Sandia National Laboratories

Exceptional service in the national interest

My Very First Large(ish) Language Model

Doing Model Inference, Fine Tuning & RAG at Sandia

Chris Siefert and Graham Harper

Trilinos Users Group Meeting, 10/23/2024

SAND2024-14334PE

Outline

- Background on LLMs
- Doing LLMs at Sandia
- Demos!

If you want to follow along with our examples you can get them here:
https://gitlab-ex.sandia.gov/csiefer/sandia-llm-tools

And our training data:
https://gitlab-ex.sandia.gov/gbharpe/llm-training-data

Large Language Models (LLMs) in the Popular Press

- LLMs have been *all over* the news for a couple years (maximum media hype!)

- You might have heard of...
  - OpenAI's ChatGPT (last year's SNL town hall with 1000+ attendees)
  - Facebook's LLama, Llama2, Llama3
  - Google's Bard, Gemini, Gemma
  - Microsoft/Github's Copilot
  - Search engines (Google, Bing, and more)

- But what are they and how can you make one of your very own!

Reminder from the CIO: "Engagement with ChatGPT (including questions and responses) can become part of the AI model and could result in the disclosure of Sandia and government information."

Hackers want your ChatGPT conversations! https://www.tomshardware.com/news/over-100000-chatgpt-account-credentials-made-available-on-the-dark-web

What is a Large Language Model (LLM)?

- A Large Language Model is a token prediction engine.

Given a stream of tokens what is the next token?

- The model assigns percentages to the next token and dice are rolled.

Never gonna	give	(31%)
	let	(18%)
	run	(10%)
	make	(8%)
	....	

- That new token is then added to the stream and the next token is predicted. There is nothing mystical going on here!

- **Note:** Selecting/modifying tokens/words in a human-undetectable but machine-detectable fashion is possible, and is referred to as "watermarking" an LLM.

- We'll purposely defer the "What is a token?" question for the moment.

See: For example, DeepTextMark https://arxiv.org/abs/2305.05773

What Do You Need to Make a LLM?

- Training a LLM requires three (or four) pieces...

- **Training data** – What data do you train on?
  - Common Crawl: https://commoncrawl.org/
  - OpenWebText: https://huggingface.co/datasets/openwebtext
  - The Pile: https://pile.eleuther.ai
  - The Stack (Code): https://huggingface.co/datasets/bigcode/the-stack
  - *How much of the internet do you want to download and train on?*

- **Tokenizer** – The thing that turns your input data into tokens.

- **Model architecture** – How do you turn a stream of input tokens into probabilities for an output token?

- **Optional: Input preprocessing** – User prompt and/or content filtering to keep your LLM from turning into the cesspool that is the internet (common in commercial models), deduplication, etc.

What is a Token?

- In the simplest case, each letter, number and symbol can be a token.
  - breaks text into sequence ["n","e","v","e","r"," ","g","o","n","n","a"," ","g","i","v","e",...]
  - sequence is assigned ids [14,5,22,5,18,30,7,15,14,14,1,30,7,9,22,5,...]

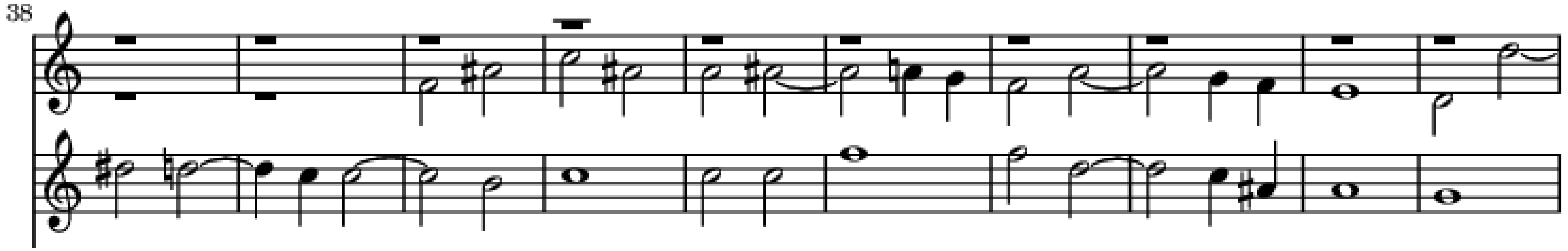
- GPT tokenizer (used in many LLMs) does something a little different.
  - All the single character tokens are still tokens.
  - But common English words and word parts are also tokens....
  - The first few non-letter GPT tokens are "the" "er" "at" "en" "it" "is" "an" "or" "es" "ed" "ing"...
  - Spaces are also included at the front of some tokens, so word breaks happen correctly.
  - GPT-2 tokenizer has 50k tokens, GPT-4 tokenizer has 200k tokens.

- **Note:** Tokenization affects model quality!
  - A model with character-by-character tokenization will generally perform worse than a model with more complex (e.g. GPT) tokenization.
  - Tokenization also depends on the structure of the input language
- But you can tokenize *any* sequence of data, even if it doesn't involve language...

See https://huggingface.co/gpt2/resolve/main/vocab.json

Example: Tokenizing MIDI Music

- Sheet music can be tokenized: Tokens represent note durations and pitches.
- Some special sauce is required (e.g. bar/position tokens & note quantization).
- Fun example: AI-generated music in the style of 15th-16th century polyphony.

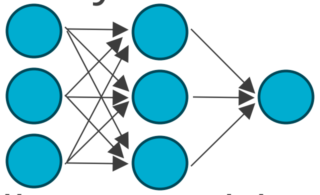


See: https://github.com/Natooz/MidiTok
Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions https://dl.acm.org/doi/10.1145/3394171.3413671
Training Data from: *The Josquin Research Project* https://josquin.stanford.edu/

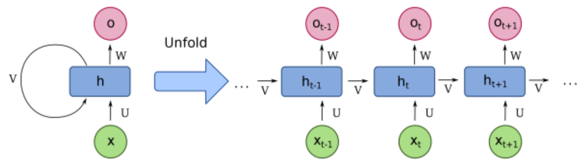
Where do LLMs Come From?

- Unlike Feed-Forward Neural Networks (FNNs) or Recurrent Neural Networks (RNNs), LLMs are relatively modern (2017).
- Model complexity has evolved accordingly.

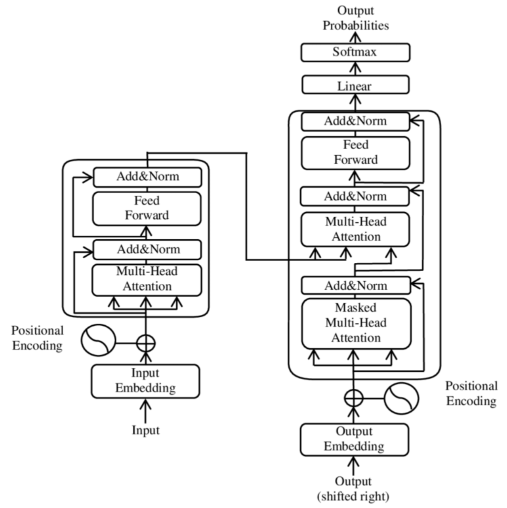
FNN:



RNN:



Transformer:



- *Attention is All You Need*, by the folks at Google Brain introduced the **Transformer** architecture, on which most LLMs are based.

- There plenty of variations on the Transfomer architecture, e.g.
  - *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context* (CMU/Google Brain).
  - *LLaMA: Open and Efficient Foundation Language Models* (Facebook).
  - *GPT-NeoX-20B: An Open-Source Autoregressive Language Model (*EleutherAI)
  - And many more!

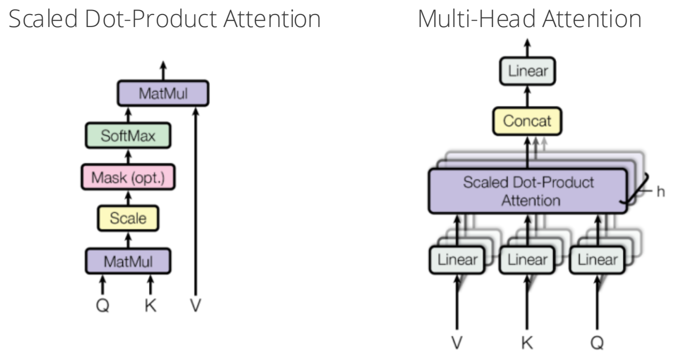
Attention Is All You Need https://arxiv.org/abs/1706.03762
Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context https://arxiv.org/abs/1901.02860
GPT-NeoX-20B: An Open-Source Autoregressive Language Model https://arxiv.org/abs/2302.13971
Images from Wikipedia pages for RNN, and Transformer

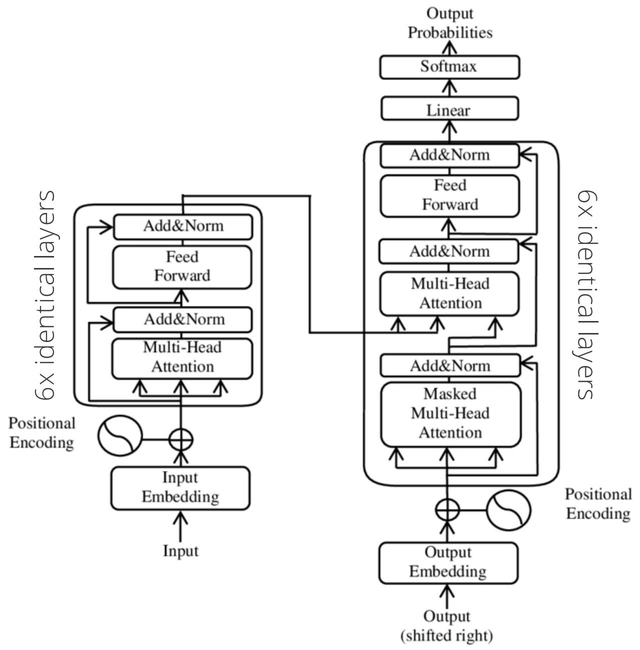
What is a Transformer?

- Big picture:
  - input is received, tokenized, encoded
  - secret sauce, **attention**
  - potential outputs are assigned probabilities, then selected

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Sauce:
  - Attention is a subspace similarity metric that allows the model to capture relevant information (think dot products!)
  - Multi-head attention projects into multiple subspaces, allowing simultaneously processing of different important information
  - "Prove there are infinitely many primes; do it in the style of a Shakespeare play through dialogue between two parties"
  - Self-attention connects all positions of output with a small number of operations- cheaper than RNNs for maintaining context

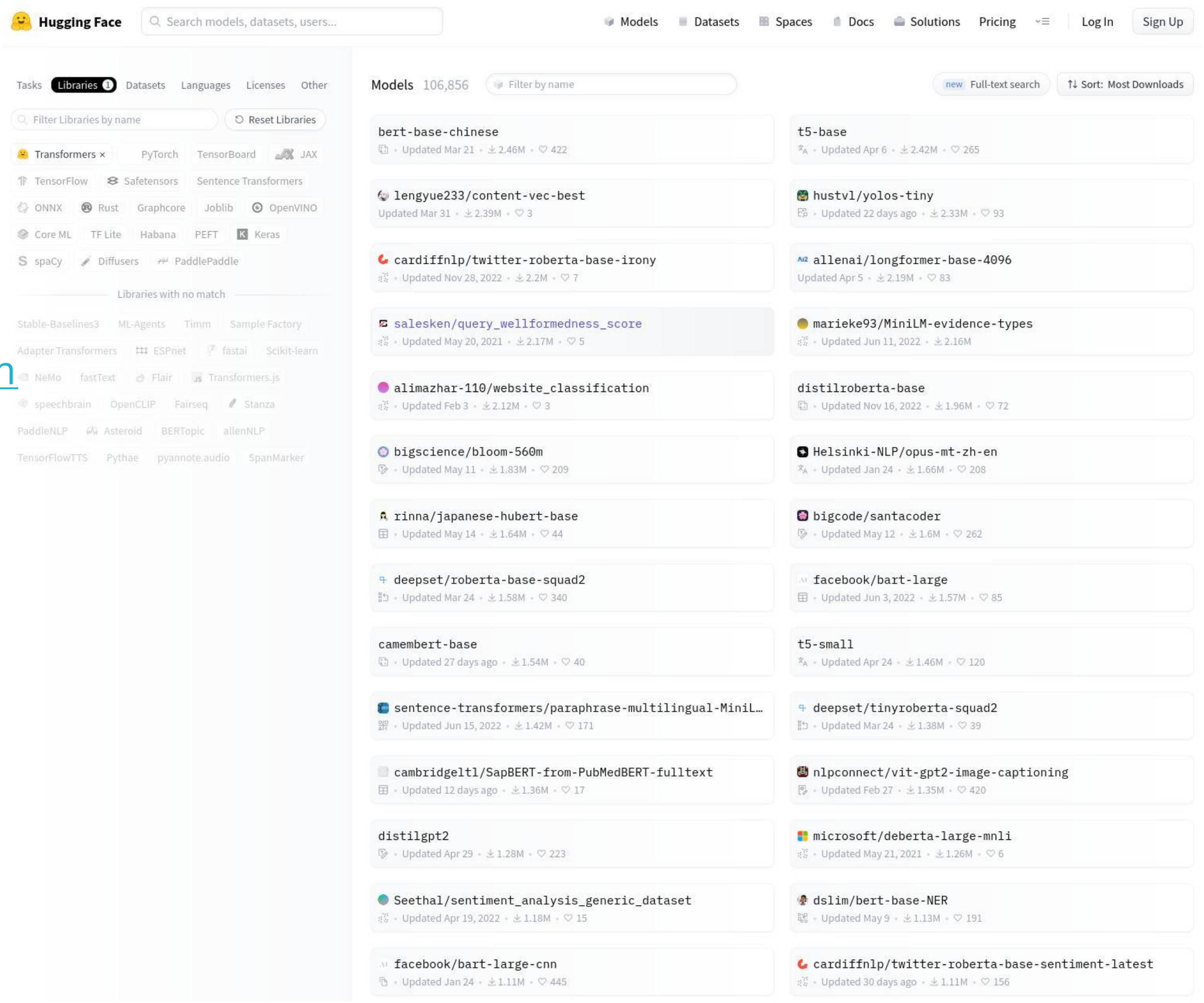




Where can you get LLMs?

- There are many open efforts to share LLMs
  - Huggingface
    https://huggingface.co/models?library=transformers
    (also see LLM leaderboards)
    https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard
  - Github
    https://github.com/openai

- Most useful beginner LLMs include
  - NanoGPT
    https://github.com/karpathy/nanoGPT



Models 106,856

bert-base-chinese
Updated Mar 21 • 2.46M • 422

t5-base
Updated Apr 6 • 2.42M • 265

lengyue233/content-vec-best
Updated Mar 31 • 2.39M • 3

hustvl/yolos-tiny
Updated 22 days ago • 2.33M • 93

cardiffnlp/twitter-roberta-base-irony
Updated Nov 28, 2022 • 2.2M • 7

allenai/longformer-base-4096
Updated Apr 5 • 2.19M • 83

salesken/query_wellformedness_score
Updated May 20, 2021 • 2.17M • 5

marieke93/MiniLM-evidence-types
Updated Jun 11, 2022 • 2.16M

alimazhar-110/website_classification
Updated Feb 3 • 2.12M • 3

distilroberta-base
Updated Nov 16, 2022 • 1.96M • 72

bigscience/bloom-560m
Updated May 11 • 1.83M • 209

Helsinki-NLP/opus-mt-zh-en
Updated Jan 24 • 1.66M • 208

rinna/japanese-hubert-base
Updated May 14 • 1.64M • 44

bigcode/santacoder
Updated May 12 • 1.6M • 262

deepset/roberta-base-squad2
Updated Mar 24 • 1.58M • 340

facebook/bart-large
Updated Jun 3, 2022 • 1.57M • 85

camembert-base
Updated 27 days ago • 1.54M • 40

t5-small
Updated Apr 24 • 1.46M • 120

sentence-transformers/paraphrase-multilingual-MiniL…
Updated Jun 15, 2022 • 1.42M • 171

deepset/tinyroberta-squad2
Updated Mar 24 • 1.38M • 39

cambridgeltl/SapBERT-from-PubMedBERT-fulltext
Updated 12 days ago • 1.36M • 17

nlpconnect/vit-gpt2-image-captioning
Updated Feb 27 • 1.35M • 420

distilgpt2
Updated Apr 29 • 1.28M • 223

microsoft/deberta-large-mnli
Updated May 21, 2021 • 1.26M • 6

Seethal/sentiment_analysis_generic_dataset
Updated Apr 19, 2022 • 1.18M • 15

dslim/bert-base-NER
Updated May 9 • 1.13M • 191

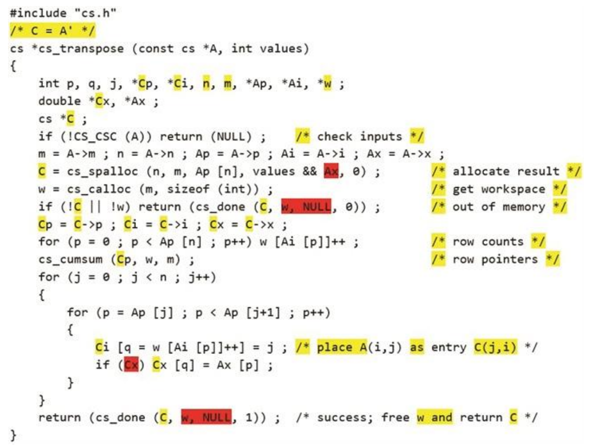
facebook/bart-large-cnn
Updated Jan 24 • 1.11M • 445

cardiffnlp/twitter-roberta-base-sentiment-latest
Updated 30 days ago • 1.11M • 156

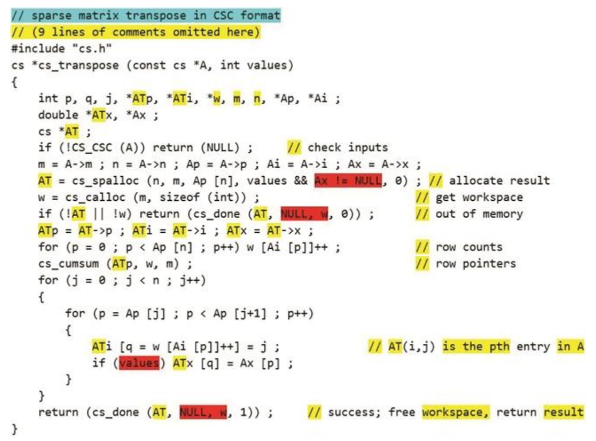
Why LLMs Make Lawyers Nervous

- There are legal issues at almost every facet
  - Plagiarism of copyrighted content without attribution
  - Improper use of AI-generated content
  - Defamatory statements about real people
  - Underlying core issue: model input/output should be checked

- Github Copilot has been shown to plagiarize copyrighted code without attribution



```
#include "cs.h"
/* C = A' */
cs *cs_transpose (const cs *A, int values)
{
    int p, q, j, *Cp, *Ci, n, m, *Ap, *Ai, *w ;
    double *Cx, *Ax ;
    cs *C ;
    if (!CS_CSC (A)) return (NULL) ;    /* check inputs */
    m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ; Ax = A->x ;
    C = cs_spalloc (n, m, Ap [n], values && Ax, 0) ;        /* allocate result */
    w = cs_calloc (m, sizeof (int)) ;                       /* get workspace */
    if (!C || !w) return (cs_done (C, w, NULL, 0)) ;        /* out of memory */
    Cp = C->p ; Ci = C->i ; Cx = C->x ;
    for (p = 0 ; p < Ap [n] ; p++) w [Ai [p]]++ ;           /* row counts */
    cs_cumsum (Cp, w, m) ;                                  /* row pointers */
    for (j = 0 ; j < n ; j++)
    {
        for (p = Ap [j] ; p < Ap [j+1] ; p++)
        {
            Ci [q = w [Ai [p]]++] = j ; /* place A(i,j) as entry C(j,i) */
            if (Cx) Cx [q] = Ax [p] ;
        }
    }
    return (cs_done (C, w, NULL, 1)) ;  /* success; free w and return C */
}
```



```
// sparse matrix transpose in CSC format
// (9 lines of comments omitted here)
#include "cs.h"
cs *cs_transpose (const cs *A, int values)
{
    int p, q, j, *ATp, *ATi, *w, m, n, *Ap, *Ai ;
    double *ATx, *Ax ;
    cs *AT ;
    if (!CS_CSC (A)) return (NULL) ;     // check inputs
    m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ; Ax = A->x ;
    AT = cs_spalloc (n, m, Ap [n], values && Ax != NULL, 0) ; // allocate result
    w = cs_calloc (m, sizeof (int)) ;                        // get workspace
    if (!AT || !w) return (cs_done (AT, NULL, w, 0)) ;       // out of memory
    ATp = AT->p ; ATi = AT->i ; ATx = AT->x ;
    for (p = 0 ; p < Ap [n] ; p++) w [Ai [p]]++ ;            // row counts
    cs_cumsum (ATp, w, m) ;                                  // row pointers
    for (j = 0 ; j < n ; j++)
    {
        for (p = Ap [j] ; p < Ap [j+1] ; p++)
        {
            ATi [q = w [Ai [p]]++] = j ;              // AT(i,j) is the pth entry in A
            if (values) ATx [q] = Ax [p] ;
        }
    }
    return (cs_done (AT, NULL, w, 1)) ;     // success; free workspace, return result
}
```

Concerns of code generation: https://sinews.siam.org/Details-Page/ethical-concerns-of-code-generation-through-artificial-intelligence
Lawsuits: https://www.reuters.com/technology/australian-mayor-readies-worlds-first-defamation-lawsuit-over-chatgpt-content-2023-04-05/
More lawsuits: https://www.theverge.com/2023/6/9/23755057/openai-chatgpt-false-information-defamation-lawsuit
Lawyer cites made-up cases using ChatGPT: https://apnews.com/article/artificial-intelligence-chatgpt-courts-e15023d7e6fdf4f099aa122437dbb59b
Large Libel Models: https://www2.law.ucla.edu/volokh/ailibel.pdf

Training/Using LLMs at Sandia

- SNL has a number of resources you can use for your LLMs.
- If you're just doing inference, basically any beefy workstation should do the job.
- If you want to train, you'll probably want an accelerator of some kind…
  - CEE resources: cee-a100-00X, cee-h100-00X, ampere
  - NM HPC resources: vortex, hops, doom
  - CA HPC resources: kahuna/glinda
  - Testbeds: various NVIDIA and AMD hardware
- If you need a functional python
  - SEMS: https://sems-atlassian-son.sandia.gov/confluence/display/SEMSKB/SEMS+TPL+Modules
  - AUE on CEE: `module load aue/python/3.11.6`

Training/Using LLMs at Sandia

- We'll focus on models from huggingface.co as their transformers library has 378,475 different models and/or model architectures built in.

- This includes stuff like GPT-2, BERT, GPT-BigCode, Llama, etc.

- Unfortunately, like many (most?) python tools, HF's transformers will *not* respect proxy servers in your environment and does not respond well to SSL interception.

- The **snl.py** tool in our repo fixes this: https://gitlab-ex.sandia.gov/csiefer/sandia-llm-tools
- This is also available in **shirty**: https://cee-gitlab.sandia.gov/classification/shirty/-/blob/develop/src/shirty/util/ssl.py

- What to expect next:
  - Inference on an existing model and conversational models
  - Fine-tuning an existing model
  - Training from scratch

For supported models in HF, see: https://huggingface.co/docs/transformers/model_doc/auto

Inference on an Existing Model (Smallish Models)

- Our **inference.py** script uses the HuggingFace AutoModel capability.

- Inside **inference.py** are basically five lines of code, where you can specify either a local directory, or a model on the HuggingFace website.

```
tokenizer = AutoTokenizer.from_pretrained(directory_or_remote_model)

model = AutoModelForCausalLM.from_pretrained(directory_or_remote_model)

pipe = pipeline("text-generation", model=model, tokenizer=tokenizer,pad_token_id=50256)

generated_text = pipe(prompt, max_length=num_tokens,
do_sample=True,no_repeat_ngram_size=2, early_stopping=True)[0]

 print(generated_text['generated_text'])
```

Which tokenizer to use

Which model to use

Define the pipeline

Generate text

Inference on an Existing Model: Example

```
./inference.py --model=cerebras/Cerebras-GPT-111M --prompt="We hold these truths to be self-evident" --num-runs=2

Loading tokenizer…
Loading model…
Prompt:  We hold these truths to be self-evident
------------------------------------
We hold these truths to be self-evident and truth-deaf.

(Cameron) We could write these words as:

_I_ _can only see what is _something_
_he _need_ to feel
------------------------------------
We hold these truths to be self-evident. Even if we assume that we've "been" before now, the true identity becomes more important and a form of self sustaining the unity that separates us and that of all of us is "be
```

The Same, Only Bigger (13B vs. 111M)

Note the bigger model!

```
./inference.py --model=cerebras/Cerebras-GPT-13B --prompt="We hold these truths to be self-evident" --num-runs=2

Loading tokenizer...
Loading model...
Prompt:  We hold these truths to be self-evident
------------------------------------
We hold these truths to be self-evident, that all men are created equal;
that they are endowed by their Creator with certain unalienable rights;
among these are life, liberty and the pursuit of happiness. That to secure
these rights,
------------------------------------
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness
```

Actual Declaration: We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness

Note: When critics suggest that LLMs are plagiarism factories, they're not kidding. Be careful!

Inference on an Existing Model: Advanced Topics

- By default, inference tries to use a single GPU. For CPU inference, you need to ask.
- For **inference.py,** you can do this with **--device=cpu**
- You can also control:
  - Random seed used
  - How many tokens you want to generate
  - How many different samples you want
  - What your initial prompt text is
- CPUs are generally sufficient for inference for small-to-medium sized models.
- Multi-GPU inference (for *big* models) is supported with --**device=<num_of_gpus_to_use>**

Inference via Shirty LLM Endpoints

- The ATLAS/Shirty team has deployed REST endpoints for a number of common LLMs.
- So if you want to use one of those off-the-shelf, they've got you covered!

```
./inference_shirty.py --model=Mixtral-8x22B-Instruct-v0.1 --prompt="What is the purpose of the union?"
```

Prompt: What is the purpose of the union?

The purpose of a union, often referred to as a labor union or trade union, is to protect and advocate for the rights, interests, and welfare of its members, who are typically workers and employees in a particular industry or company. Unions negotiate with employers on behalf of their members to establish better working conditions, wages, benefits, and work hours. They also work to ensure that employment policies are fair and equitable and address issues such as job security, health and safety standards, and retirement benefits. Furthermore, unions provide a platform for collective bargaining, allowing workers to have a stronger voice in decisions that affect their work lives.

Using Shirty LLM Endpoints

- You need to set up your API key first: https://cee-gitlab.sandia.gov/classification/shirty/blob/develop/docs/source/microservices/llm.md

```
### Load up the OpenAI stuff ###
client = OpenAI()

### Call Shirty to do Inference ###
print('Prompt: ',args.prompt)
completion = client.chat.completions.create( model=args.model, messages=[
{"role": "user", "content": args.prompt} ], max_tokens=args.num_tokens )

print(completion.choices[0].message.content)
```

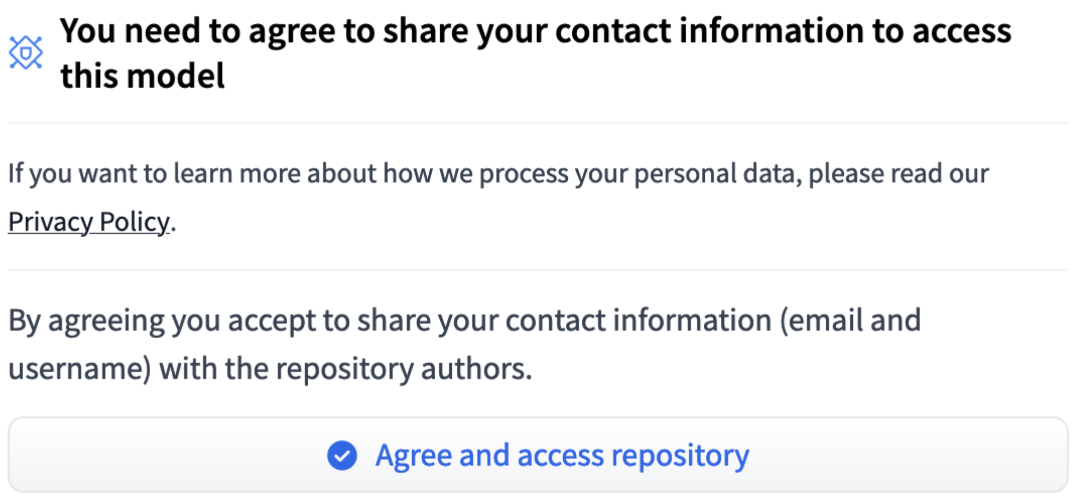
Shirty-hosted models as of 10/16/24: NousResearch/Meta-Llama-3-8B-Instruct, zephyr-7b-beta, thesven/Mistral-7B-Instruct-v0.3-GPTQ, neuralmagic/Meta-Llama-3.1-70B-Instruct-FP8 dolphin-2.7-mixtral-8x7b-AWQ

Accessing large models on Huggingface

- Some models are locked behind the huggingface website

```
Cannot access gated repo for url https://huggingface.co/mistralai/Mistral-7B-Instruct-
v0.2/resolve/main/config.json. Access to model mistralai/Mistral-7B-Instruct-v0.2 is restricted.
You must be authenticated to access it.
```

- In order to access them, authenticate through huggingface.
  - https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2



You need to agree to share your contact information to access this model

If you want to learn more about how we process your personal data, please read our Privacy Policy.

By agreeing you accept to share your contact information (email and username) with the repository authors.

- Then log in through the huggingface-cli tool (pip install huggingface_hub[cli])
- "**huggingface-cli login**", then follow instructions

The only token permissions required are read access to the model(s) you want to download

For Conversations we Need Chat Templates

- For models where we *converse*, we can use HuggingFace's chat_template functionality.
- Models marked as "conversational" are able to save and recall chat history.
- For example, run **conversation.py --device=cuda** on a gpu machine.
  - You may change the **--context="You are a helpful AI assistant"** to personalize your chatbot

```
./conversation.py --model=mistralai/Mistral-7B-Instruct-v0.2 --context="You are a pirate who likes to say 'arr' randomly"
Loading tokenizer...
Loading model: mistralai/Mistral-7B-Instruct-v0.2
Loading checkpoint shards: 100%|█████████████████████| 3/3 [00:02<00:00,  1.03it/s]
Prompt: How deep is the atlantic ocean?
Arr, the Atlantic Ocean is a vast body of water, arr, with an average depth of about 1,200 feet, or 365 meters,arr. But arr its deepest point, Arrgh, goes down to
Prompt: What is the scariest monster in that ocean?
Arrh, there be many a creature in the depths of the Atlaslican Ocean that could give a hearty pirATE heart a shiver, but the most famous one,Arrr, is certainly the Kraken, a gargant
Prompt: What is your favorite letter in the english alphabet?
Arr! That be a tough question, buccaneers! But I'd have to go with the letter "R"! Arr. It's got that curvy shape and it'll always be associated with pirates, parro
```

For Conversations we Need Chat Templates

- If **tokenizer.chat_template** does not exist, the model is not conversational.
- Chat history is stored in an array of dictionaries with the following format

```
[{"role": "system", "content" : "You are a pirate who likes to say 'arr' randomly"},
{"role": "user", "content" : "How deep is the atlantic ocean?"},
{"role": "assistant", "content" : "Arr, the Atlantic Ocean is a vast body of water, arr, with an average depth of about 1,200 feet, or 365 meters,arr. But arr its deepest point, Arrgh, goes down to"},
{"role": "user", "content" : "etc..."},
{"role": "assistant", "content" : "etc..."}]
```

- Then formatted for the specific LLM by **tokenizer.apply_chat_template(chat)**

```
<s> [INST] You are a pirate who likes to say 'arr' randomly
How deep is the atlantic ocean? [/INST]
Arr, the Atlantic Ocean is a vast body of water, arr, with an average depth of about 1,200 feet, or 365 meters,arr. But arr its deepest point, Arrgh, goes down to
```

- Chat templates are model-specific!

- See https://huggingface.co/docs/transformers/main/en/chat_templating for more information

Training a (Smallish) Model From Scratch

- We use a (modified) version of **run_clm.py** from transformers via **train_from_scratch.sh** wrapper.
- For existing model architectures, you specify the --**model_type** via the command line.
- We'll train using the Federalist Papers for training and the Bill of Rights for validation.
- What we actually ran:
  - **./train_from_scratch.sh gpt2-small federalist_papers.txt bill_of_rights.txt**
- Our options to **run_clm.py** to get a 68M parameter model (overkill, we know):
  - --model_type=gpt2 --tokenizer_name=gpt2 --torch_dtype=float16 --block_size=768 --max_steps=5000 --config_overrides=n_layer=4,n_head=4,n_positions=768
  - This took roughly 60 minutes on quad V100s.

We make no claim that the number of steps is sufficient

Doing Inference on Our From-Scratch Model

- **./inference.py --model=from-scratch-gpt2-small --prompt="It is not, however, my design to dwell upon observations of this nature." --num-runs=2**

```
Prompt:  It is not, however, my design to dwell upon observations of this nature.
------------------------------------
It is not, however, my design to dwell upon observations of this nature. It is no
there should be required, that it will see that such a variety infer to become the transient and general character of a separation; that the exclusive
------------------------------------
It is not, however, my design to dwell upon observations of this nature. It is little
Executive, in the statute the codes of consideration of Representatives; and the whole power of human nature, the people in the people. If
```

Actual text: It is not, however, my design to dwell upon observations of this nature. I am well aware that it would be disingenuous to resolve indiscriminately the opposition of any set of men (merely because their situations might subject them to suspicion) into interested or ambitious views.

Training a (Larger) Model From Scratch

- Bigger models require us to turn on the bells & whistles because we run out of GPU memory otherwise.
- In our case we use DeepSpeed. We include an example in **deepspeed3.json**.
- There are other multi-GPU training options out there, but **run_clm.py** supports deepspeed via CLI.
- Warning: You will need to have your CUDA modules loaded to use deepspeed.
- We'll try: **./train_from_scratch.sh gpt-neo federalist_papers.txt bill_of_rights.txt** to get a 1.4B parameter model.
- As before, we use an insufficient number of training steps…

Deepspeed: https://github.com/microsoft/DeepSpeed

Doing Inference on Our Bigger From-Scratch Model

- **./inference.py --model=from-scratch-gpt-neo --prompt="It is not, however, my design to dwell upon observations of this nature." --num-runs=2**

```
Prompt:  It is not, however, my design to dwell upon observations of this nature.
------------------------------------
It is not, however, my design to dwell upon observations of this nature. and peace the objects, share in thanending and whole thing the consequence, abuse are more apprehens will not be President. What of numbers: in the same
------------------------------------
It is not, however, my design to dwell upon observations of this nature. as the peace Executive shall be less papers by theFrom of the av, in some preferred
eligible andcontin seldom of time in defense canither be courts, of
```

Actual text: It is not, however, my design to dwell upon observations of this nature. I am well aware that it would be disingenuous to resolve indiscriminately the opposition of any set of men (merely because their situations might subject them to suspicion) into interested or ambitious views.

Bigger models generally need more training steps (which we didn't do).

Fine-Tuning a Model

- We can safely assume that the 200k words in the Federalist Papers aren't enough to train a general model from scratch.
- So what if we modified an existing model to look more like our data?
- This is called *fine tuning* and **run_clm.py** supports this too.
- Instead of specifying --**model_type** we specify --**model_name_or_path**.
- Our **fine_tune.sh** script shows how this can work (~90 minutes)
  - **./fine_tune.sh cerebras-111M federalist_papers.txt bill_of_rights.txt --max_steps=5000**

We make no claim that the number of steps is sufficient

Fine-Tuning a Model: Cerebras-111M

- **./inference.py --model=fine-tuning-cerebras-111M --prompt="It is not, however, my design to dwell upon observations of this nature." --num-runs=2**

```
Prompt:  It is not, however, my design to dwell upon observations of this nature.
-----------------------------------
It is not, however, my design to dwell upon observations of this nature. I
see not due to see such a distinction. The some of which I have
appeared in my case, no doubt, in most cases, are those which
-----------------------------------
It is not, however, my design to dwell upon observations of this nature. I
consider how far the trial by jury wears the uniformity of a Constitution
to regulate the temper of the human affairs. The ordinary hours of people
is generally
```

Actual text: It is not, however, my design to dwell upon observations of this nature. I am well aware that it would be disingenuous to resolve indiscriminately the opposition of any set of men (merely because their situations might subject them to suspicion) into interested or ambitious views.

Fine-Tuning a Model: Cerebras-2.7B

- **./inference.py --model=fine-tuning-cerebras-2.7B --prompt="It is not, however, my design to dwell upon observations of this nature." --num-runs=2**

Prompt: It is not, however, my design to dwell upon observations of this nature.

It is not, however, my design to dwell upon observations of this nature. Ishall add one further observation, which will perhaps be somewhat respectable, under the form presented to us. It is, that those who declare the mischiefs of the existing

----

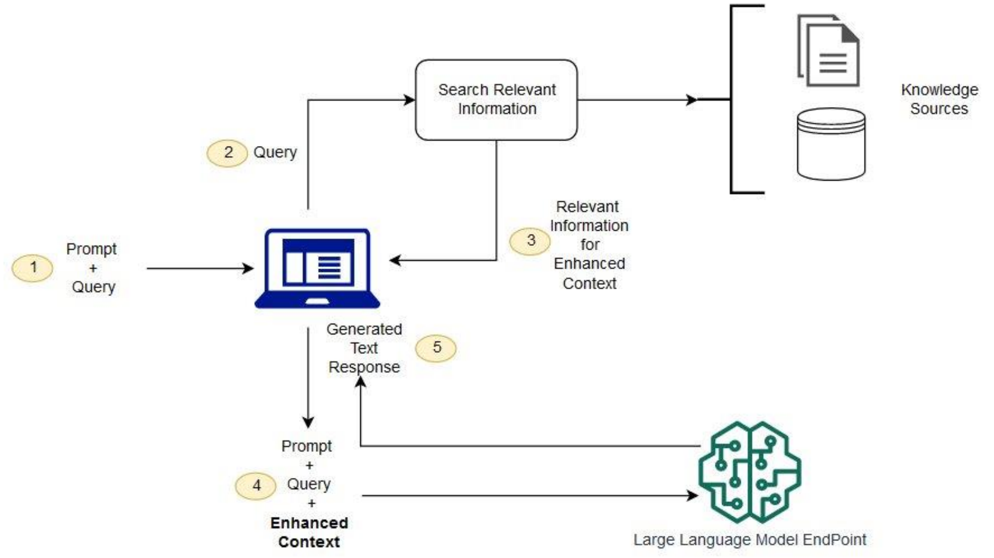
It is not, however, my design to dwell upon observations of this nature. I will only add that, in the first place, I SHALL ASSUMBLIES TO RESQUIRE TIME THEIR PUBLIC SUPPORT TO PRESERVE THE NECIABLIENCE OF

Actual text: It is not, however, my design to dwell upon observations of this nature. I am well aware that it would be disingenuous to resolve indiscriminately the opposition of any set of men (merely because their situations might subject them to suspicion) into interested or ambitious views.

Bigger models generally need more training steps (which we didn't do).

Retrieval-Augmented Generation (RAG)

- Say you had an LLM that doesn't know about your topic…. but you have a lot of topic data.
- Now you *could* fine-tune a model on your topic data. But that takes lots of GPU.
- Or you could search your topic data for the best matches and let the LLM summarize the output.
- This approach is called Retrieval-Augmented Generation (RAG) and lets small models punch *substantially* above their weight.



Search Relevant Information
Knowledge Sources
2 Query
Relevant Information for Enhanced Context
3
1
Prompt + Query
Generated Text Response
5
Prompt + Query + Enhanced Context
4
Large Language Model EndPoint

Image from: https://docs.aws.amazon.com/images/sagemaker/latest/dg/images/jumpstart/jumpstart-fm-rag.jpg

Inference without RAG

```
./inference.py --model=apple/OpenELM-270M-Instruct --prompt="What is the purpose of the union?"

Loading tokenizer...

Loading model...

Prompt:  What is the purpose of the union?

-------------------------------------

I've been reading a lot about unions and their role in the labor movement, and I'm curious about their purpose.


Their mission seems clear: to represent, bargain, organize, represent the interests of workers in their respective united labor organizations. They do this by negotiating collective bonds, representing their members in collectively biddable contracts, advocating for their rights, etc.
```

The model associates "union" with organized labor, which is not what we want here.

Inference with RAG

```
./rag_demo.py --model=apple/OpenELM-270M-Instruct--prompt="What is the purpose of the union?"

Loading RAG data:  rag_data

Splitting RAG text

Generating RAG vector store w/ embedding : BAAI/bge-large-en-v1.5

Loading the LLM: cerebras/Cerebras-GPT-111M

User prompt:  What is the purpose of the union?

------------------------------------

Answer: The preservation of the Union, as a whole, and the superintendence of intercourse, political and commercial, with foreign countries.

The first advantage consists in the preservation of the Union, as a whole, and the superintendence of intercourse, political and commercial, with foreign countries. The former is the most essential, and the consequence of the union, as a whole, is the preservation of the Union as a whole. The United States, as a whole, possesses the necessary power, the necessary means, and the necessary influence, to safeguard itself against external aggression, and to promote peace and tranquillity in foreign countries.
```

Now this is a lot better! It clearly associates "The Union" with "The United States"

Database Search Results: What is the purpose of the union?

From the first document returned by the Retriever:

THE necessity of a Constitution, at least equally energetic with the one proposed, to the preservation of the Union, is the point at the examination of which we are now arrived. This inquiry will naturally divide itself into three branches—the objects to be provided for by the federal government, the quantity of power necessary to the accomplishment of those objects, the persons upon whom that power ought to operate. Its distribution and organization will more properly claim our attention under the succeeding head. The principal purposes to be answered by union are these--the common defense of the members; the preservation of the public peace as well against internal convulsions as external attacks; the regulation of commerce with other nations and between the States; the superintendence of our intercourse, political and commercial, with foreign countries.

The first line of the LLM's reponse using RAG:

Answer: The preservation of the Union, as a whole, and the superintendence of intercourse, political and commercial, with foreign countries.

Chainlit: Hosting your own web interface

- Say you had a custom LLM, but your coworkers want to use a web interface.
- Chainlit lets you do that!
- Three major pieces you have to write:
  - History storage mechanism
  - Loading model / prepping langchain: `on_chat_start`
  - Message response inference : `on_message`
- You can do a whole lot more with Chainlit, but those are the basics.

History Storage

- Define a function which is used to query your history structure:

```
def get_session_history(user_id: str, conversation_id: str) -> BaseChatMessageHistory:
    if (user_id, conversation_id) not in store:
        store[(user_id, conversation_id)] = InMemoryHistory()
    return store[(user_id, conversation_id)]
```

- Using a dictionary of lists seems to be a common choice for simple examples

- See: https://api.python.langchain.com/en/latest/runnables/langchain_core.runnables.history.RunnableWithMessageHistory.html

Getting chainlit ready: on_chat_start

```
### Get the model ###
settings = await cl.ChatSettings(
        [
            Select(
                id="Model",
                label="Huggingface Model",
                values=["family/Modelname"],
                initial_index=0,
            )
        ]
    ).send()
model = settings["Model"]
```

```
### Define a RunnableWithMessageHistory ###
h_chain = RunnableWithMessageHistory(
      chain,
      get_session_history,
      input_messages_key="question",
      history_messages_key="history",
      history_factory_config=[
      ConfigurableFieldSpec(
          id="user_id",
          annotation=str,
          name="User ID",
          default="",
          is_shared=True,
      ),
      ConfigurableFieldSpec(
          id="conversation_id",
          annotation=str,
          name="Conversation ID",
          default="",
          is_shared=True,
      )
      ]
  )
```

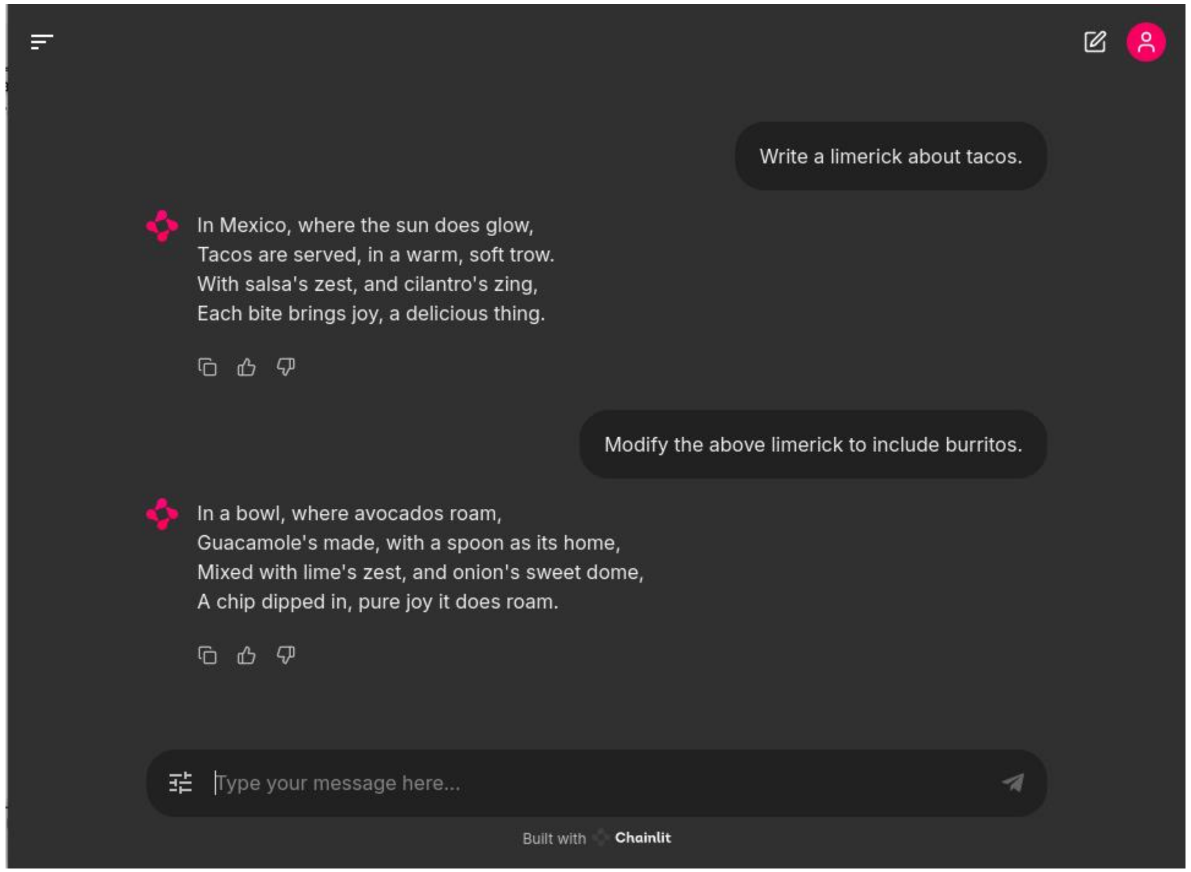
Message response: on_message

```
async def on_message(message: cl.Message):
  # Get inputs from chainlit
  runnable = cl.user_session.get("runnable")
  uid = cl.user_session.get("id")

  # Call the LLM
  msg = cl.Message(content="")
  response = runnable.invoke({"question":message.content},config={"user_id": uid,
"conversation_id":"1"})

  # Send the response to chainlit
  await msg.stream_token(response)
  await msg.send()
```

What does my Chainlit site look like?



Write a limerick about tacos.

In Mexico, where the sun does glow,
Tacos are served, in a warm, soft trow.
With salsa's zest, and cilantro's zing,
Each bite brings joy, a delicious thing.

Modify the above limerick to include burritos.

In a bowl, where avocados roam,
Guacamole's made, with a spoon as its home,
Mixed with lime's zest, and onion's sweet dome,
A chip dipped in, pure joy it does roam.

Pretty sure that's a burrito bowl and not a burrito, but anyway...

Wrapping Up

- Large Language Models are something *you* can explore at Sandia!
- Be sure to take legal and cyber issues into account!
- Feel free to use our wrappers as a jumping off point for your own explorations
  - https://gitlab-ex.sandia.gov/csiefer/sandia-llm-tools
  - https://gitlab-ex.sandia.gov/gbharpe/llm-training-data
- Tomorrow @ 11:15: Integrating Shirty with your favorite editor [1].

[1] As long as you like vim, emacs or vscode...