

Workshop Track 1

Introducing AI for LAMs: A Beginner Tutorial for Practical Generative AI Use Cases

**Fantastic Futures 2023 Annual International
Conference**

Yinlin Chen

ylchen@vt.edu

Assistant Director, Center for Digital Research & Scholarship
Virginia Tech Libraries

Today's Topics

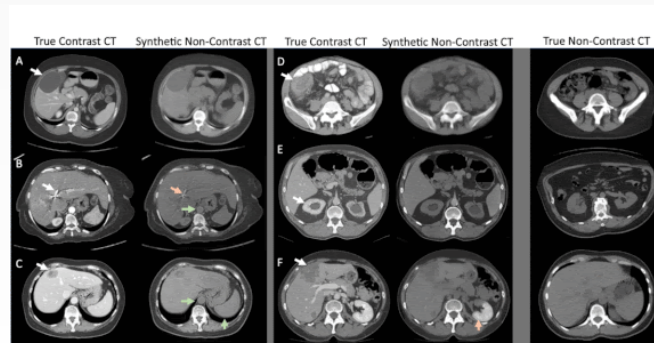
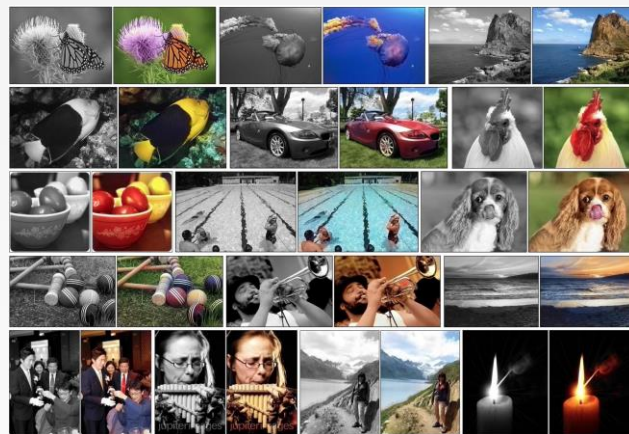
- Introduction
- Generative AI
- Large Language Models (LLMs)
- Applications
- Demonstration
- Hands-On Exercise
- Discussion
- Q & A

Welcome & Introduction

- Introduce yourselves
 - Name, role and institution
- Experience with LAMs
- Experience with AI and machine learning
- Experience with language models
- What outcomes are you hoping for from this workshop?

Generative Adversarial Network

- GANs are powerful machine learning models capable of generating realistic image, video, and voice outputs
- Applications:
 - Transform photo editing
 - Generate special effects for media and entertainment
 - Contribute to creative products from industrial design to fine art
 - Augment datasets in small data problems in fields from autonomous driving to manufacturing



Brats mri axial slices generative diffusion

[Download](#)

Model Metadata:

Overview: A generative model for creating 2D brain MRI axial slices from Gaussian noise based on BraTS dataset

Author(s): MONAI team

Downloads: 347

File Size: 99.7MB

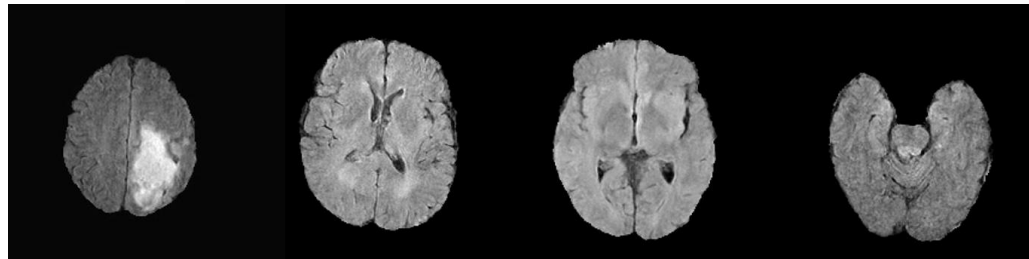
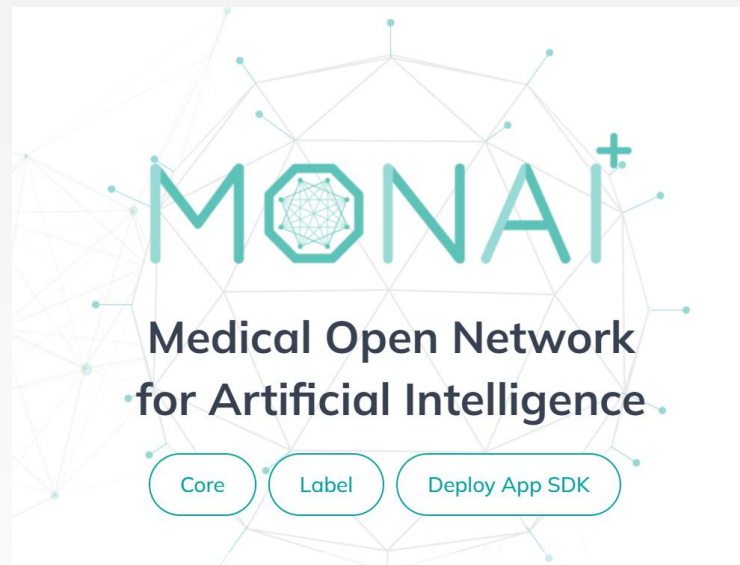
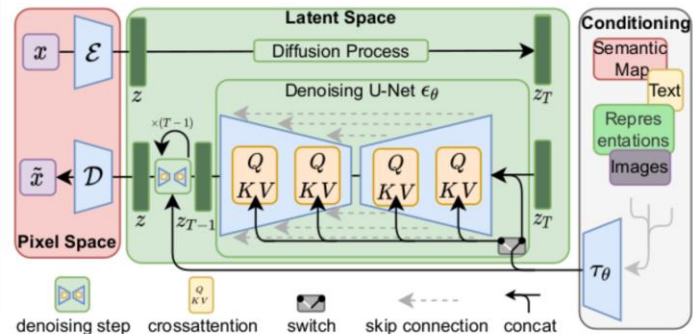
Version: 1.0.6

Model README:

Model Overview

A pre-trained model for 2D Latent Diffusion Generative Model on axial slices of BraTS MRI.

This model is trained on BraTS 2016 and 2017 data from [Medical Decathlon](#), using the Latent diffusion model [1].



AI can now
create images
out of thin air.

See how it works.

AI



Masked Autoencoders (MAE)

Paper: [Masked Autoencoders Are Scalable Vision Learners](#)

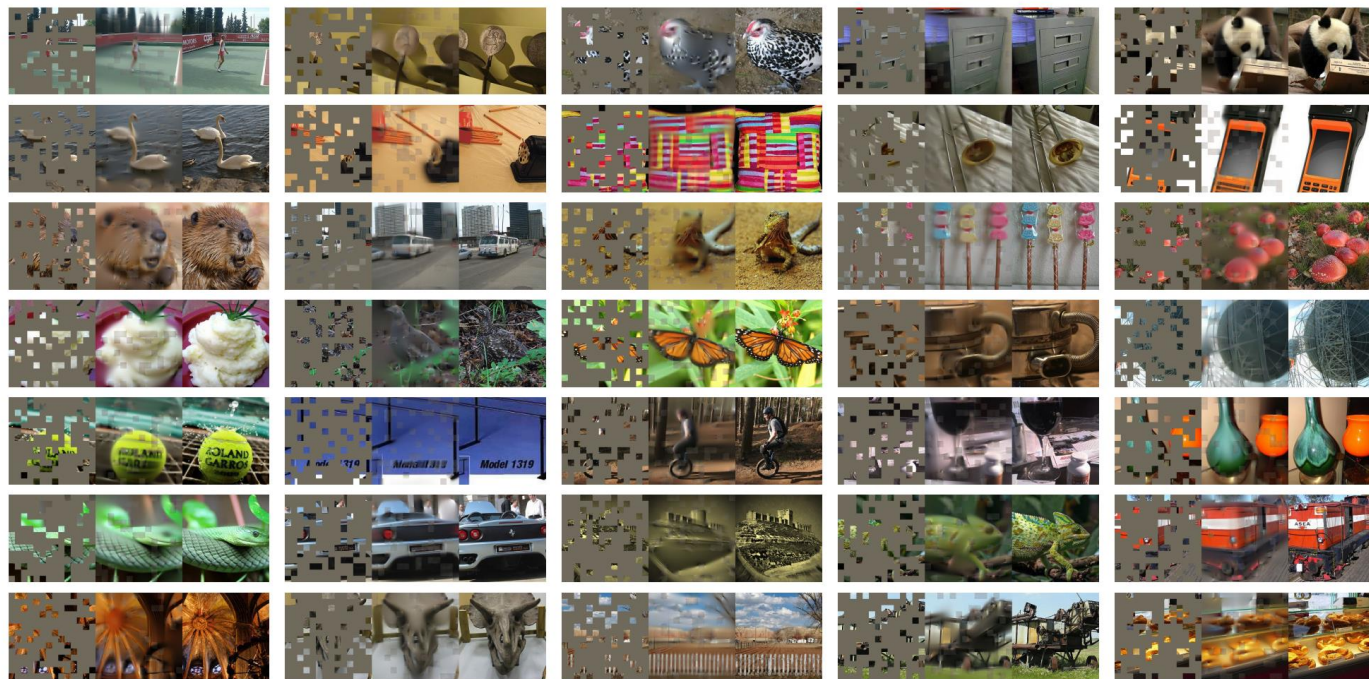


Figure 10. **Uncurated random samples** on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction (middle), and the ground-truth (right). The masking ratio is 75%.

Generative Video / Audio models



Image source: The Book of Boba Fett | Chapter 6

SONANTIC



HELPING ACTOR VAL
KILMER RECLAIM HIS
VOICE

Language Model

- A machine agent that understands and generates human language
- A subset of Nature Language Processing (NLP) focused on text generation and comprehension
- Uses statistical and machine learning techniques to predict and produce language sequences
- Acts as a tool within NLP for various applications like chatbots, translation services, and virtual assistants

Nature Language Processing (NLP)

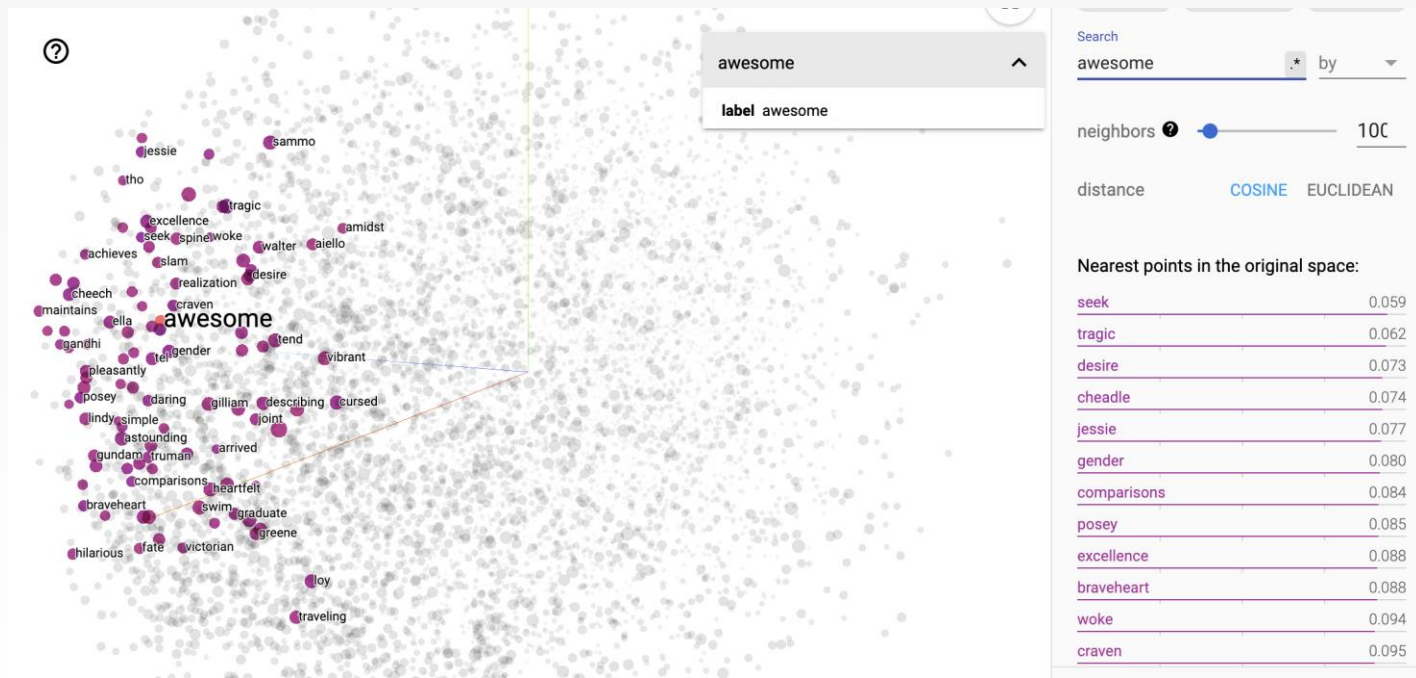
- Enables machines to understand, interpret, and generate human language
- Facilitate human-computer interaction: E.g. Alexa, Siri, Google Home, ChatGPT
- Empowers language tools: Assists in language acquisition and translations
- Drives linguistic research: Analyzes datasets to uncover linguistic patterns and trends

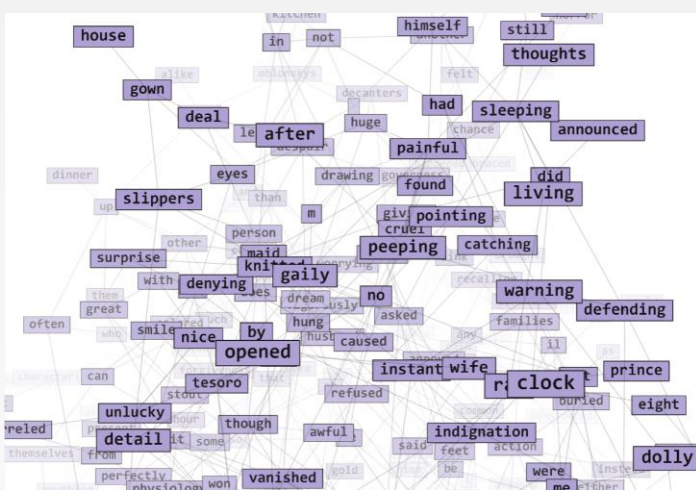
NLP Tasks

- Machine translation
- Question Answering
- Sentiment Analysis
- Named Entity Recognition (NER)
- Text Summarization
- Text Generation
- ...

Word Embedding

- Use an efficient, dense representation in which similar words have a similar encoding





-0.31529	-0.02843	-0.016746	-0.021209	0.079952	0.31622	0.064761	-0.2
0.14231	0.94446	-0.36723	-0.63264	-0.22749	0.47908	-0.66815	-0.04130
-0.20977	-0.24609	-0.40151	0.47429	0.010771	0.82454	0.33783	0.12811
-0.025396	0.48878	-0.065962	-0.12585	-0.67172	0.71344	-0.52369	-0.68
-0.27753	0.30311	-0.51805	0.28916	-0.26186	-0.009617	-0.40885	0.2119
-0.31982	0.095621	-0.32895	-0.55728	0.32756	0.86118	0.33973	0.14936
-0.049021	-0.39374	-0.17264	-0.099287	0.24752	0.48518	-0.83403	0.072
0.028308	0.053201	0.036348	-0.22297	0.064423	0.159	-0.77674	0.30433
0.13857	0.5047	0.57522	0.17839	0.41169	-0.33658	0.18725	-0.46583
-0.37804	1.0744	0.42999	-0.80921	0.88273	0.94068	0.56096	-0.075746
0.45339	0.006271	0.79116	-1.1023	0.37591	0.44677	0.049367	-0.39339
-0.51627	0.837	-0.58194	0.046615	-0.3937	0.66203	0.48482	-0.17065
-0.058123	0.63951	0.77633	0.35478	0.14914	-0.65503	0.52402	-0.53852
0.16984	-0.33992	0.55786	0.38667	-0.84554	0.28118	1.1596	-0.70108
-0.42704	-0.89916	-0.017155	-0.18568	-0.51684	0.56688	-0.051546	-0.6

What is a **tidy** thing to eat pasta with?

A tidy thing to eat pasta with is

What is a **nice** thing to eat pasta with?

A nice thing to eat pasta with is

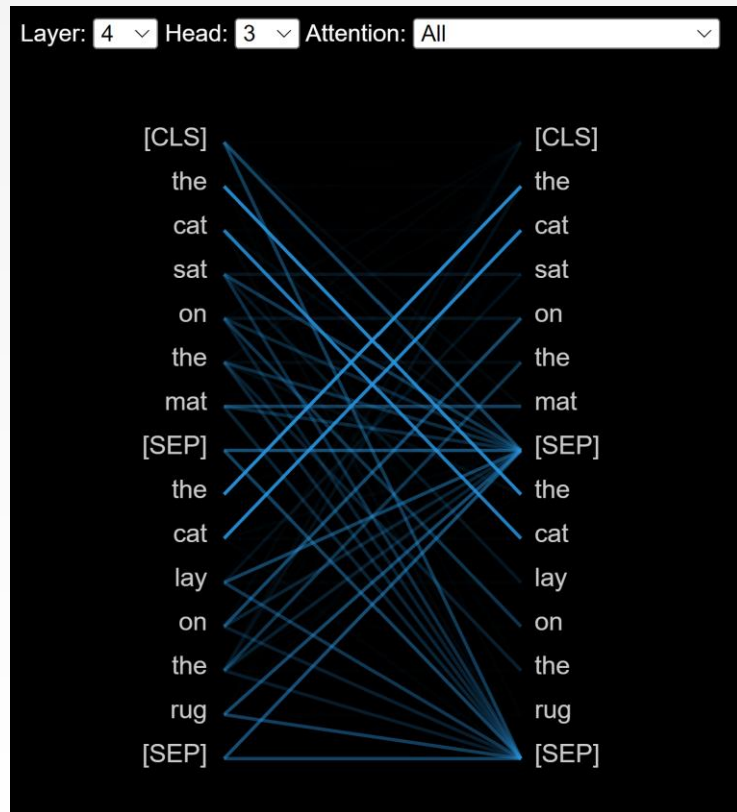
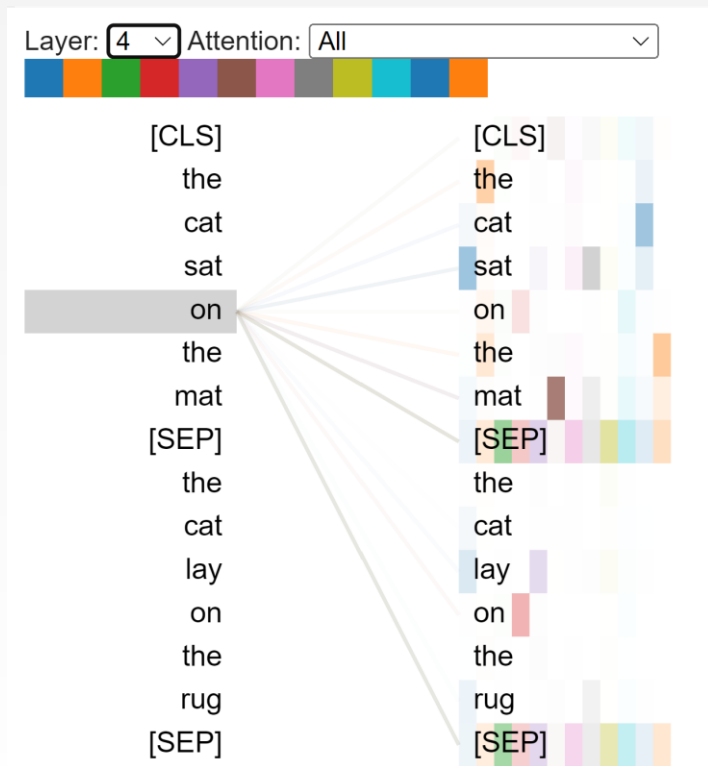
A transformer is a mathematical process that recalculates the **vectors** for each **token**; that is, it assigns new distances between each pair of **tokens**, based largely on what other **tokens** are in its utterance. If you ask "What is a tidy thing to eat pasta with?" and (in a different chat ...) "What is a nice thing to eat pasta with?" the LLM will start typing its answer...

* <https://www.theguardian.com/technology/ng-interactive/2023/nov/01/how-ai-chatbots-like-chatgpt-or-bard-work-visual-explainer>

Large Language Models (LLMs)

- Predicts the probability of word sequences to generate text that mimics human speech and writing
- Trained on extensive textual data, these models excel in predicting the next word in a sentence based on the preceding words
- Integrated to enhancing the functionality of applications like chatbots, creative writing tools, translation software, and platforms that analyze customer sentiment
- Examples of such models include OpenAI's GPT-3 and GPT-4, as well as Google's BERT, among others

Transformer: Attention Is All You Need*



* <https://dl.acm.org/doi/10.5555/3295222.3295349>

Image from: BertViz: <https://github.com/jessevig/bertviz>

AlphaCode Attention Visualization

Hover over tokens in the solution to see which tokens the model attended to when generating the solution. Click a token to select it; clicking in empty space will deselect.

Solutions were selected randomly, keeping at most one correct (passes all test cases in our dataset) and one incorrect sample per problem and language. Note that since our dataset only has a limited number of test cases, passing all tests we have cannot completely rule out false positives (~4%), or solutions that are correct but inefficient (~42%).

Check out selected problems with commentary from World-Class Competitive Programmer Petr Mitrichev: [1566_E](#) [1591_C](#) [1618_B](#) [1618_E](#) [1619_D](#) [1623_B](#)

Read our [paper](#) and [blog post](#) for more.

1553_H. XOR and Distance

Python

fail

Layer 18

play

✓ Head 1

✓ Head 2

✓ Head 3

✓ Head 4

✓ Head 5

✓ Head 6

✓ Head 7

✓ Head 8

✓ Head 9

✓ Head 10

✓ Head 11

all

none

Problem Description

```
# RATING: 1200
# TAGS: dfs and similar, graphs, trees
# LANGUAGE IS python3
# CORRECT SOLUTION
# You are given an array a consisting of n distinct elements and an integer k.
# Each element in the array is a non-negative integer not exceeding  $2^k-1$ .
#
# Let's define the XOR distance for a number x as the value of
#
# $$$f(x) = \min_{i=1}^n \min_{j=i+1}^n |(a_i \oplus x) - (a_j \oplus x)|$$$
#
# where  $\oplus$  denotes [the bitwise XOR
# operation](https://en.wikipedia.org/wiki/Bitwise_operation#XOR).
#
# For every integer x from 0 to  $2^k-1$ , you have to calculate f(x).
#
# Input
#
# The first line contains two integers n and k ( $1 \leq k \leq 19$ ;  $2 \leq n \leq 2^k$ ).
#
# The second line contains n integers a_1, a_2, ..., a_n ( $0 \leq a_i \leq 2^k-1$ ). All
# these integers are distinct.
#
# Output
```

Solution

```
n, k = map(int, input().split())
a = list(map(int, input().split()))

a_s = sorted(a)
a_size = len(a)

for i in range(2 ** k):
    b = [0] * a_size
    for j in range(a_size):
        b[j] = a[j] ^ i

    b_s = sorted(b)

    b_size = len(b)
    a_size = b_size

    min_d = 10 ** 9
    for j in range(b_size - 1):
        min_d = min(min_d, b_s[j + 1] - b_s[j])

    print(min_d, end=" ")
```

LLMs for Human-Computer Interaction

- Interpret and respond to natural language input.
- Engage in dynamic dialogues with users.
- Maintain context over multi-turn conversations.
- Provide personalized responses based on user preferences.
- Offer real-time language translation and multilingual support.
- Integrate with external systems for enhanced interactivity.
- Learn and adapt from user interactions to improve over time.

< 2 / 2 >



Brief Introduce Yinlin Chen at Virginia Tech



Finished browsing



Yinlin Chen is an Assistant Professor and also serves as the Assistant Director of the Center for Digital Research & Scholarship at Virginia Tech, where he is based at the 5004 Newman Library¹². He holds the title of Digital Library Architect within the University Libraries at Virginia Tech, and his educational background includes a Ph.D. in Computer Science and Applications from Virginia Tech, along with a M.S. and B.S. in Computer Science from National Tsing Hua University, Taiwan³.

In terms of teaching, Yinlin Chen is involved in the course "Introduction to AI" at Virginia Tech, where he holds office hours for students to consult with him on course material. His office hours are specified for Mondays from 10-11 am and Fridays from 10-11 am, with additional availability by appointment. His office is located in Newman Library 5004, and virtual consultations via Zoom are also an option^{4 5}.

His professional and academic endeavors span a variety of areas including digital libraries, cloud computing, machine learning, and software architecture in cloud-based or hybrid technology solutions.

Is this conversation helpful so far?

Regenerate

Send a message



GPT-3.5 / GPT-4 Models

gpt-3.5-turbo	Most capable GPT-3.5 model and optimized for chat at 1/10th the cost of text-davinci-003. Will be updated with our latest model iteration 2 weeks after it is released.	4,097 tokens	Up to Sep 2021
gpt-3.5-turbo-16k	Same capabilities as the standard gpt-3.5-turbo model but with 4 times the context.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-instruct	Similar capabilities as text-davinci-003 but compatible with legacy Completions endpoint and not Chat Completions.	4,097 tokens	Up to Sep 2021
gpt-3.5-turbo-0613	Snapshot of gpt-3.5-turbo from June 13th 2023 with function calling data. Unlike gpt-3.5-turbo, this model will not receive updates, and will be deprecated 3 months after a new version is released.	4,097 tokens	Up to Sep 2021
gpt-3.5-turbo-16k-0613	Snapshot of gpt-3.5-turbo-16k from June 13th 2023. Unlike gpt-3.5-turbo-16k, this model will not receive updates, and will be deprecated 3 months after a new version is released.	16,385 tokens	Up to Sep 2021

LATEST MODEL	DESCRIPTION	MAX TOKENS	TRAINING DATA
gpt-4	More capable than any GPT-3.5 model, able to do more complex tasks, and optimized for chat. Will be updated with our latest model iteration 2 weeks after it is released.	8,192 tokens	Up to Sep 2021
gpt-4-0613	Snapshot of gpt-4 from June 13th 2023 with function calling data. Unlike gpt-4, this model will not receive updates, and will be deprecated 3 months after a new version is released.	8,192 tokens	Up to Sep 2021
gpt-4-32k	Same capabilities as the standard gpt-4 mode but with 4x the context length. Will be updated with our latest model iteration.	32,768 tokens	Up to Sep 2021
gpt-4-32k-0613	Snapshot of gpt-4-32 from June 13th 2023. Unlike gpt-4-32k, this model will not receive updates, and will be deprecated 3 months after a new version is released.	32,768 tokens	Up to Sep 2021

Source: <https://openai.com/pricing#language-models>

Model Benchmark

Benchmark	Grok-0 (33B)	LLaMa 2 70B	Inflection-1	GPT-3.5	Grok-1	Palm 2	Claude 2	GPT-4
GSM8k	56.8%	56.8%	62.9%	57.1%	62.9%	80.7%	88.0%	92.0%
	8-shot	8-shot	8-shot	8-shot	8-shot	8-shot	8-shot	8-shot
MMLU	65.7%	68.9%	72.7%	70.0%	73.0%	78.0%	75.0%	86.4%
	5-shot	5-shot	5-shot	5-shot	5-shot	5-shot	5-shot + CoT	5-shot
HumanEval	39.7%	29.9%	35.4%	48.1%	63.2%	-	70%	67%
	0-shot	0-shot	0-shot	0-shot	0-shot		0-shot	0-shot
MATH	15.7%	13.5%	16.0%	23.5%	23.9%	34.6%	-	42.5%
	4-shot	4-shot	4-shot	4-shot	4-shot	4-shot		4-shot

Open LLM Leaderboard

🚩 The 🤖 Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

🤖 Submit a model for automated evaluation on the 🤖 GPU cluster on the "Submit" page! The leaderboard's backend runs the great [Eleuther AI Language Model Evaluation Harness](#) - read more details in the "About" page!

🏆 LLM Benchmark 📈 Metrics evolution through time 📄 About 🚀 Submit here!

🔍 Search for your model (separate multiple queries with `;`) and press ENTER...

Select columns to show

☒ Average **1** ☒ ARC ☒ HellaSwag ☒ MMLU ☒ TruthfulQA ☐ Type

☐ Precision ☐ Hub License ☐ #Params (B) ☐ Hub ❤️ ☐ Model sha

☒ Show gated/private/deleted models

Model types

☒  pretrained ☒  fine-tuned ☒  instruction-tuned ☒  RL-tuned ☒ ?

Precision

☒ torch.float16 ☒ torch.bfloat16 ☒ torch.float32 ☒ 8bit ☒ 4bit ☒ GPTQ

Model sizes (in billions of parameters)

☒ ? ☒ ~1.5 ☒ ~3 ☒ ~7 ☒ ~13 ☒ ~35 ☒ ~60 ☒ 70+

T	Model	Average 1	ARC	HellaSwag	MMLU	TruthfulQA
📌	ValiantLabs/ShiningValiant 📄	74.17	72.95	87.88	70.97	64.88
📌	ICBU-NPU/FashionGPT-70B-V1.2 📄	74.11	73.04	88.15	70.11	65.15
📌	sequelbox/StellarxBright 📄	74.1	72.95	87.82	71.17	64.46
📌	Riiid/sheep-duck-llama-2-70b-v1.1 📄	74.07	73.04	87.81	70.84	64.58
📌	AIDC-ai-business/Marcoroni-70B-v1 📄	74.06	73.55	87.62	70.67	64.41

Source: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Input-Output Workflow in LLMs



Prompt (Input)

- Serves as the input method for Language Models
- **Describes** the task to be performed by the model
- Supplies necessary **context** for generating relevant responses
- May include **formatting instructions** for the output
- Specifies a **role** to instruct LLMs on behavior or perspective
- May contain examples to **guide** response generation

Prompt Techniques

- **Prompt-Based:** User provides a "prompt" or initial input, and the model generates a continuation.
- **Zero-Shot:** Model makes predictions about a task without seeing examples of the task during training.
- **Few-Shot:** User provides a few examples of the desired task within the prompt, and the model generalizes from these examples to complete the task.
- **Chain-of-Thought (CoT):** Enables reasoning via intermediate steps. When combined with few-shot prompting, it improves results on tasks needing complex reasoning.

LLMs Limitations

- Do not have information on events or developments that occurred after their most recent training data cut-off
- Output quality is highly dependent on the phrasing and specificity of the input prompt
- Produce confident but inaccurate response, a phenomenon sometimes called 'hallucination'
- Operates purely on prediction, there is no guarantee that its responses will always be accurate

Applications



CSAI Study Buddy

I'm a study assistant here to help you understand
your content.

By Yinlin Chen

Summarize this paper.

Clarify this paragraph.

Explain this concept.

Detail this topic.



Message CSAI Study Buddy...



ChatGPT can make mistakes. Consider checking important information.



You

how many topics do you have



CSAI Study Buddy

The topics covered in the "Machine Learning II" course, instructed by Yinlin Chen in Fall 2023 at Virginia Tech, include:

1. Decision Tree
2. Naïve Bayes
3. Parameter Estimation
4. Maximum Likelihood
5. Laplace Smoothing
6. Perceptron.



You

explain what is MDP



CSAI Study Buddy

A Markov Decision Process (MDP) is a mathematical framework used for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning.

MDPs are characterized by the following components:

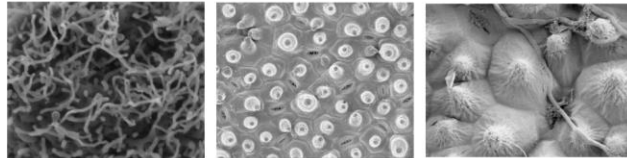
1. **States (S):** These are the distinct situation ↓ configurations that the agent or decision-maker can be in. The set of all possible states is called the state space.

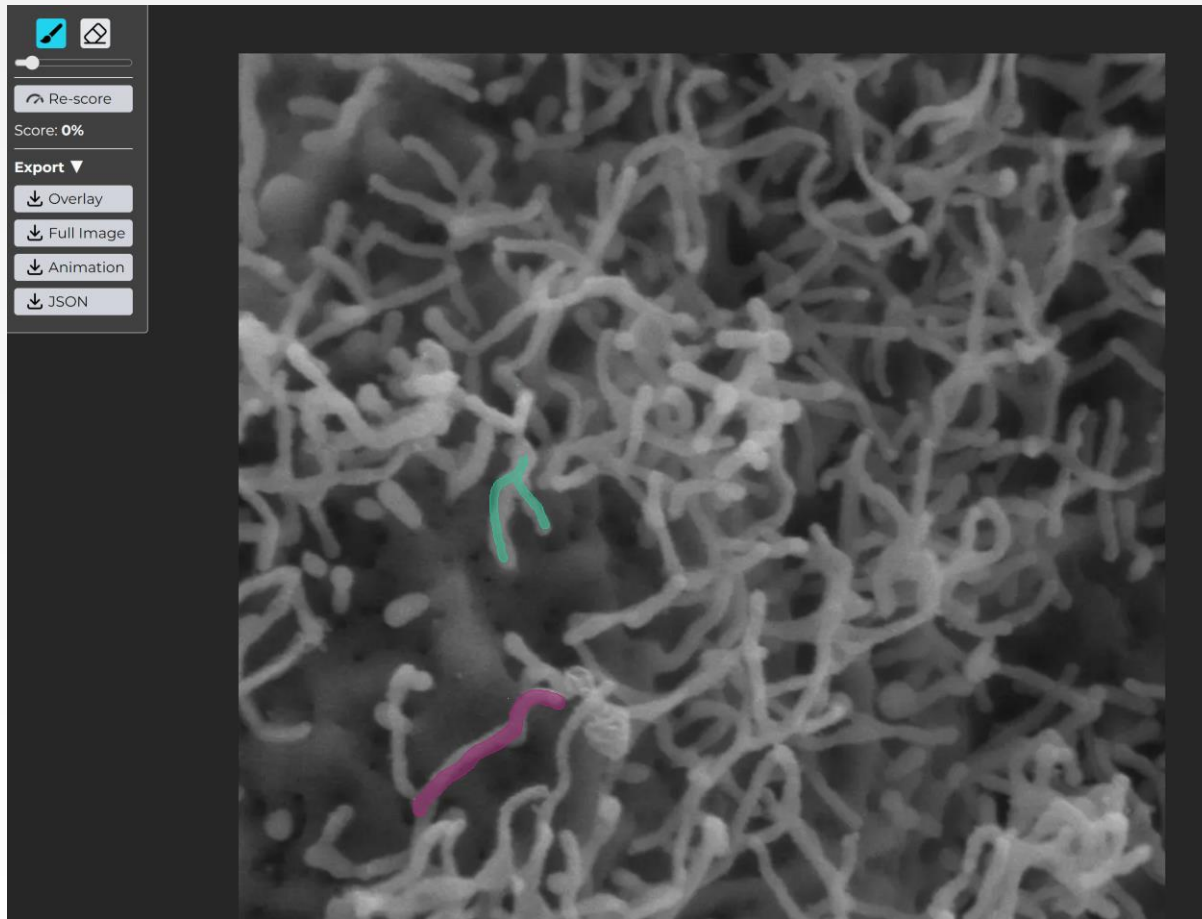
Behind Density Lines: Machine Learning and Citizen Scientists in Quantifying Scanning Electron Microscopy Images

Monét Roberts¹ Yinlin Chen² Tanner Upthegrove³

¹Department of Biomedical Engineering and Mechanics, ²University Libraries, ³Institute for Creativity, Arts, and Technology
Virginia Tech, Blacksburg, VA, USA

Amid the pandemic, screen interactions have surged by 60-80%, accentuating digital eye strain, attention deficits, and posture issues. This study addresses a key challenge in image analysis post-acquisition, especially in scanning electron microscopy (SEM) critical for health and material sciences applications. For instance, surface morphologies in cancer research and fiber quantification in biomaterials demand precise analysis, often hindered by time constraints and human error. This project proposes a machine learning workflow to accurately quantify SEM images, leveraging previously quantified SEM images to create a high-quality training dataset. Concurrently, a community engagement pipeline is devised to involve citizen scientists in SEM analysis, beginning with a pilot study and extending to interactive sessions at the Virginia Tech Science Festival. This dual approach aims to bolster SEM analysis accuracy and foster a reciprocal educational ecosystem, setting a strong foundation for future machine learning applications and community-driven scientific explorations.







When hovering over the image, SAM is running in the browser.

Tools

[+ Upload](#)[Gallery](#)

Hover & Click

Click an object one or more times.
Shift-click to remove regions.

**Add Mask**

Remove Area

Reset

Undo

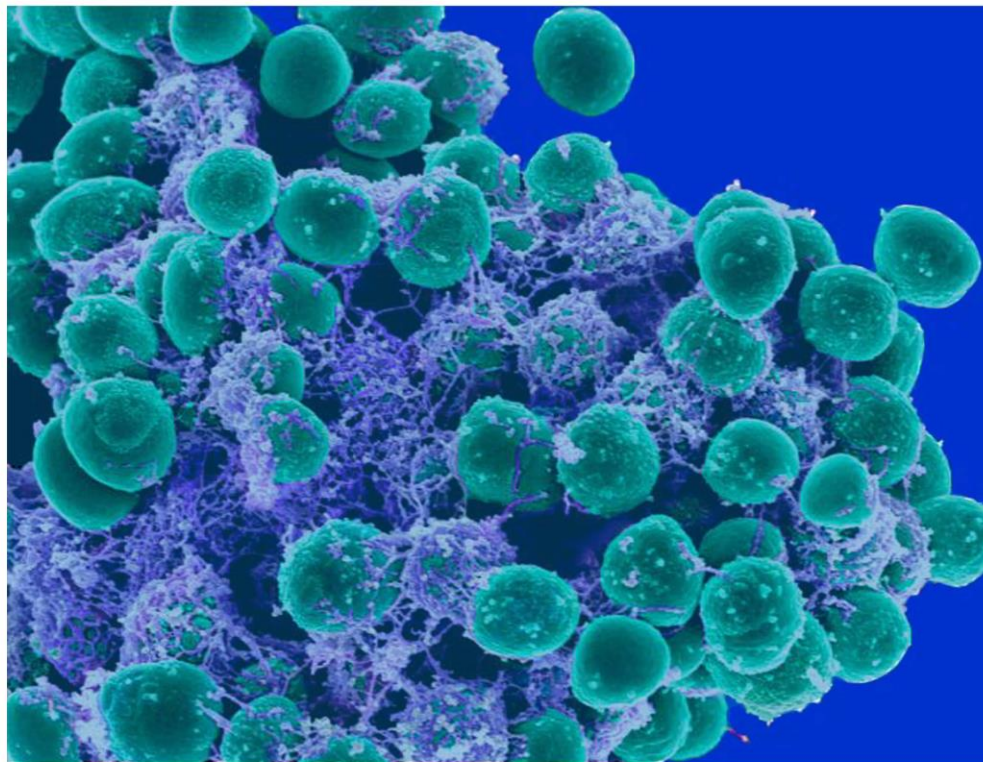
Redo



Multi-mask



Cut out object

**Box****Everything****Cut-Outs**

Leveraging Large Language Models to Understand the Financial Behavior of Women and Minority Groups

Patti J. Fisher¹

Yinlin Chen²

¹Department of Apparel, Housing, and Resource Management and ²University Libraries
Virginia Tech, Blacksburg, VA, USA

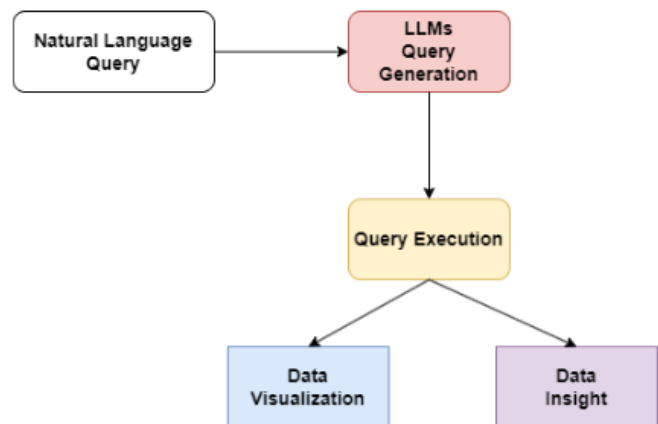
Our project represents a pioneering effort in financial data analysis, focusing on the financial behaviors of women and minority groups, using large language models (LLMs) like OpenAI GPT4 in natural language processing (NLP) tasks. By developing a user-friendly interface to analyze the Survey of Consumer Finances (SCF) data, we aim to highlight potential disparities in financial decision-making within these communities. This initiative combines the expertise of specialists in financial behavior and artificial intelligence, utilizing transformer and foundation models for data processing and analysis. Our anticipated outcomes include offering valuable insights to financial institutions and policymakers, promoting financial inclusion, and contributing significantly to research on economic inequality.

Research Question

The proposed project addresses these research questions:

- Which respondents have high financial risk tolerance?
- Which women have high financial risk tolerance?
- Which population subgroups are not willing to take any financial risks?
- Who are the “poorest poor” in the United States (bottom 5% of wealth and income)?
- What are the household characteristics of those who save regularly?"

The exploration of these questions is significantly relevant to the broader research context of financial behavior and economic inequality [1]. The SCF is a detailed triennial survey encompassing US households' financial characteristics, including income, net worth, liabilities, and assets. Using this data, researchers can gain insights into financial behavior trends, thereby informing policy decisions related to consumer finance and economic inequality [2].



LLMs for Semantic Web Interactions

- Integrate with SPARQL endpoints to access the latest linked data
- Apply constraints from SPARQL queries to anchor responses in factual context
- Include mechanisms to accurately cite the sources of their provided information
- Tailored to interpret natural language input and produce natural language output, enhancing user interaction

Semantic Web

- Aims for seamless data sharing and reuse across various applications, facilitating a more intelligent and responsive web experience.
- Establishes a "web of data" with well-defined meaning, enabling machines to interpret web content beyond mere keyword matching.
- Driven by W3C standards, the initiative fosters advanced data integration and interoperability, connecting data across domains and communities.
- Utilizes frameworks like RDF, OWL, and SPARQL, which provide a structured way to describe relationships between things and to query that data.

Linked Data

- Method for publishing and interlinking Resource Description Framework (RDF) data on the web.
- Can be accessed using URIs, HTTP(S) and RESTful APIs, SPARQL, and Semantic Web standards.
- Enables navigation between data sources using self-described RDF links.
- Facilitates crawling of the Semantic Web by search engine robots.

Accessing Linked Data

- **SPARQL Endpoint:** A service conformant with the SPARQL protocol for processing SPARQL queries.
- **Linked Data APIs:** Web services facilitating access to linked data. E.g., DBpedia Lookup API.
- **Linked Data Libraries and Frameworks:** Software tools aiding in parsing, querying, and manipulating linked data, facilitating its use.

DBpedia

- Implements Linked Data, enhancing dataset accessibility.
- Extracts data from Wikipedia and transforms it into accessible URIs.
- URIs mirror Wikipedia's structure, maintaining consistency.
- Hosts multilingual labels and abstracts, broadening global reach.
- Access via SPARQL queries, allowing for sophisticated data retrieval.

SPARQL Endpoint

SPARQL Query Editor About Tables ▾

Conductor Facet Browser Permalink

Extensions: cxml save to dav sponge User: SPARQL

Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

```
SELECT ?person ?personLabel WHERE {  
  ?person dbo:award dbr:Nobel_Prize_in_Literature.  
  ?person rdfs:label ?personLabel.  
  FILTER (lang(?personLabel) = 'en')  
}
```

Results Format HTML ▾

Execute Query Reset

Execution timeout 10000 milliseconds

Options

☒ Strict checking of void variables

☒ Strict checking of variable names used in multiple clauses but not logically connected to each other

DBpedia Lookup API



Documentation at <https://github.com/dbpedia/dbpedia-lookup>
Search API at <https://lookup.dbpedia.org/api/search?query=Leipzig>
Auto-Complete API at <https://lookup.dbpedia.org/api/prefix?query=Leipzig>

Test it here:

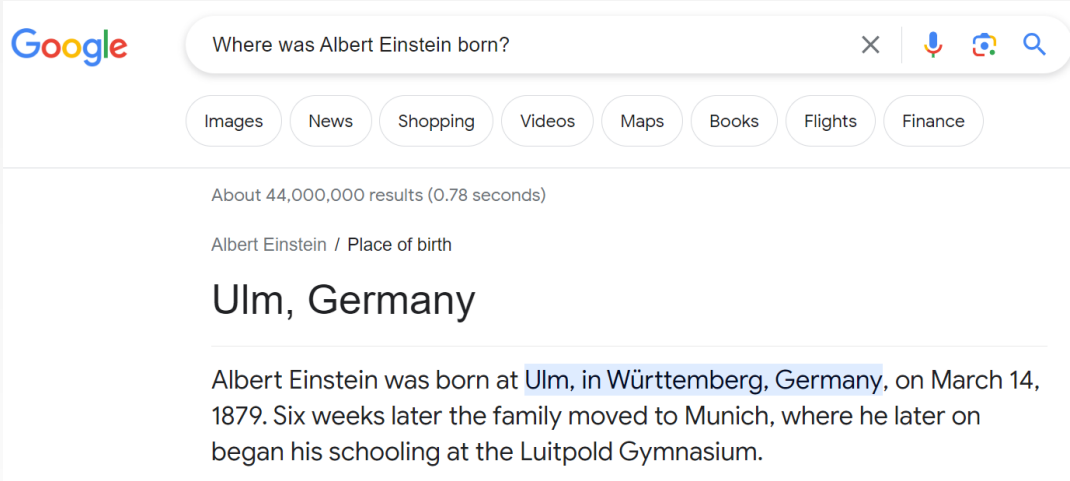
Where was Albert Einstein born?

Search

Top 10 Results:

Eastern European Time - http://dbpedia.org/resource/Eastern_European_Time
Crambidae - <http://dbpedia.org/resource/Crambidae>
Sweden - <http://dbpedia.org/resource/Sweden>
Switzerland - <http://dbpedia.org/resource/Switzerland>
Soul music - http://dbpedia.org/resource/Soul_music
Nigeria - <http://dbpedia.org/resource/Nigeria>
George Hampson - http://dbpedia.org/resource/George_Hampson
Kensington - <http://dbpedia.org/resource/Kensington>
ETH Zurich - http://dbpedia.org/resource/ETH_Zurich
Stockholm - <http://dbpedia.org/resource/Stockholm>

Where was Albert Einstein born?



Google search results for "Where was Albert Einstein born?". The search bar shows the query and icons for voice search, image search, and a magnifying glass. Below the search bar are tabs for Images, News, Shopping, Videos, Maps, Books, Flights, and Finance. The results show "About 44,000,000 results (0.78 seconds)" and "Albert Einstein / Place of birth". The main result is "Ulm, Germany". A snippet below states: "Albert Einstein was born at Ulm, in Württemberg, Germany, on March 14, 1879. Six weeks later the family moved to Munich, where he later on began his schooling at the Luitpold Gymnasium."

Where was Albert Einstein born?

Images News Shopping Videos Maps Books Flights Finance

About 44,000,000 results (0.78 seconds)

Albert Einstein / Place of birth

Ulm, Germany

Albert Einstein was born at Ulm, in Württemberg, Germany, on March 14, 1879. Six weeks later the family moved to Munich, where he later on began his schooling at the Luitpold Gymnasium.



Wikipedia

https://en.wikipedia.org/wiki/Albert_Einstein

Albert Einstein

Albert Einstein was born in **Ulm, in the Kingdom of Württemberg in the German Empire**, on 14 March 1879. ... His parents, secular Ashkenazi Jews, were Hermann ...

Born: 14 March 1879; **Ulm, Kingdom of Württe...** Citizenship: **Kingdom of Württemberg**, part ...

Education: Federal polytechnic school in Züric... Died: 18 April 1955 (aged 76); **Princeton, N...**



Where was Albert Einstein born?



Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire, on 14 March 1879.



Chatbot

Where was Albert Einstein born?

Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire.

Type a message...

Submit

Retry

Undo

Clear

Demonstration

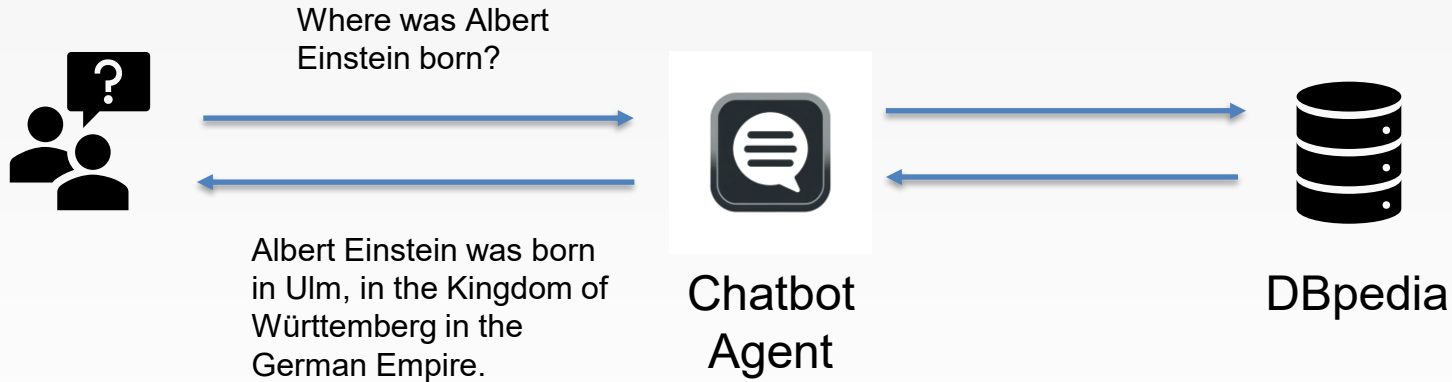
Example Questions

1. What is the population of New York City?
2. What is the capital city of Canada?
3. Who is the author of The Lord of the Rings?
4. When was Barack Obama born?
5. Which river is the longest in the world?
6. Who created DBpedia?

LLMs for Semantic Web Query

- Understand user intent and formulate semantic web queries
- Enhanced query interpretation and generation
- Generate structured queries (e.g., SPARQL) for precise information searching
- Improve query accuracy and relevance through semantic understanding

User Query to Chatbot Response



Delegate Complexity to LLMs

```
(projects) [ubuntu@yinlin LLMs (main X)]$ python openaidbpedia.py
Enter your question: Where was Albert Einstein born?
SELECT ?birthPlace WHERE {
    dbr:Albert_Einstein dbo:birthPlace ?birthPlace .
}
birthPlace: http://dbpedia.org/resource/Ulm
birthPlace: http://dbpedia.org/resource/German_Empire
birthPlace: http://dbpedia.org/resource/Kingdom_of_Württemberg
```

Response: Albert Einstein was born in Ulm, in the Kingdom of Württemberg in the German Empire.

Crafting Prompt

Model: GPT-4

 generate a sparql query to get information from dbpedia "Where was Albert Einstein born?"

 SPARQL (SPARQL Protocol and RDF Query Language) is a semantic query language for databases, used to retrieve and manipulate data stored in Resource Description Framework (RDF) format. Here's how you might structure a SPARQL query to obtain the birthplace of Albert Einstein from DBpedia:

```
sparql Copy code

PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>

SELECT ?birthplace WHERE {
  res:Albert_Einstein dbo:birthPlace ?birthplace .
}
```

This query will return all resources that are associated with Albert Einstein's birthplace. It's worth noting that DBpedia's results may include more than one location, due to the nature of how information is linked in the database.

GPT-4 ⓘ



 generate the sparql query to get information from dbpedia without explanation for "Where was Albert Einstein born?" Only need sparql query

< 2 / 2 >



```
sparql Copy code

SELECT ?birthPlace WHERE {
  dbr:Albert_Einstein dbo:birthPlace ?birthPlace .
}
```



Is this conversation helpful so far?  

Considerations & Limitations

- **Hallucinations:** LLMs may produce incorrect or syntactically wrong SPARQL queries.
- **Explanation:** Outputs may contain explanations that break the SPARQL query structure or syntax.
- **Temperature Parameter:** The 'temperature' parameter influences the randomness of predictions, affecting the diversity of generated responses.
- **Scale and Cost:** LLMs such as GPT-3 and GPT-4 API involve costs and rate limits. Consider cost implications for scalability with high traffic or extensive usage.

Model Selection: GPT-3 vs. GPT-4

GPT-4

With broad general knowledge and domain expertise, GPT-4 can follow complex instructions in natural language and solve difficult problems with accuracy.

[Learn about GPT-4](#)

Model	Input	Output
8K context	\$0.03 / 1K tokens	\$0.06 / 1K tokens
32K context	\$0.06 / 1K tokens	\$0.12 / 1K tokens

Cost: 20 times higher.

Cost: 30 times higher.

GPT-3.5 Turbo

GPT-3.5 Turbo is optimized for dialogue.

[Learn about GPT-3.5 Turbo ↗](#)

Model	Input	Output
4K context	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
16K context	\$0.003 / 1K tokens	\$0.004 / 1K tokens

Source: <https://openai.com/pricing#language-models>

Hands-On Exercise

Overview

- OpenAI API: <https://openai.com/blog/openai-api>
- LangChain: <https://www.langchain.com/>
- Hugging face: <https://huggingface.co/>
- Gradio: <https://www.gradio.app/>
- DBpedia Chatbot

OpenAI API

- A cloud-based service for accessing OpenAI's GPT (Generative Pre-trained Transformer) models
 - GPT-4 and GPT-3.5: Models can understand as well as generate natural language or code
 - DALL·E: A model that can generate and edit images given a natural language prompt
 - Whisper: A model that can convert audio into text
 - Embeddings: A set of models that can convert text into a numerical form
 - Moderation: A fine-tuned model that can detect whether text may be sensitive or unsafe

Welcome to the OpenAI platform

Start with the basics

Quickstart tutorial

Learn by building a quick sample app

Examples

Explore some example tasks

Build an application



GPT

Learn how to generate text and call functions



GPT best practices

Learn best practices for building with GPT models



Embeddings

Learn how to search, classify, and compare text



Speech to text

Learn how to turn audio into text



Image generation

Learn how to generate or edit images



Fine-tuning

Learn how to train a model for your use case

Build a ChatGPT plugin



Introduction Beta

Learn the basics of building a ChatGPT plugin



Examples Beta

Explore ChatGPT plugin examples

Examples

Explore what's possible with some example applications



Grammar correction

Convert ungrammatical statements into standard English.



Summarize for a 2nd grader

Simplify text to a level appropriate for a second-grade student.



Parse unstructured data

Create tables from unstructured text.



Emoji Translation

Translate regular text into emoji text.



Calculate time complexity

Find the time complexity of a function.



Explain code

Explain a complicated piece of code.



Keywords

Extract keywords from a block of text.



Product name generator

Generate product names from a description and seed words.



Python bug fixer

Find and fix bugs in source code.



Spreadsheet creator

Create spreadsheets of various kinds of data.



Tweet classifier

Detect sentiment in a tweet.



Airport code extractor

Extract airport codes from text.



Mood to color

Turn a text description into a color.



VR fitness idea generator

Generate ideas for fitness promoting virtual reality games.



Marv the sarcastic chat bot

Marv is a factual chatbot that is also sarcastic.



Turn by turn directions

Convert natural language to turn-by-turn directions.



Interview questions

Create interview questions.



Function from specification

Create a Python function from a specification.



Improve code efficiency

Provide ideas for efficiency improvements to Python code.



Single page website creator

Create a single page website.

LangChain

- Framework for developing language model-powered applications
- Enables context-aware and reasoning applications.
- Modular components and off-the-shelf chains for ease of use and customization.
- Standard, extendable modules: Model I/O, Retrieval, Chains, Agents, Memory, Callbacks.
- Building applications with LLMs through composability.

Hugging Face

 **Hugging Face**

[Models](#) [Datasets](#) [Spaces](#) [Docs](#) [Solutions](#) [Pricing](#) [⌵](#)

Tasks [Libraries](#) [Datasets](#) [Languages](#) [Licenses](#) [Other](#)

Multimodal

[Feature Extraction](#) [Text-to-Image](#) [Image-to-Text](#)

[Text-to-Video](#) [Visual Question Answering](#)

[Document Question Answering](#) [Graph Machine Learning](#)

Computer Vision

[Depth Estimation](#) [Image Classification](#)

[Object Detection](#) [Image Segmentation](#)

[Image-to-Image](#) [Unconditional Image Generation](#)

[Video Classification](#) [Zero-Shot Image Classification](#)

Natural Language Processing

[Text Classification](#) [Token Classification](#)

[Table Question Answering](#) [Question Answering](#)

[Zero-Shot Classification](#) [Translation](#)

[Summarization](#) [Conversational](#)

[Text Generation](#) [Text2Text Generation](#) [Fill-Mask](#)

[Sentence Similarity](#)

Audio

Models 395,187 [new](#) [Full-text search](#) [Sort: 1](#)

 **openai/whisper-large-v3**
Automatic Speech Recognition • Updated 5 days ago • ⬇ 28.4k • ♡ 378

 **01-ai/Yi-34B**
Text Generation • Updated about 17 hours ago • ⬇ 24.3k • ♡ 820

 **latent-consistency/lcm-lora-sdx1**
Text-to-Image • Updated 5 days ago • ⬇ 15.4k • ♡ 164

 **openchat/openchat_3.5**
Text Generation • Updated 6 days ago • ⬇ 11.7k • ♡ 504

 **NousResearch/Yarn-Mistral-7b-128k**
Text Generation • Updated 12 days ago • ⬇ 28.5k • ♡ 414

 **meta-llama/Llama-2-7b-chat-hf**
Text Generation • Updated 1 day ago • ⬇ 876k • ♡ 1.76k

 **HuggingFaceH4/zephyr-7b-beta**
Text Generation • Updated 10 days ago • ⬇ 101k • ♡ 775

 **coqui/XTTS-v2**
Text-to-Speech • Updated 1 day ago • ⬇ 27.3k • ♡ 93

 **mistralai/Mistral-7B-v0.1**
Text Generation • Updated Oct 12 • ⬇ 427k • ♡ 1.79k

 **segmind/SSD-1B**
Text-to-Image • Updated 5 days ago • ⬇ 305k • ♡ 529

 **stabilityai/stable-diffusion-xl-base-1.0**
Text-to-Image • Updated 15 days ago • ⬇ 9.69M • ♡ 3.49k

 **latent-consistency/lcm-lora-sdv1-5**
Text-to-Image • Updated 5 days ago • ⬇ 28.7k • ♡ 70

 **distil-whisper/distil-large-v2**
Automatic Speech Recognition • Updated about 9 hours ago • ⬇ 36.8k • ♡ 247

 **01-ai/Yi-34B-200K**
Text Generation • Updated about 17 hours ago • ⬇ 1.09k • ♡ 134

 **01-ai/Yi-6B**
Text Generation • Updated about 17 hours ago • ⬇ 12.6k • ♡ 269

 **mistralai/Mistral-7B-Instruct-v0.1**
Text Generation • Updated Oct 11 • ⬇ 272k • ♡ 1.01k

DBpedia Chatbot

 Chatbot

Where is Daegu

Daegu is located in South Korea. It is the fourth largest city in the country after Seoul, Busan, and Incheon. It is situated in the southeastern part of the Korean Peninsula.

대구에서 유명한 것 하나를 나열하시오.

대구에서 유명한 것 중 하나는 '캠프 헨리'입니다.

Type a message...

Submit

 Retry

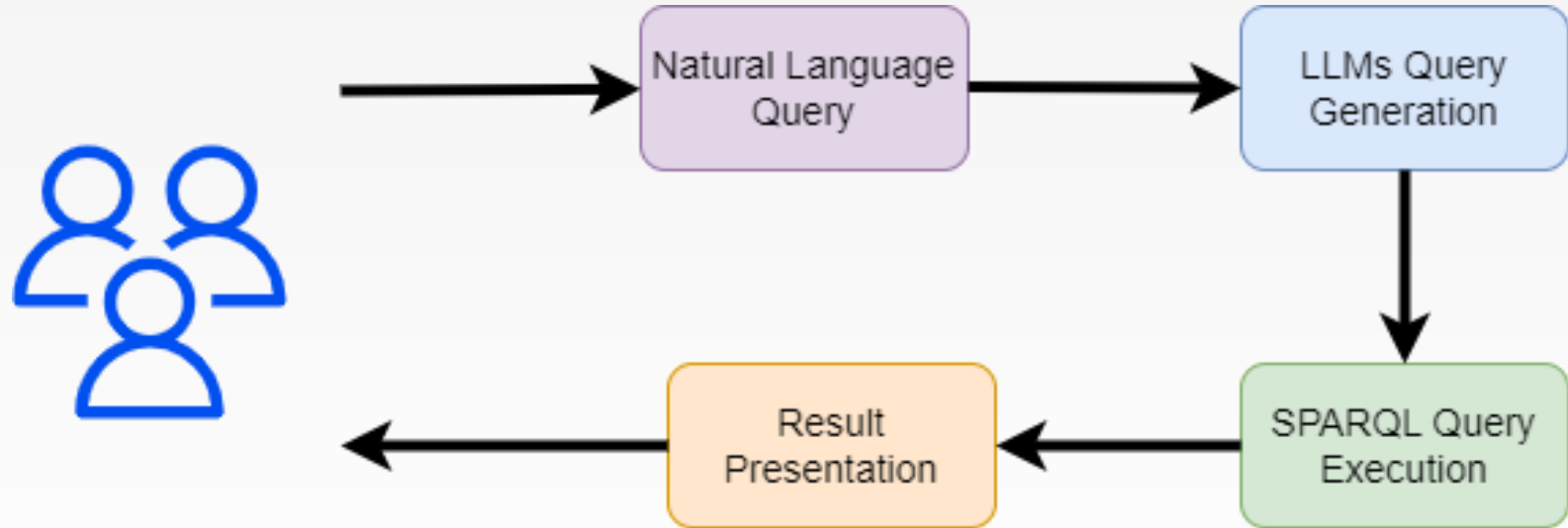
 Undo

 Clear

DBpedia Chatbot Features

- Answer Ontology-based queries: Retrieve precise information based on DBpedia's structured data about a wide range of topics.
- Engage in Intelligent Dialogue: Interact using natural language processing to maintain context and store conversations for future reference.
- Provide sources and justifications: Offer detailed explanations and citations from DBpedia for comprehensive understanding.
- Customize knowledge graphs: Develop personalized databases with the integration of proprietary data alongside DBpedia's datasets.

Program Workflow



Program Internal Workflow

- Collect the user's question
- Select a model
- Setup a prompt
- Retrieve the SPARQL query from LLMs
- Execute the SPARQL query at the DBpedia endpoint
- Response in plain language

ChatBot Backend

```
SELECT ?place WHERE {
  dbr:Daegu dbo:location ?place .
}
{'head': {'link': [], 'vars': ['place']}, 'results': {'distinct': False, 'ordered': True, 'bindings': []}}
```

The information for your query 'Where is Daegu' is as follows: .
Daegu is located in South Korea. It is the fourth largest city in the country after Seoul, Busan, and Incheon. It is situated in the southeastern part of the Korean Peninsula.

[HumanMessage(content='韓國首都?'), AIMessage(content='韓國的首都是首爾。'), HumanMessage(content='Where is Daegu'), AIMessage(content='Daegu is located in South Korea. It is the fourth largest city in the country after Seoul, Busan, and Incheon. It is situated in the southeastern part of the Korean Peninsula. '), HumanMessage(content='대구에서 유명한 것 하나를 나열하시오.')]]

```
SELECT ?subject ?label WHERE {
  ?subject dbo:location dbr:Daegu .
  ?subject rdfs:label ?label .
  FILTER (lang(?label) = 'ko')
} LIMIT 1
{'head': {'link': [], 'vars': ['subject', 'label']}, 'results': {'distinct': False, 'ordered': True, 'bindings': [{'subject': {'type': 'uri', 'value': 'http://dbpedia.org/resource/Camp_Henry'}, 'label': {'type': 'literal', 'xml:lang': 'ko', 'value': '캠프 헨리'}}]}}
```

The information for your query '대구에서 유명한 것 하나를 나열하시오.' is as follows: subject: http://dbpedia.org/resource/Camp_Henry; label: 캠프 헨리.
대구에서 유명한 것 중 하나는 '캠프 헨리'입니다.

Tips and Strategies

- Log user inquiries and corresponding responses for analysis.
- Solicit user feedback for continuous service enhancement.
- Prioritize the GPT-3 model for initial inquiries; escalate to GPT-4 as required.
- Optimize cost by efficiently managing token usage in LLM interactions, ensuring minimal and necessary input/output lengths.
- Utilize LLMs selectively for queries that require their sophisticated capabilities, ensuring alignment with appropriate use cases.

Discussion

Discussion

- The potential applications of metadata and the semantic web when utilizing Large Language Models (LLMs) in Libraries, Archives, and Museums (LAMs)
- Presentation and join discussion

Q & A

Thank You!

ylchen@vt.edu

<https://www.yinlinchen.com/>