NAIRR Pilot

National Artificial Intelligence Research Resource Pilot

AI Using Large Language Models
Danny Havert, Indiana University
Track 2, Wednesday
AI Workshop Denver, CO April 2-3, 2025

Outline

Presentation (45 min)

- What are Large Language Models?
- What are LLMs good for?
- LLM Applications in Industry
- LLM Applications in Academia
- Running your own LLM

Hands-on Activities (45 min)

- LLM Interactions through Open WebUI
- LLM Interactions through API

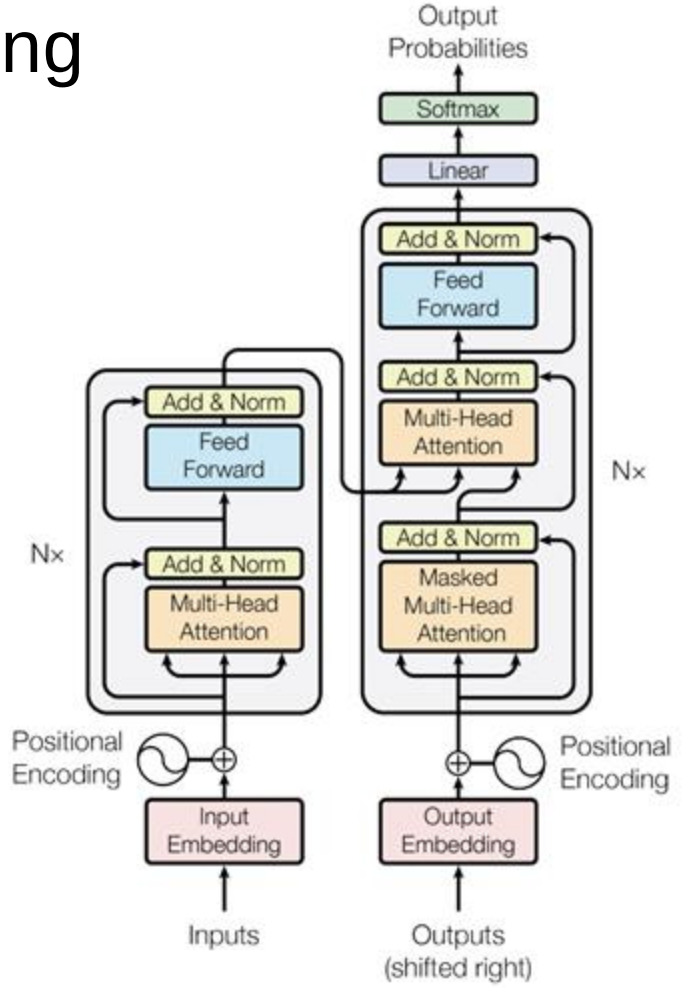
What are Large Language Models?

- **What are Large Language Models?**
- What are LLMs good for?
- LLM Applications in Industry
- LLM Applications in Academia
- Running your own LLM

A Transformative Approach to Language Processing

- Large Language Models derive from the field of Natural Language Processing (NLP), which analyzes language with a variety of approaches
- Most Large Language Models (LLMs) are built upon a type of neural network called a **transformer model**
  - First developed by Google in the landmark paper "Attention Is All You Need" (Vaswani 2017)
- Transformer models augment traditional NLP tokenization techniques and parallelizes the processing of these tokens

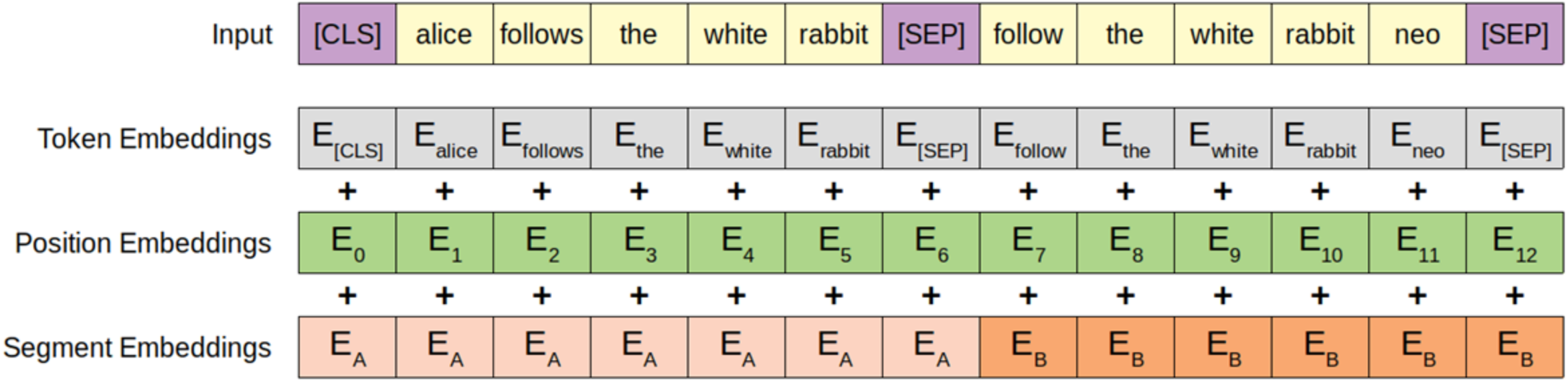
- For a great visual and conceptual explanation of how LLMs work, I highly recommend Grant Sanderson's (a.k.a. *3Blue1Brown)* video series "Neural networks"
  - https://www.3blue1brown.com/topics/neural-networks



Source: Vaswani (2017), Attention Is All You Need (https://doi.org/10.48550/arXiv.1706.03762)

The Transformer Model – Inputs & Embeddings

1. Tokenization - Text is converted into tokens-- fragments of words, characters, or punctuation.
2. Input Embeddings - tokens are converted into a numerical vector that can be mapped to semantic meaning
    - Also, positional encodings based on each token's position in the sentence.



The Transformer Model – Attention & Feeding Forward

3. Attention Layer (Self-Attention) - Modifies token vectors by considering other nearby tokens.
   - Context is everything. Meaning of words changes based on context.

I bought an apple. It was delicious.

I bought an apple. It runs well.

4. Feed Forward Neural Network - Purpose is to transform the output of the attention layer into a higher-dimensional space, allowing the model to capture more complex patterns and relationships in the input sequence.
5. Normalization & Output Embeddings
   - Normalization occurs after the attention and feed forward layers
   - Output embeddings maps output vector to vocabulary with a probability distribution
   - Finally, the next word in the sequence is picked from the probability distribution.

Putting the "Large" in LLM

- “Large” can refer to the size and complexity of both the model itself, and the datasets it has trained on
- Models are often described by their parameter size, the number of adjustable values within the model, which can reflect the size and complexity of the model
- Higher-parameter models are more hardware-intensive to run, demanding more performant GPUs to run inference in a reasonable time, enough memory to hold the models while running them, and occupy a relatively large storage footprint

The parameter size of the model is often appended to the name of the model itself, e.g. Meta Llama 3.1 405B is a 405 Billion parameter size model

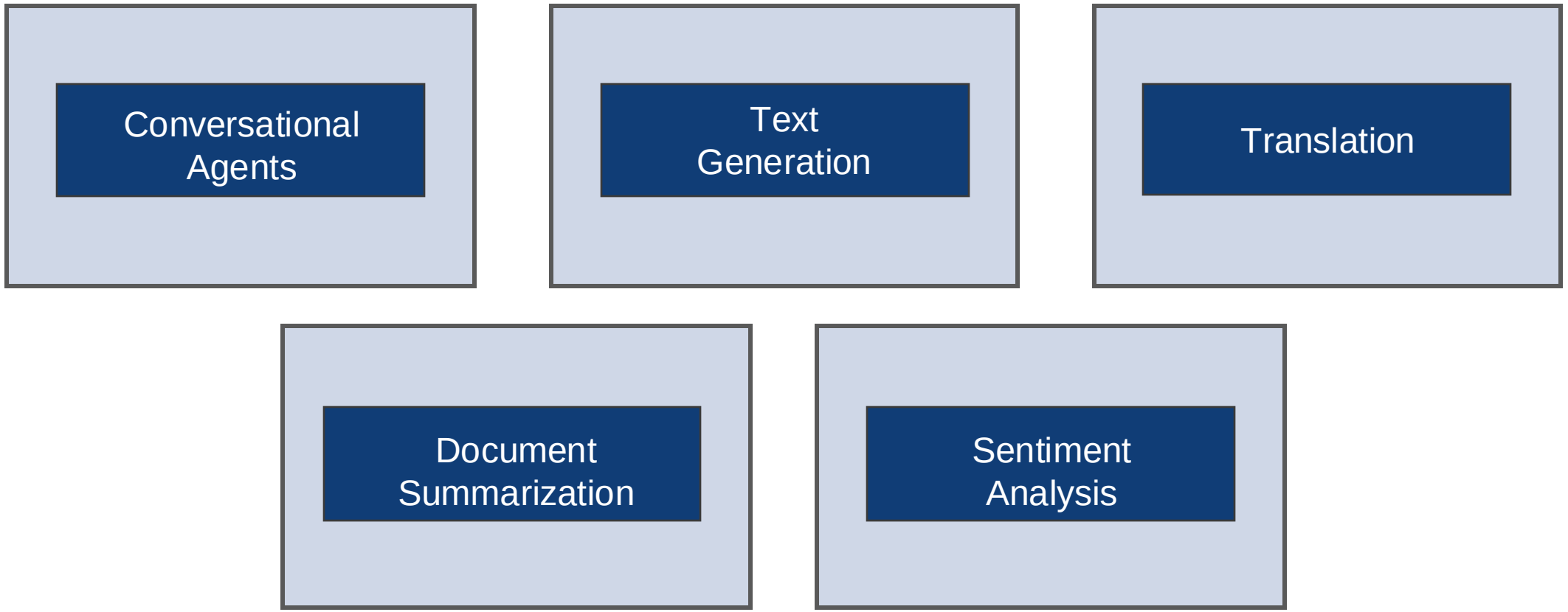
	Training Data	Params	Input modalities	Output modalities	Context length	GQA	Token count	Knowledge cutoff
Llama 3.1 (text only)	A new mix of publicly available online data.	8B	Multilingual Text	Multilingual Text and code	128k	Yes	15T+	December 2023
		70B	Multilingual Text	Multilingual Text and code	128k	Yes		
		405B	Multilingual Text	Multilingual Text and code	128k	Yes		

LLM Development Process

What are LLMs good for?

- What are Large Language Models?
- **What are LLMs good for?**
- LLM Applications in Industry
- LLM Applications in Academia
- Running your own LLM

LLMs – What are they good for?



Text Generation

- Text generation is *the* fundamental capability of LLMs
  - most popular LLMs will perform well at generalized text generation
- Code development & completion
  - Microsoft's GitHub Copilot can provide generated code and serve as a pair programmer
  - Meta's Code Llama is an open-weights alternative
- Document writing assistance
  - Grammarly, a product that has long been in the consumer NLP space, now provides an LLM writing assistant

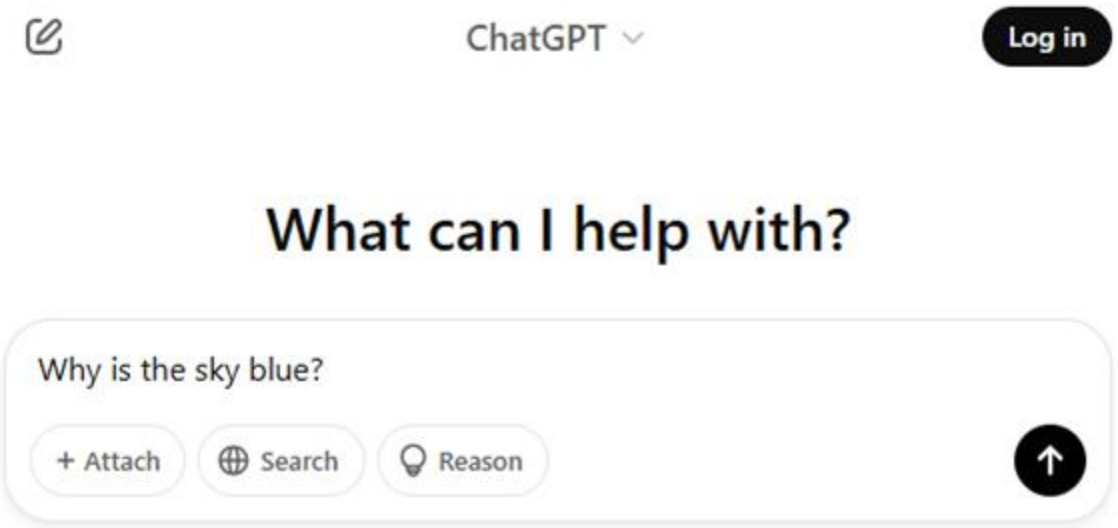
Write a function in JavaScript that calculates the the number of days between two dates. The function should account for errors such as the end date being before the start date.

Here's a JavaScript function that calculates the number of days between two dates, and it also accounts for errors such as the end date being before the start date:

javascript Copy

```javascript
function calculateDaysBetweenDates(startDate, endDate) {
  // Convert both dates to Date objects if they are not already
  startDate = new Date(startDate);
  endDate = new Date(endDate);

  // Check for invalid dates
  if (isNaN(startDate.getTime()) || isNaN(endDate.getTime())) {
    return "Invalid date format.";
  }

  // Check if the end date is before the start date
  if (endDate < startDate) {
    return "Error: End date cannot be before start date.";
  }

  // Calculate the difference in milliseconds
  const timeDifference = endDate - startDate;

  // Convert the difference from milliseconds to days
  const daysDifference = timeDifference / (1000 * 3600 * 24);

  return daysDifference;
}
```

Conversational Agents



- LLMs are trained on lots of conversational text and are therefore well suited to conversational tasks!
- Fine-tuned (re-trained to adapt to a task) chat models
  - Often used to provide a more consistent, reliable, and safer experience
  - Involve supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to provide safeguards and improve result accuracy

Translation

- Translation tasks have always been a pillar of LLM development
  - “Attention Is All You Need” focused on the creation of a model designed for machine translation tasks
- Many popular LLMs have broad enough training data to perform translation well enough
  - Google’s Cloud Translation API provides access to a fine-tuned translation LLM
  - Claude and GPT-4 both score well in multilingual translation
  - Aya and BLOOM are active open-source multilingual LLMs
- Good translation does not always imply good localization -- effective delivery of author intent can be missing in the target language output

Large Language Machines have become a lot more powerful in tackling translation tasks in recent years.

French:

Les modèles de langage de grande échelle sont devenus nettement plus puissants pour traiter les tâches de traduction ces dernières années.

German:

Die großen Sprachmodelle haben in den letzten Jahren deutlich an Leistungsfähigkeit für die Bearbeitung von Übersetzungsaufgaben gewonnen.

Chinese (Simplified):

大语言模型近年来在翻译任务处理能力方面取得了明显进步。

Chinese (Traditional):

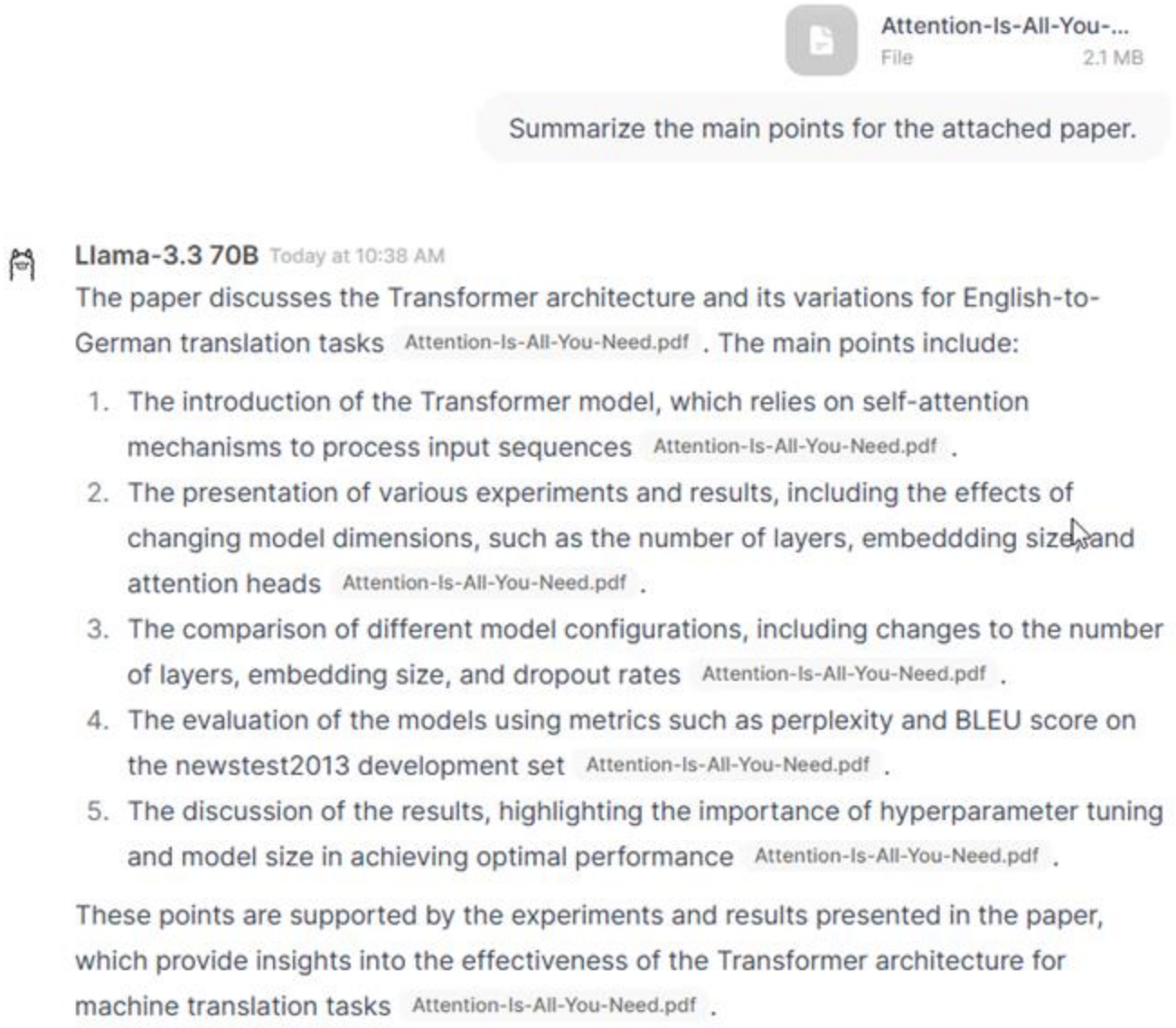
大語言模型近年來在翻譯任務處理能力方面取得了明顯進步。

Japanese:

大規模言語モデルは、近年、翻訳タスクの処理能力が大幅に向上した。

Document Summarization

- A few approaches to summarization:
  - Extractive summarization- summary via extracting key sentences
  - Abstractive summarization - generating new sentences to describe core ideas and meaning
  - ‘Multi-level summarization’ - a mix of extractive and/or abstractive
- Larger context window sizes (amount of tokens the model can process at once) are useful for document summarization tasks, especially when summarizing long texts or multiple texts
- Encoder-Decoder (seq2seq) models are tailored for text summarization, though most popular LLMs are capable enough and have larger context windows

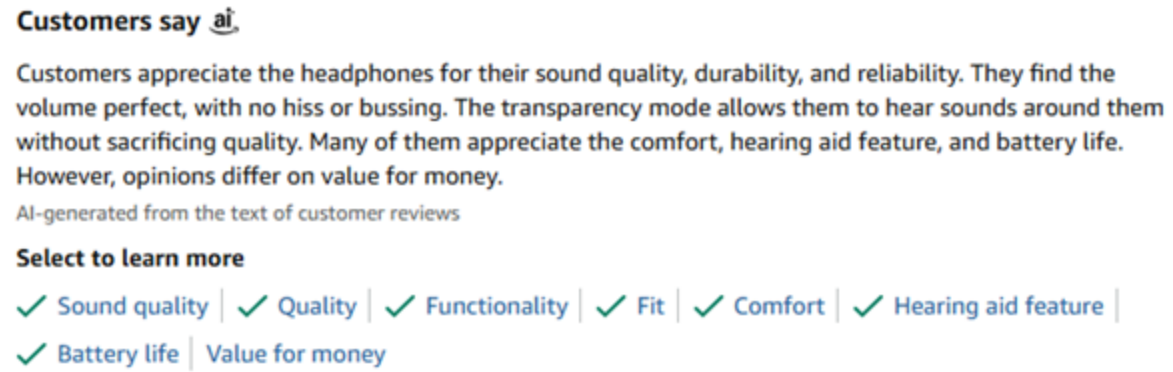


Sentiment Analysis

- Sentiment analysis is a classification problem, assigning text to a specific category (e.g. positive, negative, neutral) according to emotional tone
- Encoder-only transformer models like BERT and variants like RoBERTa are still considered to be good options for this case
  - MUCH more efficient
- The base models of popular LLM offerings can tackle sentiment analysis problems, but usually perform this task better after a bit of fine-tuning

Use Cases:

- Customer service and feedback



- Market analysis
- Social media monitoring

Using LLMs, and Customization

- Many models are publicly available as a service-- usually at cost
  - Available models can range from generalist in nature, not tuned to specific use-cases, to specific purpose
  - Generally used remotely via cloud-hosted services (model is not downloaded and used directly on a user's system)
- Motivations for creating custom LLM platforms can range from security concerns, cost-savings, model customization, ability to target specific applications and tasks, and more
- Training a model from scratch can be expensive- the Meta Llama 3.1 405B model consumed 30.84 million GPU hours to train.
- Several methods exist to help adapt pre-existing models to suit your specific use-case and run them

Common Options for Customizing Models

	Works by	Useful for
Fine-Tuning	Updating the model's weights with domain- specific or task-specific data	Adjusting the model to perform a task or adopt knowledge it underperforms in
Retrieval-Augmented Generation (RAG)	Providing domain-specific data as a searchable data store for the LLM to use in responses	Providing domain-specific or up-to-date knowledge
Prompt Engineering	Adjusting the prompts (input) sent to the model, rather than fully adjusting the model itself	Suggesting the base model to parse input and provide output in a specific way

LLM Applications in Industry

- What are Large Language Models?
- What are LLMs good for?
- **LLM Applications in Industry**
- LLM Applications in Academia
- Running your own LLM

BloombergGPT

- BloombergGPT is a 50-billion parameter financial domain-specific model capable of providing unique assessments of the massive datasets available on the Bloomberg Terminal
- Built from scratch, trained on 363 billion tokens of 40 years' of financial English language documents in addition to a public language dataset
- Limited size of dataset of interest provided unique challenges in producing a competitive model
- Developed to improve existing NLP workflows within the organization and provide offerings to customers

https://arxiv.org/pdf/2303.17564

Med-PaLM 2

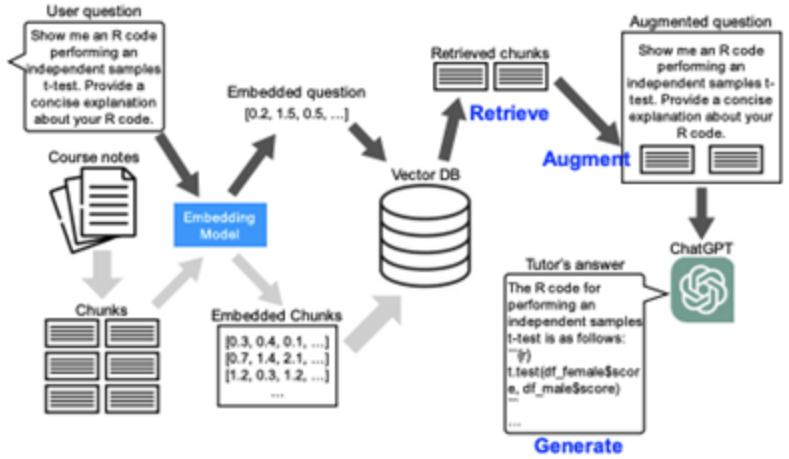
- Model fine-tunes the PaLM 2 model with medical domain-specific data and uses a novel prompting strategy, Ensemble Refinement (ER), where the LLM produces several answers to a prompt before generating a final answer
- Achieves an 85%+ accuracy on the MedQA dataset of US Medical Licensing Examination (USMLE)-style questions--a passing score that is considered ‘expert-level”
- Designed serve as a supplemental resource for medical professionals
- Can assist in the creation of patient history and after-visit summaries, long and short-form Q&A, and multiple-choice Q&A
- Available via Google Cloud Platform

LLM Applications in Academia

- What are Large Language Models?
- What are LLMs good for?
- LLM Applications in Industry
- **LLM Applications in Academia**
- Running your own LLM

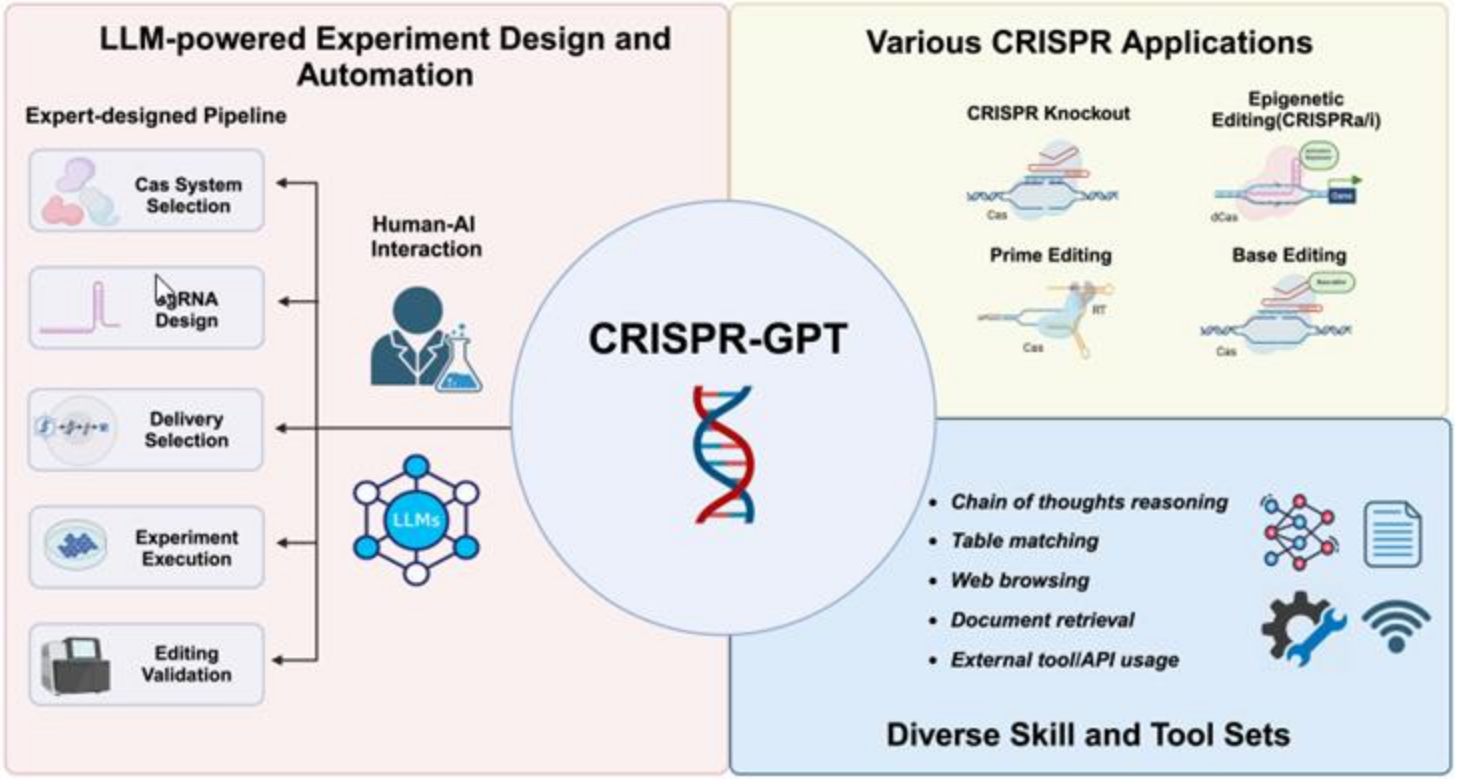
RAG-Based LLM Statistics Tutor

- LLMs provide excellent opportunities in the development of Intelligent Tutoring Systems (ITS)
- Dr. Lee at the University of North Texas Developed a ChatGPT-based tutor to assist statistics student with natural language-based guidance and code drafting assistance
- Retrieval-Augmented Generation (RAG) was implemented to:
  - Provide pre-defined, validated, up-to-date information
  - Help prevent the LLM from producing ‘hallucinations’, or inaccurate or fabricated misinformation



Source: Lee (2024), Developing a computer-based tutor… (https://doi.org/10.1007/s10639-024-13129-5)

CRISPR-GPT: An LLM Agent for Automated Design of Gene-Editing Experiments



Source: Huang et al. (2024), CRISPR-GPT: An LLM Agent... (https://doi.org/10.48550/arXiv.2404.18021)

Miscellaneous LLM Applications in Research

- Summarization, literature review
- Brainstorming (Hypotheses, experimental protocols, approaches to data analysis or visualization, names for your project)
- Knowledge-Base chatbot assistant
- Programming, debugging assistant
  - Use it in your IDE with Continue , Zed, etc
- Writing / proofreading assistant
- Language translation for technical writing and jargon

For a list of concrete applications, check out:

- LLM4SR: A Survey on Large Language Models for Scientific Research
  - https://github.com/du-nlp-lab/LLM4SR

Running your own LLM

- What are Large Language Models?
- What are LLMs good for?
- LLM Applications in Industry
- LLM Applications in Academia
- **Running your own LLM**

Why would you want to run your own LLM?

- Data privacy and security.
  - Essential for Protected Health Information (PHI)
- Customization and flexibility
  - Translation service fine-tuned for a specific language
  - Use RAG to build a model with extensive knowledge on scientific papers in your field
- Cost-effectiveness (long term)
- No rate-limiting
- Integration with custom systems

Options for Serving

- Software – What tools to use?
  - For personal consumption -
    - Llama.cpp - https://github.com/ggml-org/llama.cpp
    - Llamafile - https://github.com/Mozilla-Ocho/llamafile
    - Ollama - https://github.com/ollama/ollama
  - For serving to others
    - vLLM - https://github.com/vllm-project/vllm
    - SGLang - https://github.com/sgl-project/sglang

Options for Serving

- Hardware – Where to run your LLM?
  - LLMs *can* run on CPU – slow but less expensive
    - Has a use case in fire-and-forget text analysis
    - If you need quick response times, GPU is a *must*
  - VRAM is most notable limiting factor for GPU
  - ACCESS ecosystem
    - HPC – for training
      - Expanse, Delta, and many more
    - Cloud – for serving
      - Jetstream2
  - Commercial Cloud – if you can afford

How to Shop for LLMs

- Chatbot Arena for comparison rankings (Elo score) - https://lmarena.ai/
- Benchmarks with private / uncontaminated problem sets
  - LiveBench - https://livebench.ai/
  - Kagi LLM Benchmarking Project - https://help.kagi.com/kagi/ai/llm-benchmark.html
- Benchmarks with public problem sets
  - MMLU, etc. - for list, see https://github.com/leobeeson/llm_benchmarks
- HuggingFace for downloading models - https://huggingface.co/

- What will fit on the hardware you have?
  - Weights need to fit in VRAM!
  - Use quantization to reduce VRAM usage of model
    - **16-bit** floating point - Typical native format for training
    - **8-bit** - 2x smaller with negligible capability loss - Great tradeoff!
    - **4-bit**, **5-bit K-means clustering** - Even smaller, but start to notice capability loss
    - **2-3 bits** - Silly demos, running LLMs on your phone

Hands-on Exercises

LLM Interactions through OpenWebUI

- https://llm.jetstream-cloud.org
    - Requires an ACCESS account

LLM Interactions through API

- JupyterHub – *link and password to be shared live*