

SolrAI Proposal

Introduction

I am an experienced software developer, web designer, and database / knowledge base developer with a track record of delivering successful projects and solutions to clients across a variety of industries for nearly half a century. With a background in COBOL, Basic, Perl, Java, JavaScript, SQL, REST API, PHP and Drupal CMS development, I excel at providing innovative solutions to complex IT challenges. In addition to my coding and development skills, I also have a lot of experience in Systems Design and Project Management having done consulting and development work for both the Los Angeles Unified and Lynwood School Districts, Greenwood Publishing (eBooks), The Fien Group (InfoBases) and the Alliance of Motion Picture and Television Producers (AMPTP).

For the past 15+ years, I have been primarily involved in developing and maintaining custom Drupal sites for various clients, including SchoolBoard.net, a Drupal CMS used by school boards for “paperless” board meetings across the United States, and the AMPTP Member Website, a Drupal site that provides secure access to over 100 years of articles and collective bargaining agreements (CBA), including the CBAs with the Writer’s Guild of America (WGA), Director’s Guild of America (DGA) and the Screen Actor’s Guild (SAG).

In the Drupal contrib world, I am probably most well-known for the first OG User Roles (5.x and 6.x) module which made it possible to assign role permissions in Organic Groups. <https://www.drupal.org/node/87679>

My LinkedIn profile is here: <https://www.linkedin.com/in/ron-parker-14aa6b18>

On January 30, 2023, I posted a feature request where I outlined my initial thoughts for a Drupal-OpenAI chat system: <https://www.drupal.org/project/openai/issues/3337774>

By March, I had come up with a full design for the system, and created PHP prototypes (along with flowcharts) for both content ingestion (embedding): <https://www.drupal.org/project/openai/issues/3339686#comment-15011514> and data query: <https://www.drupal.org/project/openai/issues/3339686#comment-15024846>

I am actually about 35% through the SolrAI project I am proposing here. Just could use a little help turning this into a fully functional and easy-to-use module, which the funding will help me acquire. I’ve watched Drupal grow over the years as I’ve grown with it. I see this as my opportunity to give back to the community in a big way.

Executive Summary

Currently, in the rapidly advancing world of Artificial Intelligence and Large Language Models, there is a conspicuous absence of a module designed specifically to allow Drupal users to conversationally interact with their Drupal site content. This lack of conversational engagement hinders the ability to access and interact with site content intuitively, efficiently and in a more human-like manner, creating an AI gap in the Drupal

user experience. To address this, I intend to develop a 'Drupal Solr-AI Conversational Content Query Module', or 'SolrAI' for short.

The SolrAI Module is an advanced conversational AI module designed to run on Drupal CMS. It leverages the power of Search API Solr, a popular search API for Drupal, to index and retrieve website content efficiently. The system embeds text files into a document store and integrates the capabilities of the latest OpenAI model, GPT-4, to provide a conversational interface for querying this content. Users can ask questions or initiate conversations related to the site's content, and the module will respond with accurate responses that take into account the context of the conversation and the specifics of the content it's referencing. This essentially enables users to "chat" with the site content, providing a more fluid and natural conversation experience.

This module brings the power of AI to Drupal-based websites, enhancing user engagement and providing a revolutionary way to interact with website content.

I have structured the development plan into nine distinct phases, which I am confident can be successfully executed within a 90-day timeframe:

- **Phase I.** Develop and test ingestion prototype. This is a PHP script which will create a vector store directly from the Solr index
- **Phase II.** Develop and test query prototype. A PHP script that facilitates user interaction with the content stored in the vector database. Through the use of REST API, this script will orchestrate chat completions with OpenAI's GPT-4 model, using the stored content as the conversational context. The ultimate objective is to construct a dynamic conversational chat system that seamlessly integrates with the existing Drupal content base, thereby transforming the way users engage with the site's content.
- **Phase III.** Begin development of SolrAI module proper. Essentially, take the code from the PHP scripts developed in Phases I and II and incorporate them into a Drupal module.
- **Phase IV.** Develop and test the query user interface. Here we hope to create a query UX similar to that of ChatGPT, but also allowing the user to select the specific site content to use in the query, set model temperature for responses, and choose the number of documents to use for context.
- **Phase V.** Add configuration UI and update the Solr ingestion system. Site administrators will need an interface to:
 - Select content datasets and fields to embed.
 - Assign user permissions.
 - Set ingestion parameters such as vector store and OpenAI keys and API endpoints, document "chunk" and "overlap" sizes, document summary configuration, whether to ignore orphaned files, whether to force updates, when to run updates, etc...
 - Any additional items discovered during development.
- **Phase VI.** Develop and test object maintenance code. Add the CRUD code required to maintain synchronization between the Solr index and the vector store. This will consist primarily of Search API Solr event calls that will update the vector objects when the local Drupal entities are updated.

- **Phase VII.** Develop and test access control code. This is the code which will determine what vector objects the user has permission to “view”, based upon their access to the entities on the Drupal site. These will be the same. Initially, node access and group access will form the core of this access control strategy.
- **Phase VIII.** Add and test miscellaneous features. These are features which will come up during the development process. Their importance to the overall project will determine whether they are included at this stage or not.
- **Phase IX.** After all tests and Drupal checks are completed, release as contrib module to community for evaluation and real-world testing.

I should note that Phases I and II are completed. I believe that with the appropriate consulting and development assistance in the areas above where I need help, I can accomplish a fully functional module with the above feature set within 90 to 120 days.

Pain Point

For the past two decades, the Drupal system, much like many other online platforms, has operated on a keyword search paradigm. This system demands precision in terms of lexical matches for search queries, often necessitating the use of boolean operators to filter results. This approach can prove cumbersome and non-intuitive as it doesn't reflect how humans naturally search for information in real-life situations.

The current keyword-centric search paradigm poses challenges to the user experience. It lacks understanding of semantic meaning, struggles with synonyms and variations, lacks personalization, has difficulty with ambiguous queries, and provides a linear search experience. This often leads to less relevant search results and a less intuitive search process. In contrast, the proposed SolrAI module, a semantic and context-aware search system, is designed to overcome these limitations. It can interpret the intent behind search queries, handle different wordings, personalize results based on user behavior, clarify ambiguous queries, and provide an interactive search experience, thus making it easier for users to find the information they need.

Consider a user visiting a health website to find information about a common cold. If they enter the keyword "cold" into the current keyword-centric search system, the system might return a range of results including information about "cold weather", "cold drinks", or "cold storage". This is because the system can't understand the user's intent or the context behind the keyword.

On the other hand, with the SolrAI module, the user could type, "What are the symptoms of a common cold?" and the system would understand the semantic meaning behind the query. It could even follow up with personalized recommendations like, "You were reading about flu symptoms last week. Would you like to compare them with the symptoms of a common cold?" This kind of interaction is more intuitive and helps the user find the information they need more efficiently.

A semantic, context-aware search system will understand the meaning and context behind search queries, handle synonyms and variations, personalize search results based on user preferences and behavior, handle ambiguous queries more effectively,

and provide a more interactive, dialogue-based search experience. In short, a more human-like approach to data retrieval. These improvements would significantly enhance the user experience, making it easier and more intuitive for users to find the information they're looking for.

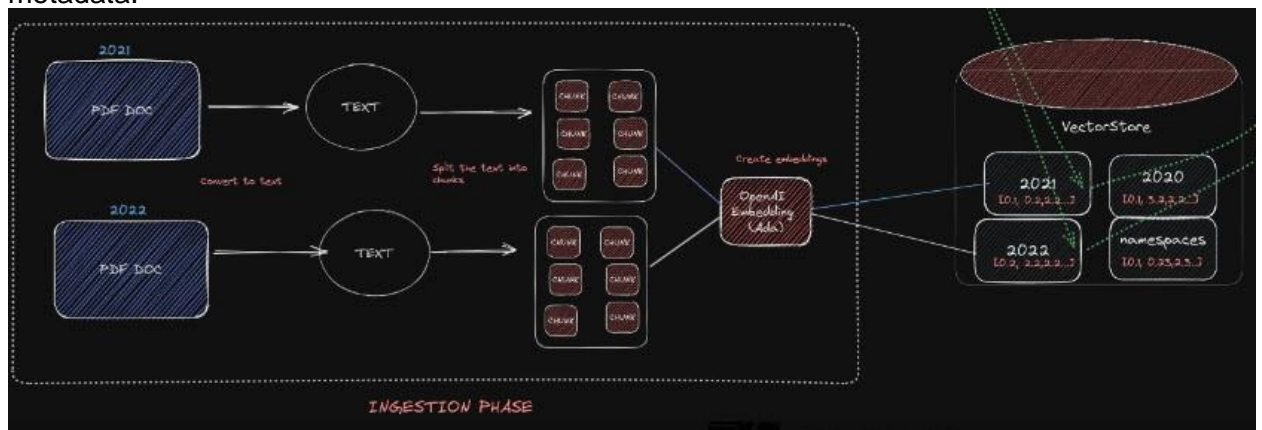
However, implementing Semantic Search in Drupal is not as straightforward as many online resources suggest, due to its diverse content management system, consisting of various entity types, access control systems, and robust categorization.

Existing tools and frameworks, such as LangChain, may be suitable for smaller sites with limited documents but fall short in providing the necessary content architecture and scalability for larger sites and enterprise-level organizations. These entities require a system that can accommodate hundreds or even thousands of documents, with added requirements for document organization, classification, and controlled access, as well as a multi-layered search facility that allows for both legacy (keyword) and semantic search.

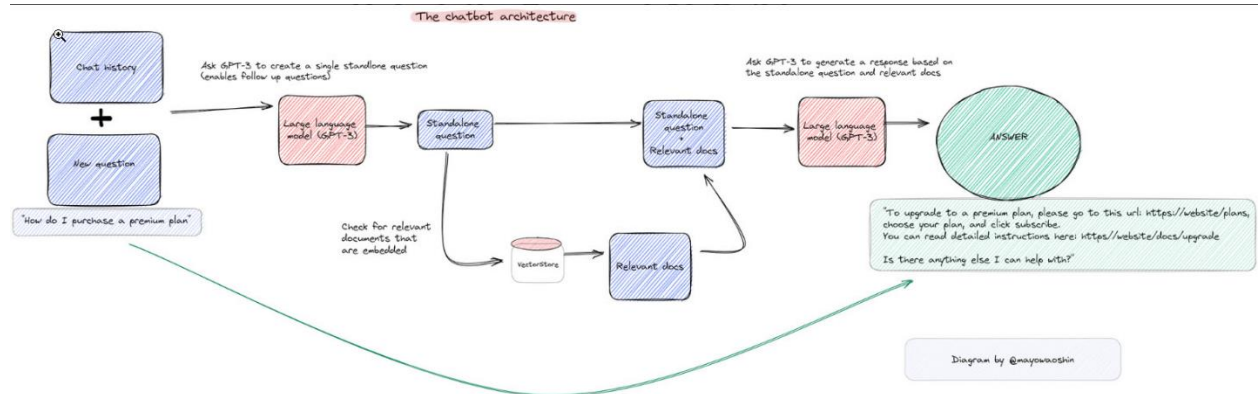
Thus, we encounter two significant pain points – the absence of an intuitive, conversation-based search function within Drupal, and the overall need for a scalable, flexible query system for larger sites and enterprises that allows both legacy and semantic search patterns. Addressing these issues is crucial to bridge the gap between the current keyword-centric search paradigm and the more natural, conversation-driven interactive model that users increasingly expect in the era of AI and Large Language Models. The 'SolrAI' project will not only add semantic search to any Drupal site, but it will also position Drupal as a viable alternative to the current leading conversational chat development frameworks.

Solution

Our proposed solution, the SolrAI module, leverages the existing Apache Solr index created by the Search API Solr module, which contains not only the entire text of a site's content but also all the organizational information (groups, taxonomy, parent node, etc.), referred to as "metadata." SolrAI utilizes this index to create a vector store of all of a site's content, handling the chunking of documents and preserving the source content metadata.



The SolrAI module also includes an AI-powered conversational chat completion query component, which enables users to conduct semantic searches against the vector store while maintaining site content organizational tags and permissions. This creates a Drupal content chatbot that works seamlessly within the Drupal organizational framework.



But, why? By understanding complex queries and generating context-aware responses, SolrAI provides a far more intuitive and interactive user experience than the current keyword-centric search paradigm.

Consider a Drupal-based news website with articles spanning a wide range of topics. Currently, if a user wants to find articles about the impact of climate change on polar bears, they would have to try different combinations of keywords and sift through the results, which could include articles that only tangentially mention their topic of interest.

With SolrAI, the user could type in a complex query like "show me articles about the impact of climate change on polar bears". The system would understand the intent behind the query and provide a list of relevant articles.

Additionally, SolrAI could engage with the user in a more conversational manner. For instance, after the user reads an article, SolrAI could follow up with a question like "Would you like to read more about climate change or about polar bears?" This kind of intuitive, personalized interaction could significantly enhance user engagement and make the process of finding information more enjoyable.

By implementing such a context-aware and conversation-driven search system, Drupal could position itself as a strong contender among conversational chat development frameworks, offering a user-friendly and efficient way to navigate and interact with website content.

But, how? To implement this feature on any Drupal site, the following modules will need to be installed: Search API Solr, Groups, Taxonomy, and SolrAI. SolrAI currently uses Weaviate, an open-source vector database, for vector storage, allowing users to house their vector store either in the cloud or on local servers.

By pairing a SolrAI Drupal site with an open-source Large Language Model (LLM) and an open-source vector store, we can create a 100% open-source AI-powered chat conversation Q&A system - potentially the first end-to-end system of its kind. Additionally, the development of SolrAI will spur more PHP development in the AI-powered conversational chat space, diversifying the market currently dominated by Python.

This innovative and impactful solution addresses the pain points of lacking an intuitive, conversation-based search function within Drupal and provides a scalable, flexible query system for larger sites and enterprises, allowing them to benefit from both existing and new semantic search capabilities.

Funding and Resources

I am seeking a total of \$18,000 (USD) in funding to support the development, testing, and release of the SolrAI Drupal Module. This amount is based on a detailed analysis of the various phases of the project, the expected time commitment, and the resources required for each phase. The funding will be allocated as follows:

1. **Development Costs:** \$7,000. This covers the cost of the time and resources required to code the module, including prototyping, the creation of the user interface, and the development of other key features.
2. **Testing and Quality Assurance:** \$2,000. This covers the cost of testing the module during development and before release, including the creation of test scripts and the execution of test cases, as well as any necessary adjustments based on testing results.
3. **Documentation and User Guides:** \$2,500. This covers the time and resources necessary to develop comprehensive documentation and user guides to ensure that the module can be effectively used and maintained by the Drupal community. Lack of clear, detailed documentation has always been my biggest complaint with Drupal. I'd like to create the Gold Standard for how it should be done.
4. **Release and Community Support:** \$2,500. This covers the cost of preparing and releasing the module as a contrib module to the Drupal community, as well as providing initial support and updates after the release.
5. **Project Management and Overheads:** \$4,000. This covers the costs associated with managing the project, including planning, coordination, communication, and other administrative tasks. It also covers potential overheads such as software, hardware, or other resources that may be required for the successful execution of the project.

I believe that this funding will enable me to create a highly effective and impactful module that will significantly enhance the user experience on Drupal-based websites. It will also allow me to engage other Drupal developers and consultants, hopefully bringing in young new local talent.

The development of the SolrAI Drupal Module is a step forward in the integration of AI technologies into the Drupal framework, and I am eager to contribute to this exciting evolution.

Your support will not only bring this project to fruition but also contribute to the broader Drupal community by fostering innovation and expanding the capabilities of Drupal as a powerful content management system.

Deliverables

Upon completion of the SolrAI Drupal module project, the following key deliverables will be provided to ensure a comprehensive and well-rounded solution for Drupal users:

1. SolrAI Drupal Module: A fully functional, AI-powered conversational content query module for Drupal, leveraging the Search API Solr index to create a vector store and integrating with OpenAI's GPT-4 model for conversational interactions.
2. The initial module will support the OpenAI LLMs and the Weaviate Open Source Vector Database.
3. Module will have these features:
 - a. Query Prototype: A PHP script facilitating user interaction with the content stored in the vector database, enabling seamless conversational chat completion with GPT-4 while preserving the Drupal content structure.
 - b. Query User Interface: A user-friendly interface for querying the site's content, allowing users to select specific content for querying, set model temperature for responses, and choose the number of documents to use for context.
 - c. Configuration UI: An intuitive interface for site administrators to manage content datasets and fields, assign user permissions, set ingestion parameters, and configure other relevant settings.
 - d. Object Maintenance Code: CRUD code for maintaining synchronization between the Solr index and the vector store, ensuring up-to-date content representation in the conversational interface.
 - e. Access Control Code: A robust access control system that determines user permissions to view vector objects, based on their access to the entities on the Drupal site, initially focusing on node access and group access.
 - f. Logging of system and end-user queries and token usage.
4. Comprehensive Documentation: Detailed documentation and user guides, covering the installation and configuration of the SolrAI module, its features and functionalities, and best practices for leveraging the module effectively.

5. SolrAI Contrib Module Release: A fully tested and verified SolrAI module, released as a contrib module to the Drupal community for evaluation and real-world testing, with ongoing support and updates to ensure its continued success.

For the **end-user** or visitor, the SolrAI module will provide a more intuitive and interactive experience. Instead of having to input precise keywords, they will be able to engage in a more natural, conversation-like interaction with the search function.

For **content creators**, the SolrAI module would simplify the process of making their content discoverable. They would no longer need to worry as much about including precise keywords in their content. Instead, they could focus on creating high-quality, meaningful content, knowing that the AI-powered search function can understand the context and meaning of their work.

For **administrators**, the SolrAI module would offer valuable insights into users' search behavior, as it could log not only the keywords used in searches but also the intent and context of the searches. This could help administrators optimize their content strategy, user interface, and overall site organization to better meet their users' needs.

Finally, for **developers and site builders**, the SolrAI module would open new possibilities for site functionality and user experience design. It could be used to create advanced features like personalized content recommendations, interactive Q&A sections, or even AI-powered chatbots. It would also offer an exciting opportunity to work with cutting-edge AI technology and contribute to the evolution of the Drupal platform.

Conclusion

I want to make sure, if I do nothing else in this proposal, that I make it clear what the value of "semantic search" or "context-awareness" means.

"Context-aware responses" in the context of this proposal means that the SolrAI module doesn't just answer based on the specific question asked by the user, but also takes into account the broader context of the conversation and the content that it's drawing upon.

For example, suppose a user is interacting with a Drupal site about classic literature and asks, "Who is the main character in Pride and Prejudice?" The system would understand the context and respond, "The main character in Pride and Prejudice is Elizabeth Bennet."

But if the user then follows up with, "What is her relationship with Mr. Darcy?" the system would need to remember the context from the previous question and respond appropriately, "Elizabeth Bennet and Mr. Darcy are romantic interests in Pride and Prejudice."

The SolrAI module achieves this with a site's existing content by using the OpenAI GPT-4 model's ability to maintain a conversation history. It feeds previous exchanges in the conversation as well as the text of the relevant content from the Drupal site to the model,

allowing it to generate responses that are consistent with the overall conversation and the specifics of the content it's referencing.

This feature ensures that the SolrAI module can provide a fluid and natural conversation experience, much like interacting with a human who remembers the context of the conversation.

This is the power of what this proposal, if accepted, brings to every Drupal user on the planet.