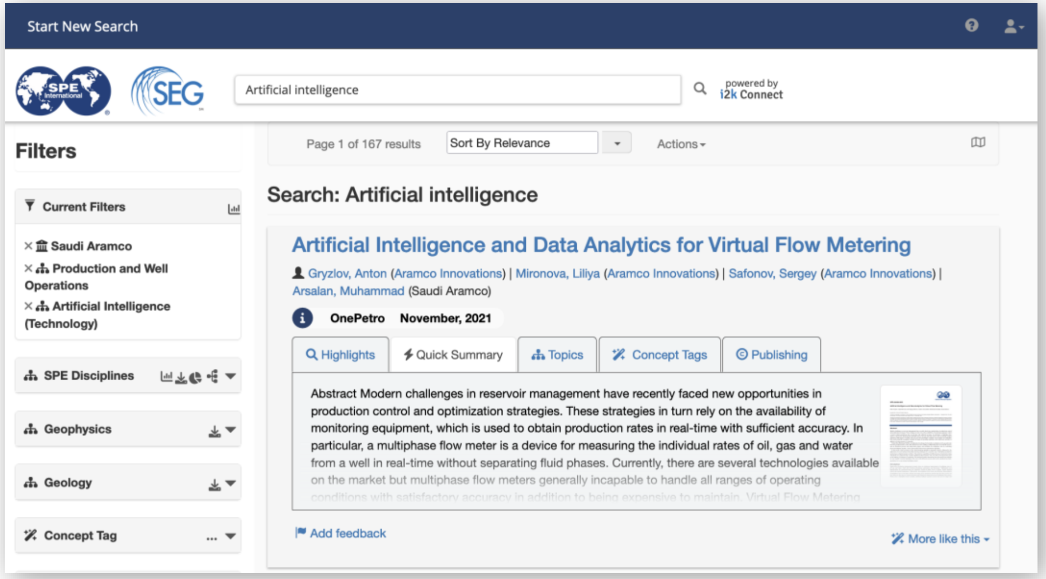
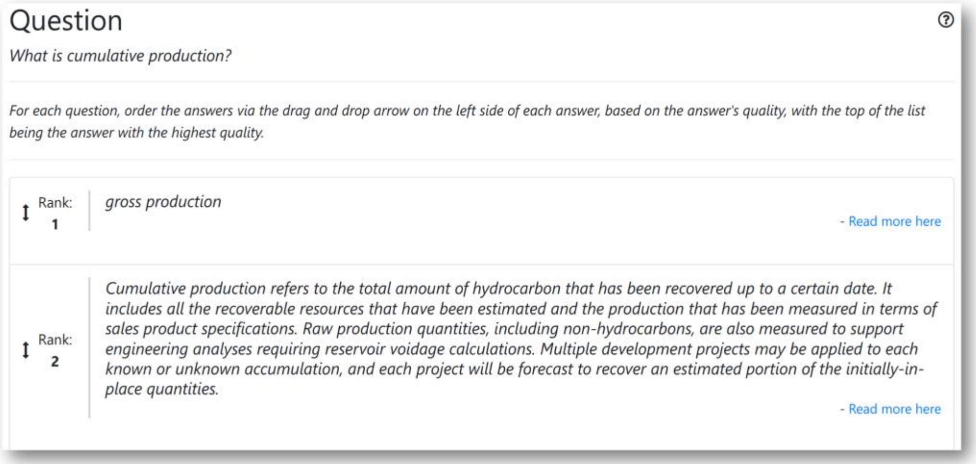
i2k Connect

- Government experience
  - Cage Code: 70GV3 and Small Business
  - DARPA HR00112190013: **Fine-Grained Knowledge Delivery in Communities of Practice** (Seedling)
  - DARPA BAA HR001121S0034: **Knowledge Management at Scale and Speed** (SRI Prime Contractor)
  - IARPA BAA W911NF-23-S-0007 **REASON** Proposer
- Relevant capabilities
  - Experience applying SOTA ML techniques to information management and discovery
  - Extensive generative AI R&D
  - Experience deploying LLMs in security-sensitive contexts
  - Advanced document processing
- Contact information
  - Becky Thomas, bthomas@i2kconnect.com





Industry-Specific Use Cases

CVX utilizes i2k Connect for analyzing upstream unstructured content to auto-classify, tag, and enrich metadata for ingestion into SharePoint. Originally, Noble Energy use case AI analysis of entire unstructured content upstream of content or asset classification and D&A purposes.

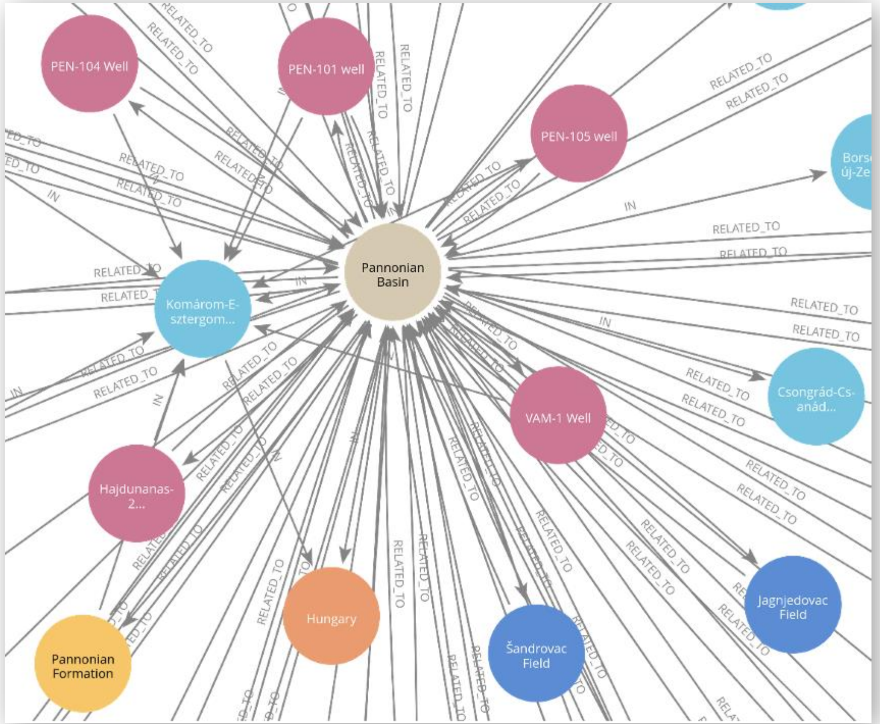
SLB has OEM'd i2k Connect to provide AI automated classifications against 18 taxonomies to provide insights to their clients within the DELFI cognitive E&P Platform.

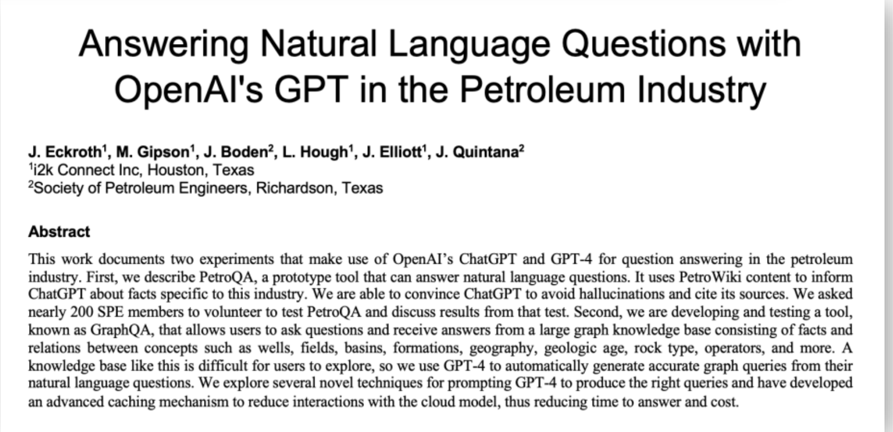
Society of Petroleum Engineers. Conducted LLM research associated with industry domain evaluating available LLM options to determine viability, accuracy, training, fine-tuning. Membership tested. i2k Connect powers the Research Portal, which enables research across all SPE, SEG, and 19 other societies' digital assets (papers, videos, etc.) at https://search.spe.org or http://search.seg.org

Woodside Energy uses enhanced search and findability of asset-related content over 46M files (437TB) in the upstream organization. Enables users to interrogate the corpus to quickly locate the right information to get their jobs done.

i2k LLM R&D

- Question answering grounded in facts
  - Sourcing answers directly from documents
  - Grounding with knowledge graphs
- Published papers, continuing research
  - QA experiments with SPE volunteers
  - Finetuning domain-specific LLMs
  - ***LLM Redact*** – LLMs on docs w/ redactions
  - ***LLM Trace*** – Framework to keep track of LLM-generated data within a system
  - ***QuoteLLM*** – Extracting training text verbatim
  - Building LLM agents





Answering Natural Language Questions with OpenAI's GPT in the Petroleum Industry

J. Eckroth[1], M. Gipson[1], J. Boden[2], L. Hough[1], J. Elliott[1], J. Quintana[2]
[1]i2k Connect Inc, Houston, Texas
[2]Society of Petroleum Engineers, Richardson, Texas

Abstract

This work documents two experiments that make use of OpenAI's ChatGPT and GPT-4 for question answering in the petroleum industry. First, we describe PetroQA, a prototype tool that can answer natural language questions. It uses PetroWiki content to inform ChatGPT about facts specific to this industry. We are able to convince ChatGPT to avoid hallucinations and cite its sources. We asked nearly 200 SPE members to volunteer to test PetroQA and discuss results from that test. Second, we are developing and testing a tool, known as GraphQA, that allows users to ask questions and receive answers from a large graph knowledge base consisting of facts and relations between concepts such as wells, fields, basins, formations, geography, geologic age, rock type, operators, and more. A knowledge base like this is difficult for users to explore, so we use GPT-4 to automatically generate accurate graph queries from their natural language questions. We explore several novel techniques for prompting GPT-4 to produce the right queries and have developed an advanced caching mechanism to reduce interactions with the cloud model, thus reducing time to answer and cost.

LLM Risk

DeepMind

Ethical and social risks of harm from Language Models

Laura Weidinger[1], John Mellor[1], Maribeth Rauh[1], Conor Griffin[1], Jonathan Uesato[1], Po-Sen Huang[1], Myra Cheng[1,2], Mia Glaese[1], Borja Balle[1], Atoosa Kasirzadeh[1,3], Zac Kenton[1], Sasha Brown[1], Will Hawkin
Stepleton[1], Courtney Biles[1], Abeba Birhane[1,4], Julia Haas[1], Laura Rimell[1], Lisa Anne Hendricks[1], W
Isaac[1], Sean Legassick[1], Geoffrey Irving[1] and Iason Gabriel[1]
d, [2]California Institute of Technology, [3]University of Toronto, [4]University College Dublin

act

per aims to help structure the risk landscape associated with large-scale Language Models (LMs). In foster advances in responsible innovation, an in-depth understanding of the potential risks posed by odels is needed. A wide range of established and anticipated risks are analysed in detail, drawing on ciplinary literature from computer science, linguistics, and social sciences.

er outlines six specific risk areas: I. Discrimination, Exclusion and Toxicity, II. Information Hazards, III. rmation Harms, IV. Malicious Uses, V. Human-Computer Interaction Harms, VI. Automation, Access, ironmental Harms.

Gender bias and stereotypes in Large Language Models

Hadas Kotek
Apple & MIT
Cupertino, CA, USA
hadas@apple.com

Rikker Dockum
Swarthmore College
Swarthmore, PA, USA
rdockum1@swarthmore.edu

David Q. Sun
Apple
Cupertino, CA, USA
dqs@apple.com

OWASP | OWASP Top 10 for LLM

OWASP Top 10 for LLM

This is a draft list of important vulnerability types for Artificial Intelligence (AI) applications built on Large Language Models (LLMs).

LLM01: Prompt Injections

Prompt Injection Vulnerabilities in LLMs involve crafty inputs leading to undetected manipulations. The impact ranges from data exposure to unauthorized actions, serving attacker's goals.

LLM02: Insecure Output Handling

These occur when plugins or apps accept LLM output without scrutiny, potentially leading to XSS, CSRF, SSRF, privilege escalation, remote code execution, and can enable agent hijacking attacks.

LLM03: Training Data Poisoning

LLMs learn from diverse text but risk training data poisoning, leading to user misinformation. Overreliance on AI is a concern. Key data sources include Common Crawl, WebText, OpenWebText, and books.

LLM04: Denial of Service

An attacker interacts with an LLM in a way that is particularly resource-consuming, causing quality of service to degrade for them and other users, or for high resource costs to be incurred.

LLM05: Supply Chain

LLM supply chains risk integrity due to vulnerabilities leading to biases, security breaches, or system failures. Issues arise from pre-trained models, crowdsourced data, and plugin extensions.

LLM06: Permission Issues

Lack of authorization tracking between plugins can enable indirect prompt injection or malicious plugin usage, leading to privilege escalation, confidentiality loss, and potential remote code execution.

LLM07: Data Leakage

Data leakage in LLMs can expose sensitive in proprietary details, leading to privacy and secu breaches. Proper data sanitization, and clear t are crucial for prevention.

LLM08: Excessive Agency

When LLMs interface with other systems, unr agency may lead to undesirable operations an Like web-apps, LLMs should not self-police; c be embedded in APIs.

LLM09: Overreliance

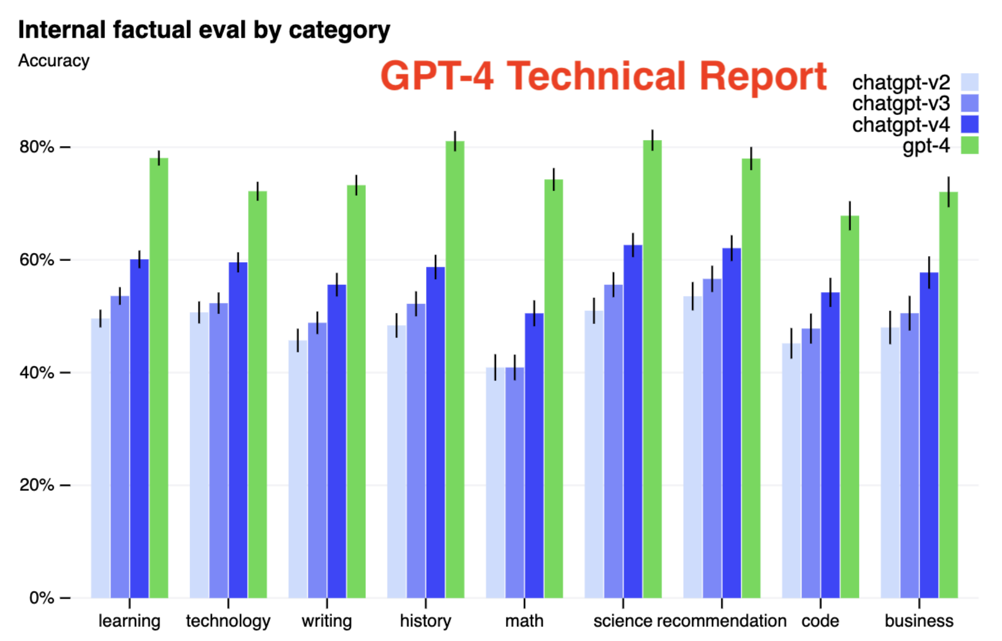
Overreliance on LLMs can lead to misinforma inappropriate content due to "hallucinations." proper oversight, this can result in legal issues reputational damage.

LLM10: Insecure Plugins

Plugins connecting LLMs to external resource exploited if they accept free-form text inputs, malicious requests that could lead to undesir or remote code execution.

LLMs as Factual Reasoners: Insights from Existing Benchmarks and Beyond

Philippe Laban Wojciech Kryściński Divyansh Agarwal Alexander R. Fabbri
Caiming Xiong Shafiq Joty Chien-Sheng Wu
Salesforce AI
{plaban, wojciech.kryscinski, dagarwal, afabbri, cxiong, sjoty, wu.jason}@salesforce.com



Abstract

ent appearance of LLMs in practi- having methods that can effectively al inconsistencies is crucial to re- opagation of misinformation and st in model outputs. When testing actual consistency benchmarks, we ew large language models (LLMs) npetitively on classification bench- ctual inconsistency detection com- ditional non-LLM methods. How- r analysis reveals that most LLMs complex formulations of the task s issues with existing evaluation , affecting the evaluation precision. this, we propose a new protocol ency detection benchmark creation

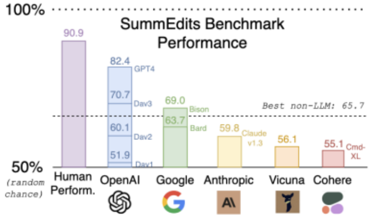


Figure 1: SUMMEDITS is a benchmark to evaluate the factual reasoning abilities of LLMs, measuring if models detect factual inconsistencies when they occur in summaries. Capable detection models can help build more reliable NLG systems.

Mitigating LLM Risk

LLM01: Prompt Injections

Prompt Injection Vulnerabilities in LLMs involve crafty inputs leading to undetected manipulations. The impact ranges from data exposure to unauthorized actions, serving attacker's goals.

LLM02: Insecure Output Handling

These occur when plugins or apps accept LLM output without scrutiny, potentially leading to XSS, CSRF, SSRF, privilege escalation, remote code execution, and can enable agent hijacking attacks.

LLM03: Training Data Poisoning

LLMs learn from diverse text but risk training data poisoning, leading to user misinformation. Overreliance on AI is a concern. Key data sources include Common Crawl, WebText, OpenWebText, and books.

LLM04: Denial of Service

An attacker interacts with an LLM in a way that is particularly resource-consuming, causing quality of service to degrade for them and other users, or for high resource costs to be incurred.

LLM05: Supply Chain

LLM supply chains risk integrity due to vulnerabilities leading to biases, security breaches, or system failures. Issues arise from pre-trained models, crowdsourced data, and plugin extensions.

Answering Natural Language Questions with OpenAI's GPT in the Petroleum Industry

J. Eckroth[1], M. Gipson[1], J. Boden[2], L. Hough[1], J. Elliott[1], J. Quintana[2]
[1]i2k Connect Inc, Houston, Texas
[2]Society of Petroleum Engineers, Richardson, Texas

QuoteLLM Research Project

LLM Redact Research Project

Finetuning Domain-Specific LLMs

LLM Trace Research Project

LLM06: Permission Issues

Lack of authorization tracking between plugins can enable indirect prompt injection or malicious plugin usage, leading to privilege escalation, confidentiality loss, and potential remote code execution.

LLM07: Data Leakage

Data leakage in LLMs can expose sensitive information or proprietary details, leading to privacy and security breaches. Proper data sanitization, and clear terms of use are crucial for prevention.

LLM08: Excessive Agency

When LLMs interface with other systems, unrestricted agency may lead to undesirable operations and actions. Like web-apps, LLMs should not self-police; controls must be embedded in APIs.

LLM09: Overreliance

Overreliance on LLMs can lead to misinformation or inappropriate content due to "hallucinations." Without proper oversight, this can result in legal issues and reputational damage.

LLM10: Insecure Plugins

Plugins connecting LLMs to external resources can be exploited if they accept free-form text inputs, enabling malicious requests that could lead to undesired behaviors or remote code execution.