chATLAS

An Al Assistant for the ATLAS Collaboration

DANIEL MURNANE, GABRIEL FACINI, RUNZE LI, DANIELE DEL SANTO & CARY RANDAZZO

1st Large Language Models in Physics Symposium **February 21, 2024**











chATLAS Overview

- Motivation
- History of chATLAS
- Data Gathering
- Infrastructure
- Deployment Result: Chat Interface and Demo
- Roadmap
- Updates, Challenges, and Summary



MOTIVATION

- Quickly parsing documentation and twiki
- Semantic search and availability of heterogeneous sources of ATLAS information
- Summarizing research
- Connecting the dots between different groups
- Debugging software
- Searching and summarizing JIRA and Glance information



HISTORY

- In April 2023, initial ATLAS ML Forum meeting to discuss usage of ChatGPT and Github Copilot within ATLAS
- In June 2023, presentations on several ongoing works to use LLMs within ATLAS
 - ATLAS-GPT: Daniel Murnane
 - ChATLAS: Gabriel Facini
 - Google Bard + ATLAS: Kaushik De
 - Analysis Description Language + GPT: Gokhan Unel
- Decision made to converge ATLAS-GPT and ChATLAS and create an official prototype
- Fortnightly developer meetings kicked off in August 2023
- Currently approx. seven part-time contributors
- Launched ATLAS-public demo November 16 https://chatlas-flask-chatlas.app.cern.ch/



Data Gathering: How Scraping Began

ATLAS Twiki

- Start with set of "Starting URLs"
- Recursively visit included links
- Find all headers, and visit content below
- Append metadata of twiki (parent structure, date revised, etc.)

CDS

- Discover whether the CDS paper has a Gitlab latex repo
- If latex exists, pull from repo and (planned) convert to markdown
- (Planned) Use unstructured library to parse markdown
- If latex does not exist, use nougat library to read PDF (including equations) into markdown

Indico

- Load event list
- Scrape timetable contents (date, title, speaker, etc.)
- (Planned) Pull PDF slide decks and minutes
- (Planned) Parse in the same way as in CDS



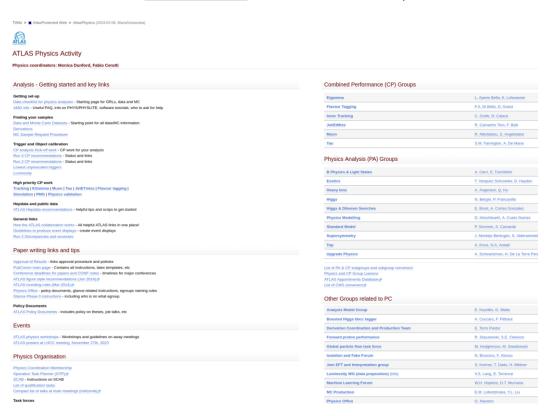
Data Gathering: Volume Measurements

CDS ATLAS Group Indico PDF **ATLAS** E-group Jira Papers & Twiki Software level Mattermos Archive Meetings Plots Codebases Notes Docs Docs



Twiki

Over 2000 ATLAS Twiki Topics



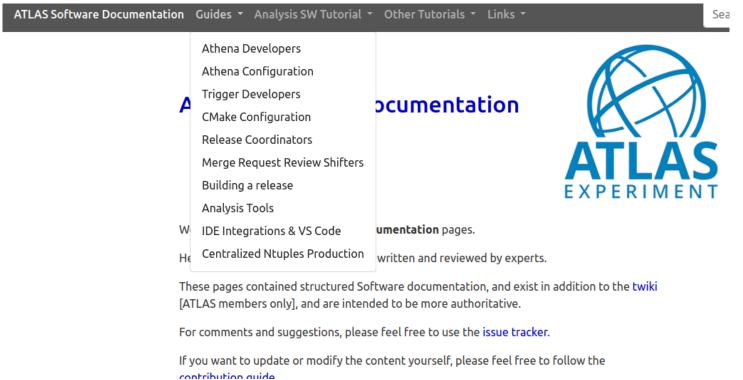
Using: BeautifulSoup and auth-get-sso-cookie



ATLAS Software Docs

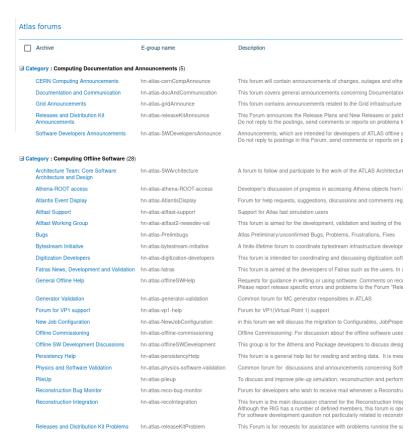
Hundreds of ATLAS Software Docs

Using: BeautifulSoup and auth-get-sso-cookie





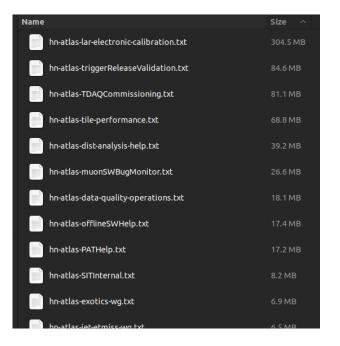
E-group Archive



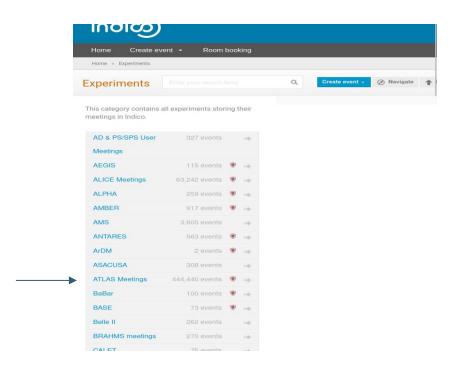
8912 topics x (1-50+ messages) for largest egroup

~ <u>tens of thousands of messages</u> to date

Using: BeautifulSoup and Selenium



Indico Meetings



Using: Nougat and Marker

440,440 ATLAS Indico Meeting Events



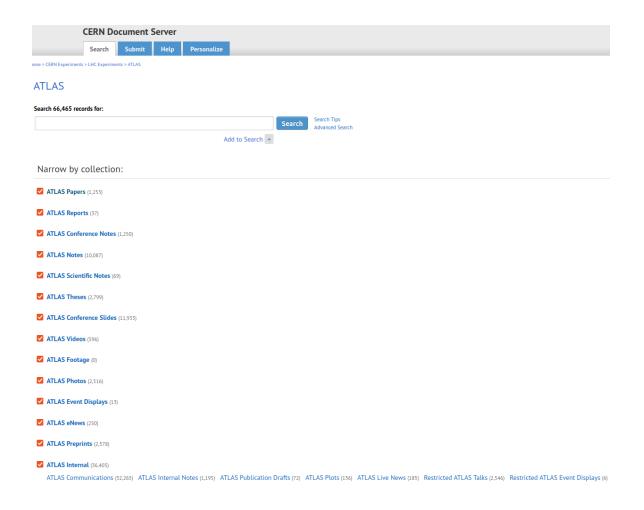
PDF Mattermost Jira ATLAS level Codebases Docs

Measurements Pending - <u>very large</u>

Using:
Pending further
experiments



CDS
Papers
& Notes



Using: Nougat and Marker

Total: <u>66,465</u> records found

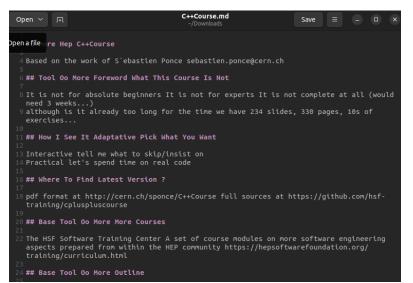
36,405 are Internal including communications, notes, etc.

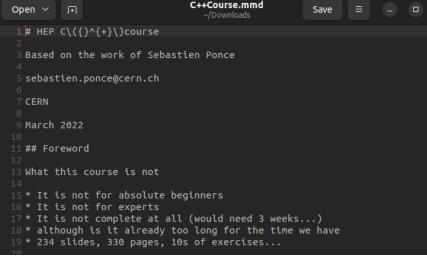


Data Gathering: A Tool Comparison

CDS
Papers
& Notes

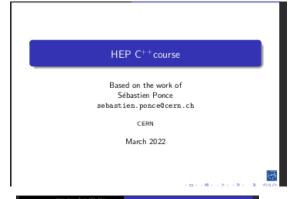
Slides more difficult to process and largest data in volume

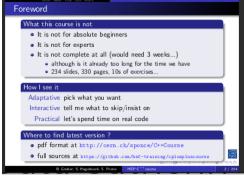




Faster

Marker vs Nougat (PDF scraping)





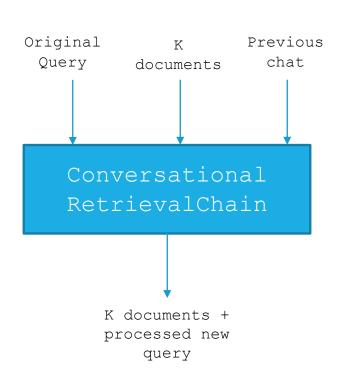
Processing: Chunking and Retrieval

Chunking

- (Current) Loop through HTML and Markdown heading sections
- If section exceeds 510 tokens, split with **SentenceTransformersTokenTextSplitter**
- Pass chunk through HuggingFace's sentence-transformers/all-MiniLM-L6-v2 model
- (Planned) Use built-in unstructured library to identify chunks
- Insert chunk into Chroma database, with metadata of file URL, twiki name

Retrieval

- All handled internally by qa = ConversationalRetrievalChain.from_llm(llm=model, retriever=db.as_retriever(), memory=memory, verbose=False,)
- LLM Model is GPT-3.5 from OpenAl API, retriever is default Chroma which contains the embedding model, memory is a buffer that retains all previous chat information
- Implicit is that the model aggregates all K-documents with a prompt to produce a new question based on the original question and the K-documents

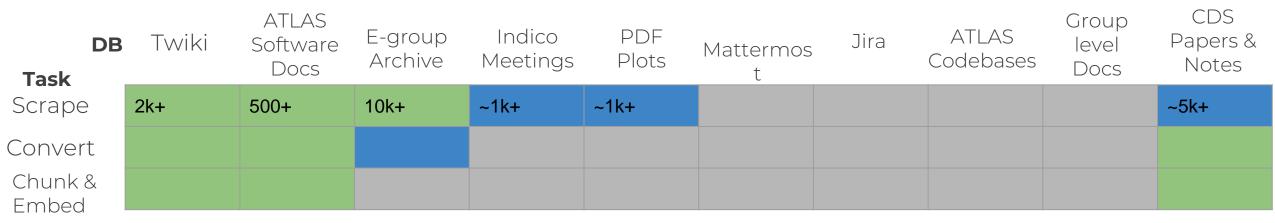




Summary of Gathering and Processing

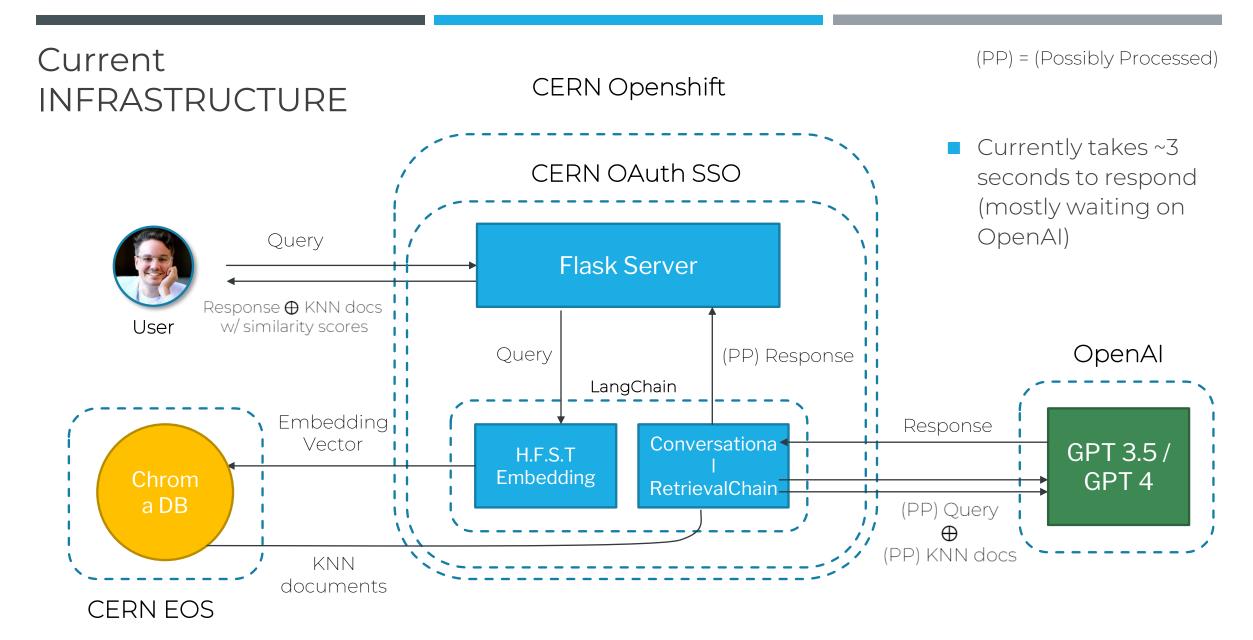
Summary includes...

- Diagram of all the possible ATLAS datasets and how many we have in the Database(DB)
- Chunked & Embedded Datasets are ready for or have been added to DB
- Stage the progress of each
- Over 37% of textualizable ATLAS datasets (minimum)
- Hardware upgrades were made to scrape in larger volume



Not yet started In Progress Complete

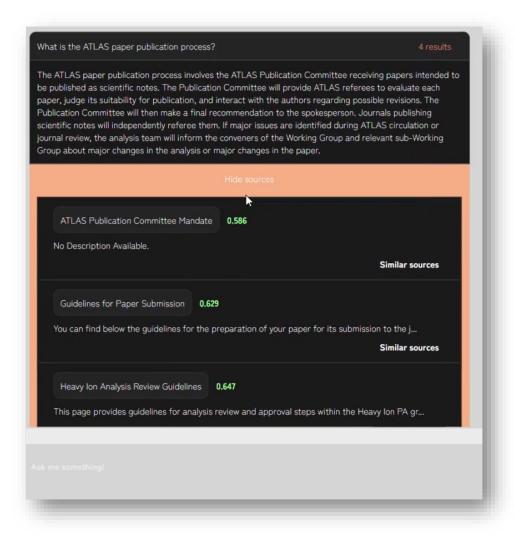




chATLAS

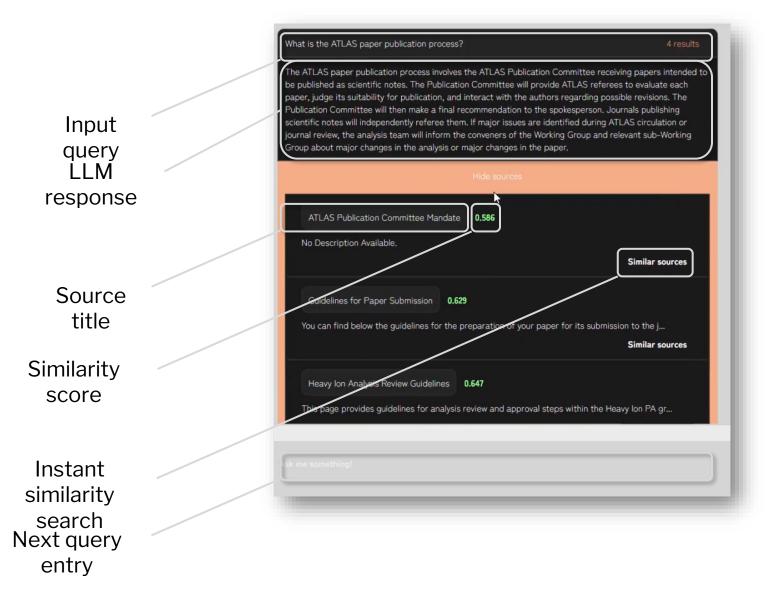
Planned (PP) = (Possibly Processed) **CERN Openshift** Infrastructure **CERN OAuth SSO** Filters: Query Which database (Twiki, etc) Flask Server Date Internal/External Response ⊕ KNN docs "Trustworthiness" w/similarity scores User Query & **Filter** LangChain Llama-7B Embedding! Response Vector on CERN H.F.S.T Conversational Embedding RetrievalChain GPUs a DB (KubeFlow?) (PP) Query (PP) KNN docs KNN documents **CERN EOS**

Chat Interface



Chat Interface

- Version 1.0 contains everything needed to answer a query
- Can cite the top sources used in the response
- Has a quick search for similar sources



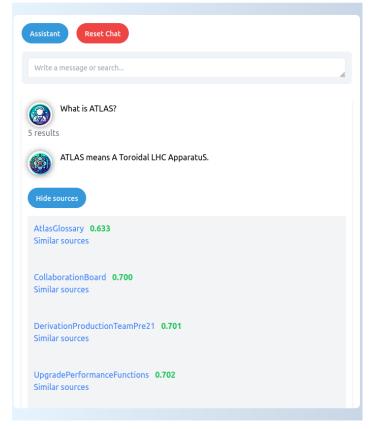
Chat Interface

- Version 1.0 contains everything needed to answer a query
- Can cite the top sources used in the response
- Has a quick search for similar sources
- Experimenting with a V2.0
 appearance that is lighter,
 and has a dedicated Assistant

 mode and Search mode



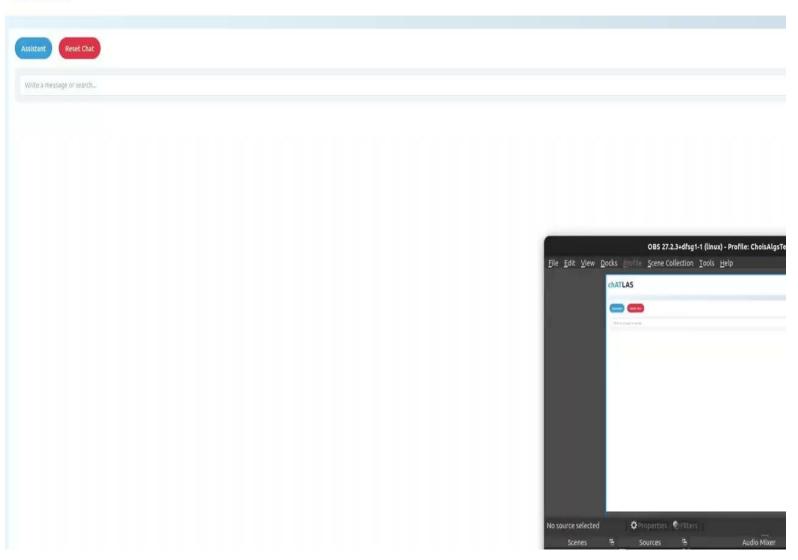




This is a chatbot application developed for ATLAS and CERN. It operates in both assistant and search modes across a variety of ATLAS databases (currently Twiki).

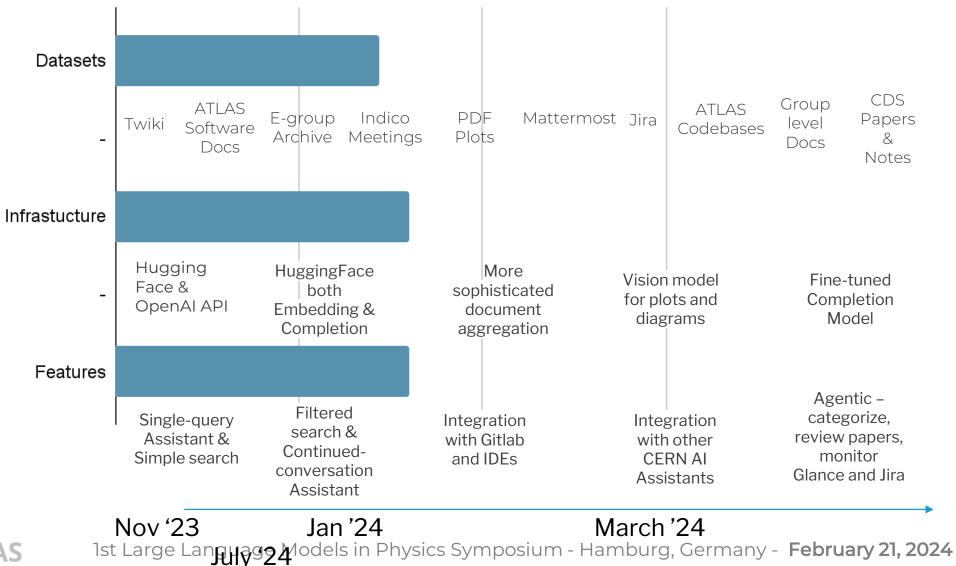
Demo







Roadmap (Milestone Tracking)



Recent Updates and Ongoing Work

- Studies of the use and performance of different models as well as response optimization via prompt engineering and similarity search techniques
- Indico scraping shown previously with Nougat and Marker
- Video caption search and embedding
- Fine-tuning model research and experiments on either CDS or Twiki
- App development, deployment, documentation, etc.
- Studies on how to include the ATLAS Glossary, and other "Dictionary" or "Reference" type documents - first-layer database search, fine-tuning, or even heuristic hand-engineering ("When is the NSW being upgraded?" - search in Glossary for any words present - place them in parentheses = "When is the NSW (New Small Wheel) being upgraded?")



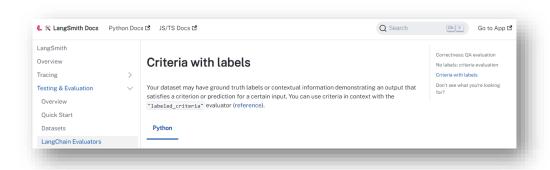
Challenges and Suggestions

- Getting the data! Highly heterogeneous file types, many behind authorisation walls, many stale
 or inaccurate, many requiring high levels of post-processing
- Community solutions could go a long way: Ensure that any experiment/collaboration databases are easily accessible and exportable. All websites should live in a git repo. All publications should be submitted and saved as latex, and compiled separately. All discussion forums should have anonymisation options. This would have saved ~1 year of data wrangling
- Hallucination is still a very real problem [https://www.arxiv-vanity.com/papers/2311.04348/]
- A high quality AI assistant probably requires fine tuning, which is an expensive task (less in gpuhours, more in expert-hours)
- Open-source solutions for UI are not particularly flexible. A tool built by+for the scientific community would be very useful! Open-source solutions for backend (retrieval, document aggregation) are perfectly fine.
- Codebase integration: experiment codebases are huge, not so well-commented, and nonobvious how to chunk. Perhaps an automated commenting algorithm as a pre-process step?



OPEN QUESTIONS

- How to avoid hallucinations? (Integrate latest research?)
- How to best "censor" politically incorrect responses (e.g. which analysis team is the best?)
- How to measure the quality of responses LangSmith Al-assisted evaluators? (Metric Development)
- What is the best dataset to gather for fine-tuning?
- How to anonymize email threads and discussion forums?



We are having a lot of fun building this thing from scratch, but if there was an open-source scientific community framework for Al Assistants, it would be even more fun!



Presentation Summary

- Our goal is to create a reliable AI assistant across all ATLAS content
- Solved(In Progress) ATLAS has significant data and presents a logistical challenge that we have largely overcome
- Complete Implemented good semantic (vector embedded) search
- Complete Prototype of Al Assistant
- Roadmap Achieving our goal by Iterative Development
- Feedback Recent Updates, Challenges, Suggestions, and Open Questions



END

