

Hiring Trained Animals:

Generative AI Patterns and Practices for Systems Engineering

Barclay R. Brown, Ph.D., ESEP
Assoc. Dir. AI Research
Collins Aerospace
INCOSE CIO
Barclay.brown@incose.net

Image created using DALL-E2 with prompt, “an explorer in the style of starlord looking for artificial intelligence in a beautiful library laboratory with advanced machines and technology, digital art”

18-oct.-23



Application Design with LLMs



- Simple Q&A
 - Prompt / response
 - Prompt engineering
 - Persona
 - Templates
 - Detailed instructions
 - Retrieval-Augmented Generation
 - Vectorization of your data
 - Retrieval of relevant information
 - Prompting of LLM
 - Application design
 - Multiple prompting hidden from user
 - LLM as a component / function
 - Combining other elements



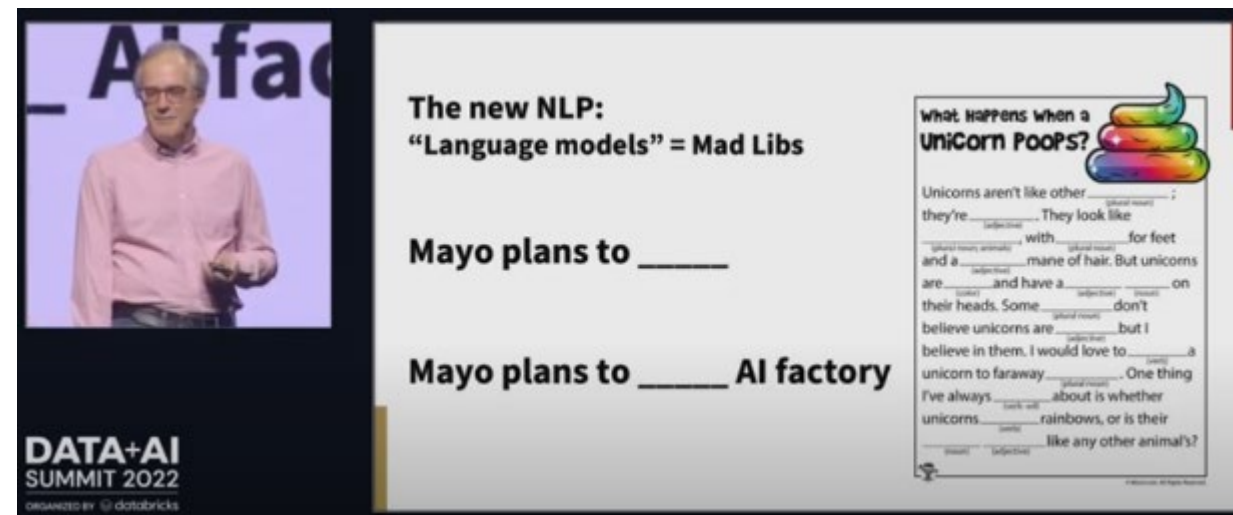
(generated by Microsoft Designer)



LLMs are based on Transformers, which add...

- Positional encoding
The dog is barking
1 2 3 4
- Attention vectors
The **dog** is **barking**
1 **2** 3 **4**
- What is it learning to do?
Predict the **NEXT WORD** given a sequence of words
- Train it on large text and reward it for correct predictions

Original paper: Attention is All you need
(<https://arxiv.org/pdf/1706.03762.pdf>)



Chris Manning (Stanford): Large Language Models learn to play the MadLibs Game

Excellent video:

<https://www.youtube.com/watch?v=YfXc4OBDmnM&t=12s>

Attention Model

(from original paper)



- Develop neural network structure
- Fill it with random numbers
- Train it by pretending to predict words from existing text
- Use error to refine weights and biases
- Train it A LOT
- Now use the model to predict words

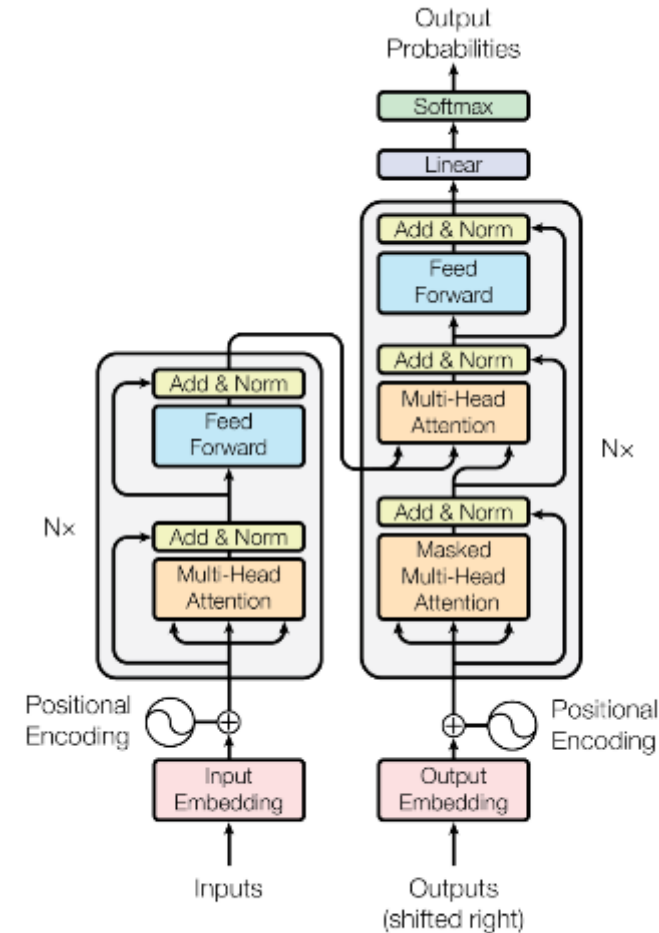


Figure 1: The Transformer - model architecture.

From: Attention is All you need (<https://arxiv.org/pdf/1706.03762.pdf>)

Open AI Models



search for your model and press ENTER...

columns to show

Average ☒ ARC ☒ HellaSwag ☒ MMLU ☒ TruthfulQA ☐ Type

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

how gated/private/deleted models

Model types

☒ pretrained ☐ fine-tuned ☐ instruction-tuned ☐ RL-tuned

Precision

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

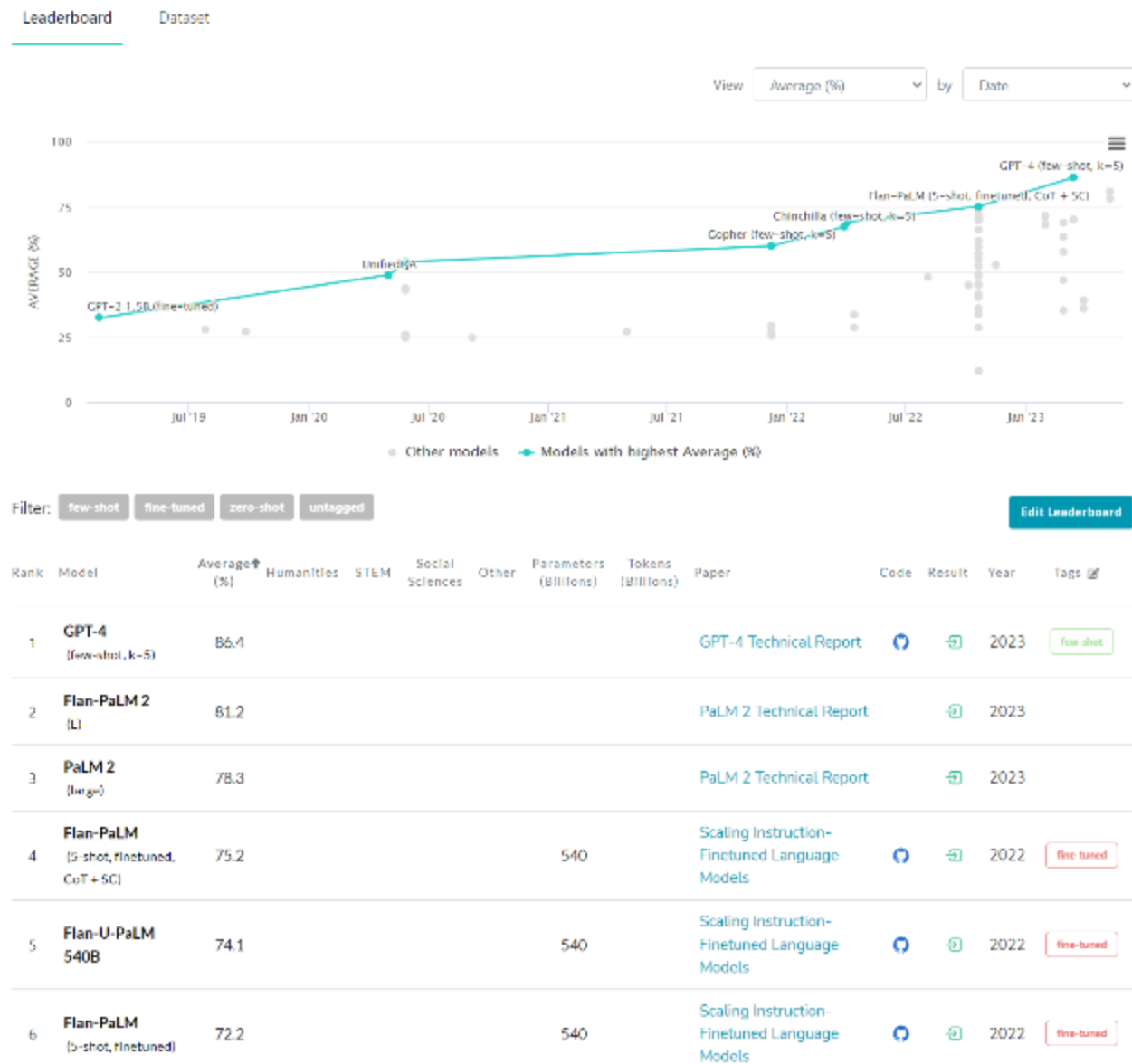
Model sizes

☒ Unknown ☒ < 1.5B ☒ ~3B ☒ ~7B ☒ ~13B ☒ ~35B ☒ 60B+

Model	Average <input checked="" type="checkbox"/>	ARC	HellaSwag	MMLU	TruthfulQA
tiiuae/falcon-180B	68.74	69.8	88.95	70.54	45.67
tiiuae/falcon-180B	68.57	69.45	88.86	70.5	45.47
tiiuae/falcon-180B	68.7	69.71	88.98	70.44	45.66
meta-llama/llama-2-70b-hf	67.35	67.32	87.33	69.83	44.92
tiiuae/falcon-180B	68.21	69.2	88.89	69.59	45.16
TigerResearch/tigerbot-70b-base	66.08	62.46	83.61	65.49	52.76
huggyllama/llama-65b	64.23	63.48	86.09	63.93	43.43
llama-65b	64.23	63.48	86.09	63.93	43.43
llama-30b	61.68	61.26	84.73	58.47	42.27
circulus/Llama-2-13b-orca-v1	62.91	62.03	82.27	57.71	49.61
tiiuae/falcon-40b	61.48	61.95	85.28	56.98	41.72
meta-llama/llama-2-13b-hf	58.66	59.39	82.13	55.77	37.38

COMMERCIAL MODELS

Multi-task Language Understanding on MMLU



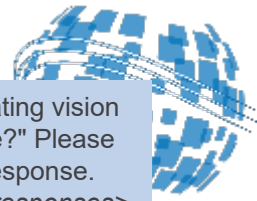
Key Concepts in LLM Applications



- LLM as component
- Prompt engineering
- Application Patterns
 - Multi-prompting
 - Retrieval-Augmented Generation
 - Deconstructing Data



LLM as a Component in a System or Application



The following numbered items are responses to the question, "In your own words (not restating vision or mission), what do you think of when you think about what the work of INCOSE should be?" Please summarize these in a list, giving more emphasis to concepts mentioned in more than one response. Use no more than ten items in your list and no more than 200 words in your response. <survey responses>

- Not necessary to turn over the entire application to the LLM
- Consider small tasks that an LLM can perform as part of larger application
 - Summarizing input text from a database or survey
 - Composing a reply, notification, or other message to users
 - Creating an introduction to a diverse set of information

(Survey Question: In your own words (not restating vision or mission), what do you think of when you think about what the work of INCOSE should be?)

1. Advancing Systems Engineering: By enhancing the discipline, its application, and expanding its body of knowledge. This involves moving from a process-focus to an outcome-focus, and making it more effective in solving real-world problems (1, 3, 5, 6, 11, 13, 21, 23).
2. Global Influence: Promoting Systems Engineering globally, addressing 21st Century's challenges, and facilitating its advance worldwide (2, 6, 12, 17, 22).
3. Tackling Societal Issues: Using Systems Engineering to address societal problems and promote sustainable use of resources (7, 9, 15).
4. Supporting Systems Engineers: Through community building, professional development, and practical support, providing clear guidance to engineer systems (10, 16, 20, 31).
5. Outreach: Expanding the awareness and promoting the value of Systems Engineering across all industries and domains (4, 8, 14).
6. Collaboration: Encouraging collaboration among Systems Engineers worldwide, to improve practices and develop systemic solutions (28, 29, 33).
7. Systems Thinking: Disseminating the importance of, and delivering methods for, practical Systems Thinking (18, 22, 32).
8. Knowledge Sharing: Building upon global system experiences, sharing relevant knowledge and good practices in the field (2, 7, 14).
9. Future Focus: Envisioning and leading the future of engineering for systems, while being creative and future-focused (10, 12, 30).
10. Practical Guidance: Providing practical guidance in systems design, impact assessment and to address complex challenges (26, 27).

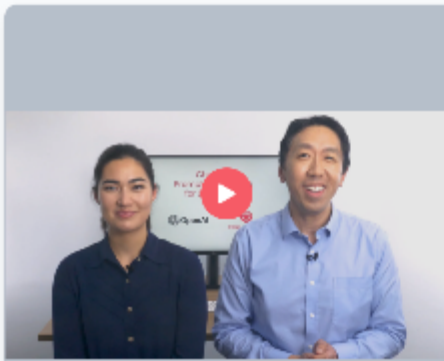
Let the LLM do what it does best




Prompt Engineering

- Give clear and specific instructions
 - Use delimiters to indicate distinct parts of input
 - Ask for structured output
 - Ask the model to check whether conditions are satisfied
- Give the model “time to think”
 - Proceed step by step
 - Give procedural instructions
- Refine prompts iteratively
 - Vary adjectives
 - Refine the persona







IN COLLABORATION WITH




ChatGPT Prompt Engineering for Developers

Go beyond the chat box. Use API access to leverage LLMs into your own applications, and learn to build a custom chatbot.

- Learn prompt engineering best practices for application development
- Discover new ways to use LLMs, including how to build your own chatbot
- Gain hands-on practice writing and iterating on prompts using the OpenAI API

 Beginner to Advanced  Isa Fulford, Andrew Ng

 Prerequisite recommendation: Basic Python

[Enroll For Free](#) [Learn more](#)

Example: Preparing for a Panel



- Prompt: you are an expert on sustainability in systems, and on systems thinking and systems engineering, and also an experienced and engaging talk show host. Please generate preparation materials for a talk show on the concept of sustainability in systems, with panelists from major manufacturing companies and sustainability organizations. First, list ten major themes in sustainability in systems, with references to articles and books. Next, generate ten thoughtful and engaging questions to ask panelists about sustainability and systems that will generate interesting conversation.

Response: Ten Major Themes in Sustainability in Systems

Circular Economy

Reference: "Cradle to Cradle: Remaking the Way We Make Things" by William McDonough & Michael Braungart

Concept: The idea of designing systems that eliminate waste and promote the reusability of products.

Energy Efficiency

Reference: "The Fifth Fuel: Strategies for Sustainable Energy" by Alliance to Save Energy

Concept: Techniques and strategies for reducing energy consumption and improving energy utilization in systems.

Life Cycle Analysis

Reference: "Life Cycle Assessment: Principles and Practice" by Mary Ann Curran

Concept: The comprehensive examination of a product's environmental and social impacts throughout its entire life cycle.

Supply Chain Sustainability

Reference: "Sustainable Logistics and Supply Chain Management" by David B. Grant, Alexander Trautrim, and Chee Yew Wong

Concept: Addressing sustainability in every link of the supply chain, from procurement to distribution.

Corporate Social Responsibility (CSR)

Ten Thoughtful and Engaging Questions for Panelists

Circular Economy: How are your organizations transitioning towards a circular economy, and what challenges have you encountered? Energy Efficiency: Can you share some specific examples of energy-efficient systems you've implemented and their impact?

Life Cycle Analysis: How do you incorporate life cycle analysis into your product design and decision-making processes?

Supply Chain Sustainability: What sustainable practices are you incorporating into your supply chain to minimize negative environmental and social impacts?

Corporate Social Responsibility: Beyond compliance, how does CSR shape your business decisions and systems design?

...

Example: formatted output



- Prompt: Consider the text below. Identify all common parts of speech in the text and create a table in json format with keys: word, part of speech, number of occurrences of the word. text: "Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal. "Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battlefield of that war. We have come to dedicate a portion of that field as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this. "But in a larger sense we cannot dedicate, we cannot consecrate, we cannot hallow this ground. The brave men, living and dead, who struggled here have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task



Sure, I can help you with that. Here's the JSON table with the identified words, their parts of speech, and the number of occurrences in the given text:

json

Copy code

```
[
  { "word": "Four", "part_of_speech": "CD", "occurrences": 1 },
  { "word": "score", "part_of_speech": "NN", "occurrences": 1 },
  { "word": "and", "part_of_speech": "CC", "occurrences": 1 },
  { "word": "seven", "part_of_speech": "CD", "occurrences": 1 },
  { "word": "years", "part_of_speech": "NNS", "occurrences": 1 },
  { "word": "ago", "part_of_speech": "RB", "occurrences": 1 },
  { "word": "our", "part_of_speech": "PRPS", "occurrences": 2 },
  { "word": "fathers", "part_of_speech": "NNS", "occurrences": 1 },
  { "word": "brought", "part_of_speech": "VBD", "occurrences": 1 },
  { "word": "forth", "part_of_speech": "RB", "occurrences": 1 },
  { "word": "on", "part_of_speech": "IN", "occurrences": 1 },
  { "word": "this", "part_of_speech": "DT", "occurrences": 3 },
```

Asking the model to check its own results



The best move for 'O' is to take position 5, the center of the board. This is the optimal move as it blocks 'X' from having a chance to win in the next move through the diagonal (positions 3, 5, 7). The updated board would look like:

markdown

Copy code

```
X | O | X
-----
- | O | -
-----
X | - | O
```

The numbers corresponding to the board positions would look like:

mathematica

Copy code

```
1-X, 2-O, 3-X, 4-empty, 5-O, 6-empty, 7-X, 8-empty, 9-O.
```

Regenerate

BA i'll take position 4 for X

Alright, if 'X' is placed in position 4, the board now looks like this:

Copy code

```
markdown
X | O | X
-----
X | O | -
-----
X | - | O
```

The numbers corresponding to the board positions would look like:

mathematica

Copy code

```
1-X, 2-O, 3-X, 4-X, 5-O, 6-empty, 7-X, 8-empty, 9-O.
```

With this move, 'X' has created two possible winning lines (vertical and diagonal), and 'O' must block one of them in the next move.

BA Is the game over

My apologies for the oversight. Yes, you're correct. The game is indeed over. With the placement of 'X' in position 4, 'X' has three in a row vertically on the leftmost column (positions 1, 4, 7). This is a winning condition for Tic-Tac-Toe.

Here is the final board for clarity:

Regenerate

Learn how to “talk” to LLMs through practice

Giving the Model “Time to Think”



- Explain as you would to an intelligent but inexperienced human
- Step by step instructions are very helpful
- Use delimiters to differentiate instructions from source material

```
text = f"""
In a charming village, siblings Jack and Jill set out
a quest to fetch water from a hilltop \
well. As they climbed, singing joyfully, misfortune \
struck—Jack tripped on a stone and tumbled \
down the hill, with Jill following suit. \
Though slightly battered, the pair returned home to \
comforting embraces. Despite the mishap, \
their adventurous spirits remained undimmed, and they
continued exploring with delight.
"""

# example 1
prompt_1 = f"""
Perform the following actions:
1 - Summarize the following text delimited by triple \
backticks with 1 sentence.
2 - Translate the summary into French.
3 - List each name in the French summary.
4 - Output a json object that contains the following \
keys: french_summary, num_names.

Separate your answers with line breaks.

Text:
```{text}```
"""

response = get_completion(prompt_1)
print("Completion for prompt 1:")
print(response)
```

# Using a Prompt and Response Templates



**Free-form emails come in requesting registration in a course. Use LLM to extract the important information**

**From:** Trent Milam  
**Sent:** Wednesday, September 6, 2023 11:05 AM  
**To:** Barclay Brown [Barclay.Brown@incose.net](mailto:Barclay.Brown@incose.net)  
**Subject:** SEQM QMI Training

Hello,  
My name is Trent Milam, member #404238, I would like to sign up for the SEQM QMI training course cohort. I'm a new member of the working group and INCOSE, excited to learn more!

Thanks,  
TM



```
{
 'reg': 'yes',
 'Fn': 'Trent',
 'Ln': 'Milam',
 'Member Number': '404238',
 'Email': 'trent.milam@gmail.com',
 'emdate': '09/06/2023'
}
```

```
messages = [{"role": "system", "content": "You are an AI assistant that helps people find information."}, \
{"role": "user", "content": "consider the email delimited by triple backticks. Extract the following \
items into a python dictionary in the following format. Show only the python dictionary--do not \
include any other messages or text: reg (yes or now): Is the email requesting to register in a course?; \
Fn: What is the registrant's first name?; Ln: What is the registrant's last name?; \
Member Number: What is the registrant's member number?; \
Email: What is the registrant's email address?; emdate: What date was the email sent? format mm/dd/yyyy; \
``` + emailtext + ```"}],
```

Ask for what you want—you might just get it!

Retrieval Augmented Generation: *An LLM Application Pattern*

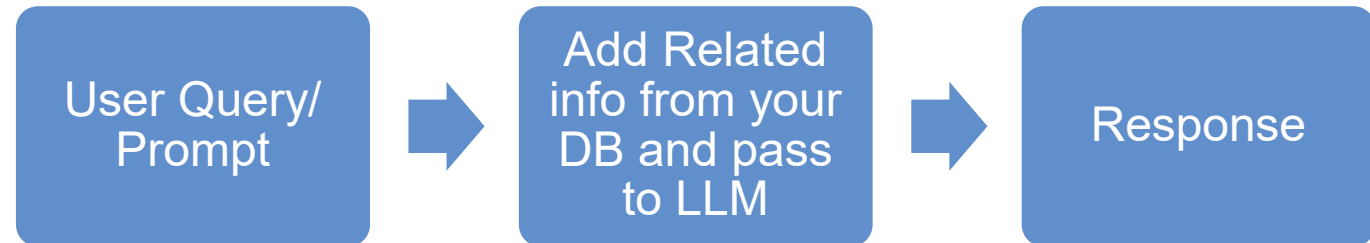


- Find source material
- Create index and store
- Take user prompt, and search for related information in index
- Pass search results along with prompt to LLM
- The indexer calls an type of LLM to build the index, **so KNOW where your data is going!**
- Indexers may use commercial embedding models like
- Llama-index can also work with open source and downloaded LLMs

LLM Chat



Retrieval-Augmented Generation



Building your DB



Building a DB of information for RAG



- Collect documents in common formats
 - Pdf
 - Text
 - Word
- Use loaders, like those from the LangChain library, to load documents into memory structures
- Split documents into chunks to allow granular retrieval (also from LangChain)
 - Split by separator
 - Recursive splitter
- Create embedding vectors for each chunk and store in a vector database (e.g. ChromaDB)
- Persist Vector DB on disk for future runs
- Rebuild index when source docs are added or updated

Collect Documents

Use Loaders

Split Docs

Split By Separator

Recursive Splitter

Create Embedding Vectors

Persist Vector DB

Rebuild Index

(Illustration created by GPT-4 in using svg output format)



Destructuring Data

- Traditional thinking: structured data is better than unstructured for computer applications
- LLM thinking: LLMs understand natural language, but structured data may not carry the meaning in the structure in an obvious way
- Metaphor: Humans often need other humans to explain structured data, tables, plots, diagrams, etc. in natural language so they can understand
- Memo from LLM: maybe explain the data to me too!

X-ref: F30 Platform code includes M3 4dr Sedan (3.0L 6cyl Turbo 7A), M3 4dr Sedan (3.0L 6cyl Turbo 6M), M3 4dr Sedan (3.0L 6cyl Turbo 6M), 340i xDrive 4dr Sedan AWD (3.0L 6cyl Turbo 8A), ActiveHybrid 3 4dr Sedan (3.0L 6cyl Turbo gas/electric hybrid 8A), 340i 4dr Sedan (3.0L 6cyl Turbo 8A), 340i xDrive 4dr Sedan AWD (3.0L 6cyl Turbo 8A), 328d xDrive 4dr Wagon AWD (2.0L 4cyl Turbodiesel 8A), 340i 4dr Sedan (3.0L 6cyl Turbo 8A)...

B	C	D	E	DR	DS
Make	Model	Year	Trim (description)	Car classification	Platform code / generation number
BMW	3 Series	2018	M3 4dr Sedan (3.0L 6cyl Turbo 7A)	Compact car	F30, F31, F80
Ford	F-150	2021	Limited 4dr SuperCrew 4WD 5.5 ft. SB (3.5L 6cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2020	Limited 4dr SuperCrew 4WD 5.5 ft. SB (3.5L 6cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	Limited 4dr SuperCrew 5.5 ft. SB (3.5L 6cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2012	M3 2dr Convertible (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
Ford	F-150	2020	Limited 4dr SuperCrew 5.5 ft. SB (3.5L 6cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2010	M3 2dr Convertible (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
BMW	3 Series	2018	M3 4dr Sedan (3.0L 6cyl Turbo 6M)	Compact car	F30, F31, F80
BMW	3 Series	2008	M3 2dr Convertible (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
BMW	3 Series	2016	M3 4dr Sedan (3.0L 6cyl Turbo 6M)	Compact car	F30, F31, F80
Ford	F-150	2021	Platinum 4dr SuperCrew 4WD 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	Platinum 4dr SuperCrew 4WD 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2012	335is 2dr Convertible (3.0L 6cyl Turbo 6M)	Compact car	E90, E91, E92, E93
BMW	3 Series	2012	M3 2dr Coupe (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
Ford	F-150	2021	King Ranch 4dr SuperCrew 4WD 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	King Ranch 4dr SuperCrew 4WD 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	Platinum 4dr SuperCrew 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2020	Platinum 4dr SuperCrew 4WD 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	Platinum 4dr SuperCrew 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2020	Platinum 4dr SuperCrew 4WD 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2010	M3 2dr Coupe (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
Ford	F-150	2020	King Ranch 4dr SuperCrew 4WD 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2021	M340i xDrive 4dr Sedan AWD (3.0L 6cyl Turbo 8A)	Compact car	G20
Ford	F-150	2021	King Ranch 4dr SuperCrew 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2006	M3 2dr Convertible (3.2L 6cyl 6M)	Compact car	E46
BMW	3 Series	2008	M3 2dr Coupe (4.0L 8cyl 6M)	Compact car	E90, E91, E92, E93
Ford	F-150	2020	Raptor 4dr SuperCrew 4WD (3.5L 6cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2020	King Ranch 4dr SuperCrew 4WD 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
Ford	F-150	2021	King Ranch 4dr SuperCrew 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2020	M340i xDrive 4dr Sedan AWD (3.0L 6cyl Turbo 8A)	Compact car	G20
Ford	F-150	2020	Platinum 4dr SuperCrew 6.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	
BMW	3 Series	2004	M3 2dr Convertible (3.2L 6cyl 6M)	Compact car	E46
Ford	F-150	2020	Platinum 4dr SuperCrew 5.5 ft. SB (5.0L 8cyl Turbo 350)	Full-size pickup tru (13th gen)	

The 2018 BMW 3 Series M3 4dr Sedan (3.0L 6cyl Turbo 7A), is a compact car using platforms F30, F31 or F60... and is classified as a Compact Car...

Tell the LLM what the data means



Summary

- LLMs can be a unique component in an application carrying specialized functions
- Patterns for employing LLMs continue to evolve
- Like any system, output from an LLM component must be verified to be trusted
- Systems approaches must be employed to result in trusted systems

