the definition and adjustment of system goals, which guide the retrieval and generation of information from the knowledge base.
- Report Retrieval: Search APIs enable users to extract specific reports based on predefined criteria, utilizing both explicit and implicit knowledge shards.
- Keyword-Based Search (RAG): APIs support the implementation of Retrieval-Augmented Generation (RAG) by allowing for keyword searches that retrieve relevant content to augment the response generation process.
- Explanatory Queries: APIs provide the ability to issue queries that require explanation or contextual understanding, leveraging the reasoning capabilities of the system to offer detailed responses.

The initial implementations of the Thinking Database knowledge processing framework plan to utilize an LLM-based Domain-Specific Inference Engine (DSIE), which incorporates customisable reasoning capabilities tailored to specific knowledge domains. The goal is to get the LLM-based DSIE configured to employ various reasoning methods, including deduction, induction, abduction, and formal thinking, to facilitate the generation and expansion of knowledge.

It is clear that this approach will not be optimal for all possible knowledge domains, but it is sufficient for use cases where knowledge can be easily represented in text, such as film script creation, short stories, and technical documentation. After validating the concept with an initial implementation, it will be essential to delve deeper into the subject by developing more formalized inference engines than those based on LLMs. Similarly, representing knowledge as simple character strings allows us to quickly group knowledge by domain, but this may not necessarily be the most optimal approach. However, this strategy of attempting to validate the concept more thoroughly is one of the most pragmatic and straightforward ways to implement a Thinking Database.

ThinkyDB Abstract Use Cases

The following table presents abstract use cases that we aim to utilize in the initial phase to validate the utility of ThinkyDB, demonstrating its effectiveness across various applications in Generative AI.

Use Case	Description
RAG Semantic Search	Get inferred knowledge into generated responses. Improves search results by interpreting user intent and providing contextually relevant answers.
Coherence of Characters	Ensures that characters in a narrative maintain consistent traits and behaviors throughout.
Coherence of Places	Verifies that locations in a narrative are consistently described and accurately referenced.
Coherence of a Document	Assesses whether a single document maintains logical consistency and clarity.
Coherence of a Set of Documents	Evaluates the consistency across multiple documents to ensure aligned themes and information.
Generate Syntetic Data for Fine Tuning	Creates artificial datasets that are tailored to specific domains to enhance model training and fine-tuning, improving performance on domain-specific tasks.
Generate Summary Reports	Analyzes the emotional tone of text, useful in customer feedback analysis or content review.
Identify Sentiment or Tone	Analyzes the emotional tone of text, useful in customer feedback analysis or content review.
Generate Coherent and Interesting Stories	Utilizes AI to craft narratives that are not only logically consistent but also engaging, ensuring a captivating reading experience across various genres and contexts.

These use cases highlight how ThinkyDB can be applied to diverse scenarios in Generative AI, from improving narrative coherence and engagement in storytelling to generating synthetic data that enhances model training and adaptation for specialized applications.

ThinkyDB and RAG-related Research

The objective of this chapter is to summarize the vision of a research plan aimed at integrating advanced Retrieval-Augmented Generation (RAG) techniques into ThinkyDB to enhance domain-specific inference capabilities and broaden its utility across various applications. The plan focuses on providing advanced RAg features for ThinkyDB users but also leveraging RAG to improve the accuracy and relevance of knowledge retrieval and inference within ThinkyDB.

Despite the potential of RAG, there are several challenges and misconceptions that need to be addressed for effective implementation in ThinkyDB. One of the primary issues with traditional RAG implementations is their tendency to fall short in real-world applications. While simple RAG setups may work for demonstrations, they often struggle with messy, unstructured data and unforeseen user queries. To overcome this, ThinkyDB must ensure robust data preprocessing and indexing strategies. This includes effective chunking techniques to maintain context integrity and avoid the common pitfall of retrieving irrelevant or contradictory information.

Another key aspect to consider is the need for precise query handling. Users often pose ambiguous or poorly formulated questions, making it essential for ThinkyDB to employ query rewriting and expansion methods. These methods will refine user queries before they reach the RAG system, ensuring that the most relevant documents are retrieved from the database. For example, leveraging techniques such as Hypothetical Document Embedding (HyDE) can generate hypothetical responses to better align retrieval with user intent.

Additionally, post-retrieval optimization is crucial for enhancing the relevance of information provided by ThinkyDB. Techniques such as re-ranking retrieved documents based on their contextual alignment with the query can significantly improve the quality of the generated responses. This can be further supported by using metadata filtering, which allows for more targeted searches within specific segments of the data.

In the second stage, to ensure that ThinkyDB can handle a variety of data types, it is necessary to develop support for multimodal data integration. This includes processing text, images, tables, and code snippets in a way that each type of data can be effectively indexed and retrieved. The system should be capable of converting different data types into compatible embeddings and using them to generate meaningful responses.

Research on Indexing and Organising Knowledge Shards.

This chapter presents a collection of ideas and approaches that will be explored as part of the MVP development for ThinkyDB, focusing on the efficient indexing and organization of Knowledge Shards to enhance retrieval and contextual relevance.

The efficient organization and indexing of Knowledge Shards within ThinkyDB are crucial for enhancing the utility and responsiveness of this Thinking Database. By assigning each shard a specific location within a conceptual space, identified by various dimensions, ThinkyDB can significantly improve its ability to retrieve and utilize the most relevant knowledge. This approach involves creating a multi-dimensional indexing framework, wherein each Knowledge Shard is placed according to attributes that define its relevance and applicability.

One key dimension for indexing is the "Person" dimension, which categorizes knowledge related to individual users, including personal data, biographical details, or specific preferences. By associating shards with individual identities, ThinkyDB can tailor responses to align with user-specific needs, enhancing the personalization and relevance of the information provided. This capability is particularly useful in scenarios where the context of a query is heavily dependent on the identity or history of the person making the inquiry.

Another essential dimension is the "Class of Objects," which involves categorizing shards based on different object classes such as vehicles, devices, or biological organisms. This classification enables ThinkyDB to streamline information retrieval for queries focused on specific types or categories of objects, thus facilitating targeted responses that are more accurate and contextually relevant.

Geographical location serves as a critical dimension for organizing Knowledge Shards, allowing ThinkyDB to handle location-specific queries effectively. By indexing knowledge according to geographical parameters, the system can provide regionally relevant information, which is invaluable for applications requiring local knowledge, such as regional regulations, customs, or environmental data.

The "Goals and Objectives" dimension is another crucial element in this indexing framework. By aligning Knowledge Shards with specific goals or objectives—such as business targets, research aims, or personal aspirations—ThinkyDB can streamline the retrieval process to focus on insights that directly correlate with the user's current objectives. This alignment ensures that the database not only responds to queries but does so in a way that supports actionable decision-making aligned with specific user intents.

Scientific domain categorization further enhances the utility of ThinkyDB by organizing Knowledge Shards according to distinct fields such as physics, biology, or economics. This dimension ensures that queries are addressed using domain-specific expertise, which is crucial for academic and research applications where the precision and specialization of knowledge are paramount.

Beyond these primary dimensions, several other attributes can be leveraged to refine the indexing of Knowledge Shards. Temporal context, for instance, allows ThinkyDB to distinguish between historical, current, and future knowledge, making it possible to provide insights that are temporally relevant. Similarly, cultural context can be used to incorporate knowledge about customs, traditions, and social norms, allowing for culturally sensitive responses.

Legal and regulatory dimensions enable the system to focus on compliance-related knowledge, which is essential in fields like finance, healthcare, and data privacy. Emotional tone indexing allows ThinkyDB to generate responses that are emotionally appropriate, catering to queries that require empathy or specific emotional intelligence. Technological domain indexing supports the retrieval of knowledge relevant to specific technological fields, such as artificial intelligence or blockchain, ensuring that the system can address highly specialized technical queries.

In addition to these, ethical considerations serve as a guiding dimension for filtering knowledge based on moral and ethical standards, which can be critical in areas involving AI ethics, bioethics, or social responsibility. Linguistic context helps in managing multilingual interactions, making ThinkyDB capable of handling queries in various languages or dialects, which enhances its global applicability.

Educational level indexing aligns Knowledge Shards according to the complexity of content, ensuring that responses are appropriate for the user's level of understanding, whether they are beginners or experts. Media type categorization allows ThinkyDB to identify the most suitable format for presenting information, be it text, audio, video, or images, thereby improving the overall user experience.

Furthermore, indexing Knowledge Shards by events and activities can streamline responses related to specific occurrences, making it easier to access knowledge about particular historical events, conferences, or other activities. Social context indexing, based on demographics or community interests, allows ThinkyDB to tailor its responses to specific societal segments, enhancing relevance in social and community-related applications.

Finally, relationship context, which categorizes knowledge based on relational attributes such as cause-effect or part-whole relationships, provides deeper insights into complex queries involving interconnected concepts. This approach not only enhances the relevance of the information retrieved but also supports a more comprehensive understanding of the relationships between different pieces of knowledge.

The potential for Knowledge Shards to contain meta-knowledge about other shards further enriches this multi-dimensional framework. By establishing a layered system of shard management, ThinkyDB can optimize its search processes, directing queries from high-level shards to more specialized ones as needed. This hierarchical approach ensures that ThinkyDB can efficiently manage and retrieve knowledge, even as the database grows in complexity and size.

The proposed multi-dimensional indexing system and the hierarchical structuring of Knowledge Shards are foundational for the development of ThinkyDB as a robust and versatile Thinking Database. By effectively categorizing and organizing knowledge, ThinkyDB can enhance its ability to provide precise, contextually relevant, and actionable insights, thus meeting the diverse needs of its users across various domains and applications.

In conclusion, while the information within an individual Knowledge Shard may be relatively unorganized due to its focused purpose, ThinkyDB will potentially house thousands or even tens of thousands of these shards, systematically organizing layers and domains of knowledge. This structure will enable the rapid identification of useful shards with processed and ready-to-use knowledge, facilitating efficient RAG queries and the generation of reports or documents relevant to user requests.

Conclusion

The introduction of Thinking Databases equipped with various types of Domain-Specific Inference Engines (DSIEs) provides a good exploration path into potentially enhancing knowledge management and reasoning across specific knowledge domains. As both concepts are in their early stages, corresponding to a Technology Readiness Level (TRL1 towards TRL2), it is crucial to outline both the underlying principles and necessary observations that will guide the further development of these systems.

This approach to integrating DSIEs within Thinking Databases is currently theoretical, emphasizing the need for rigorous, incremental research and validation. The engines are envisioned to provide tailored computational strategies that align closely with the specific requirements and complexities inherent in different fields, from scientific research to financial analysis. The promise of such technology is to facilitate more accurate, context-aware processing of domain-specific data, which could transform decision-making processes and knowledge discovery in profound ways.

However, given the nascent stage of these technologies, it is appropriate to temper expectations. The immediate focus is on defining clear research pathways that can systematically address the challenges of developing and implementing DSIEs. This involves not only theoretical formulations but also practical considerations about the scalability, adaptability, and integration of these systems with existing technological frameworks.

Moreover, the potential of DSIEs to autonomously generate and refine rational theories remains a long-term goal. Current efforts are directed towards establishing a robust foundation for such advanced capabilities, which would require significant advancements in both domain-specific modeling and computational inference.

In conclusion, while the development of Thinking Databases with DSIEs could significantly advance the field of domain-specific knowledge discovery, this report serves as a preliminary framework that outlines the initial principles and considerations. It is a first step in a long research journey that aims to progressively build towards the realization of these sophisticated systems.

Bibliography

[TDB] Thinking Database Vision (Internal Report from Axiologic Research, 2024), https://www.axiologic.net/downloads/thinking_databases_report.pdf

[TKDB] Initial implementation of ThinkyDB, a Thinking Database that uses LLMs as its DSIE https://github.com/AssistOS-AI/ThinkyDB

[RAG] Research ideas about RAG limitations and techniques to improve and enhance, https://towardsdatascience.com/what-nobody-tells-you-about-rags-b35f017e1570

[NSCR] Garcez, A. S. d., Lamb, L. C., & Gabbay, D. M. (2015). Neural-Symbolic Cognitive Reasoning. Springer. This book explores the intersection of neural networks and symbolic reasoning, providing a foundational text for understanding the integration of these two approaches.

[NSLR] Tarek R. Besold, Artur d'Avila Garcez, etc. Neural-symbolic learning and reasoning: A survey and interpretation. This survey discusses various approaches and applications of neural-symbolic integration, showcasing their relevance to AI and cognitive sciences.

[EM] Bergmann, R. (2002). Experience Management: Foundations, Development Methodology, and Internet-Based Applications. Springer. This book delves into case-based reasoning, discussing how experiences can be used for problem-solving in new situations.

[ES] Giarratano, J. C., & Riley, G. (2005). Expert Systems: Principles and Programming. Course Technology Ptr. This text covers the fundamentals of expert systems, with a focus on rule-based reasoning.

[ONT] Staab, S., & Studer, R. (Eds.). (2009). Handbook on Ontologies. Springer Science & Business Media. This comprehensive handbook provides detailed information on the development and use of ontologies in information systems.

[FT] Jeremy Howard and Sebastian Ruder, 2018. Universal Language Model Fine-Tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

[BERT] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[TR] Clark, P., Tafjord, O., & Richardson, K. (2020). Transformers as soft reasoners over language. arXiv preprint arXiv:2002.05867. This work explores how transformer models can be used for reasoning tasks, providing insight into their capability to handle logical inferences.

[LER] Evans, R., & Grefenstette, E. (2018). Learning explanatory rules from noisy data. Journal of Artificial Intelligence Research, 61, 1-64. This research discusses the extraction of interpretable rules from data, which is similar to formalizing facts and theorems for use in domain-specific reasoning.

[ATP] Polu, S., & Sutskever, I. (2021). *Generative Language Modeling for Automated Theorem Proving*. arXiv preprint arXiv:2009.03393. This study explores the application of generative language models in the context of automated theorem proving, suggesting how LLMs could be used to formulate and prove mathematical theorems.

[ETH] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, J., Tang, E., ... & Song, D. (2021). Aligning AI with Shared Human Values. arXiv preprint arXiv:2008.02275. This paper demonstrates how LLMs can be trained to reason according to human ethical standards, essentially formalizing ethical theories in the decision-making processes of AI systems.

[BIO] Zeng, W., Church, G. M., & Szolovits, P. (2021). *Language Models are Few-Shot Learners for Biological Sequence Analysis*. Bioinformatics.

[SOC] Ahn, S., Choi, H., Pärnamets, P., & Wittenberg, E. (2020). Using Language Models to Predict Human Behavior in Moral Dilemmas. Cognitive Science.