Large Language Models

Introduction and Use Cases

Contents / Agenda

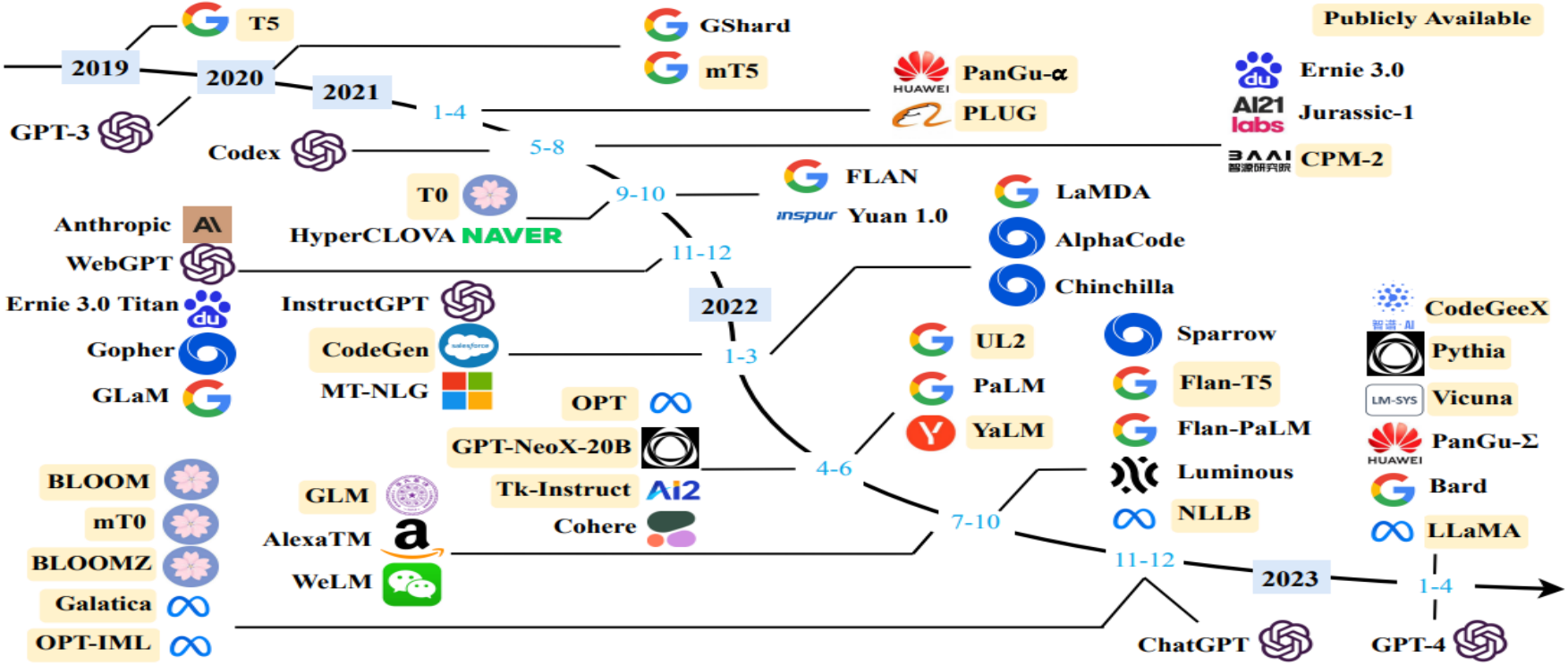
- Executive Summary
- Timeline of existing Large Language Models
- Statistics of commonly-used data sources
- Available Libraries for developing LLMs
- Data sources in the Pre-Training for LLMs
- Pre-training steps
- COT Learning Mechanism
- Knowledge recency
- Use cases
- Sample Architecture using Gatortron
- Real World Applications

Executive Summary

Ever since the **Turing Test** was proposed in the 1950s, humans have explored the mastering of language intelligence by machine. Language is essentially a complex, intricate system of human expressions governed by grammatical rules. It poses a significant challenge to develop capable **artificial intelligence** (AI) algorithms for **comprehending** and grasping a language. As a major approach, **language modeling** has been widely studied for language understanding and generation in the past two decades, evolving from statistical language models to neural language models.

Recently, the research on **Large Language Models (LLMs)** has been largely advanced by both academia and industry, and a remarkable progress is the launch of **ChatGPT** (a powerful AI chatbot developed based on LLMs), which has attracted widespread attention from society. The technical evolution of LLMs has been making an important impact on the entire AI community, which would revolutionize the way how we develop and use AI algorithms. Considering this rapid technical progress, we will review, at a high-level some of the important terms, techniques and examples that we could apply LLMs for.

Timeline of existing Large Language Models (> 10B)



Statistics of commonly-used data sources

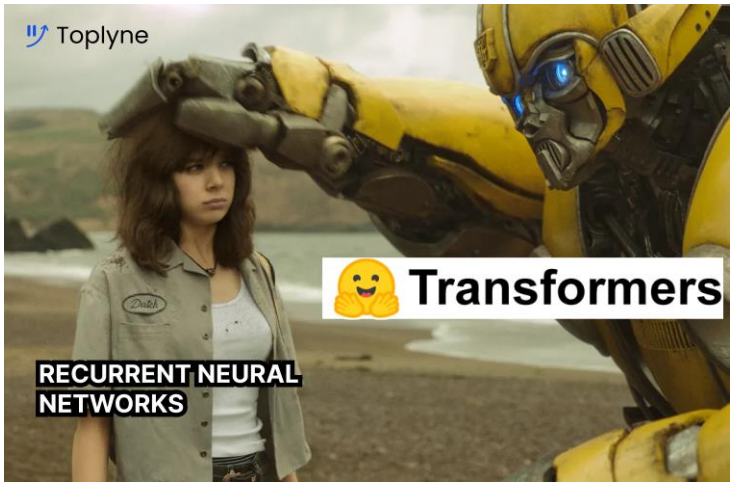
Corpora	Size	Source	Latest Update Time
BookCorpus	5GB	Books	Dec-2015
Gutenberg	-	Books	Dec-2021
C4	800GB	CommonCrawl	Apr-2019
CC-Stories-R	31GB	CommonCrawl	Sep-2019
CC-NEWS	78GB	CommonCrawl	Feb-2019
REALNEWs	120GB	CommonCrawl	Apr-2019
OpenWebText	38GB	Reddit Links	Mar-2023
Pushift.io	2TB	Reddit Links	Mar-2023
Wikipedia	21GB	Wikipedia	Mar-2023
BigQuery	-	Codes	Mar-2023
The Pile	800GB	Other	Dec-2020
ROOTS	1.6TB	Other	Jun-2022

- https://platform.openai.com/docs/api-reference/introduction
- https://platform.openai.com/docs/models/overview
- https://www.tensorflow.org/datasets/catalog/c4

Available Libraries for developing LLMs - Transformers

Transformers is an open-source Python library for building models using the Transformer architecture, which is developed and maintained by **Hugging Face**. It has a simple and user-friendly API, making it easy to use and customize various pre-trained models. It is a powerful library with a large and active community of users and developers who regularly update and improve the models and algorithms.

https://huggingface.co/learn/nlp-course/chapter1/4



Available Libraries for developing LLMs - DeepSpeed

DeepSpeed is a deep learning optimization library (compatible with PyTorch) developed by Microsoft, which has been used to train a number of LLMs, such as Megatron-Turing Natural Language Generation (**MTNLG**) and BigScience Large Open-science Open Access Multilingual Language Model (**BLOOM**). It provides the support of various optimization techniques for distributed training, such as memory optimization (ZeRO technique, gradient checkpointing), and pipeline parallelism.

MTNLG: https://developer.nvidia.com/megatron-turing-natural-language-generation
BLOOM: https://cobusgreyling.medium.com/bloom-bigscience-large-open-science-open-access-multilingual-language-model-b45825aa119e

Available Libraries for developing LLMs – Megatron-LM

Megatron-LM is a deep learning library developed by NVIDIA for training large-scale language models. It also provides rich optimization techniques for distributed training, including model and data parallelism, mixed-precision training, and Flash-Attention. These optimization techniques can largely improve the training efficiency and speed, enabling efficient distributed training across GPUs.

https://arxiv.org/abs/1909.08053

Available Libraries for developing LLMs – JAX

JAX is a Python library for high-performance machine learning algorithms developed by Google, allowing users to easily perform computations on arrays with hardware acceleration (e.g., GPU or TPU). It enables efficient computation on various devices and also supports several featured functions, such as automatic differentiation and just-in-time compilation.

https://github.com/google/maxtext

Available Libraries for developing LLMs – Colossal-AI

Colossal-AI is a deep learning library developed by HPC-AI Tech for training large-scale AI models. It is implemented based on PyTorch and supports a rich collection of parallel training strategies. Furthermore, it can also optimize heterogeneous memory management with methods proposed by PatrickStar. Recently, a ChatGPT-like model called Colossal Chat has been publicly released with two versions (7B and 13B), which are developed using Colossal-AI based on LLaMA.

https://colossalai.org
https://arxiv.org/abs/2110.14883
https://www.hpc-ai.tech/blog/colossal-ai-chatgpt
https://ai.facebook.com/blog/large-language-model-llama-meta-ai/

Available Libraries for developing LLMs – BMTrain

BMTrain is an efficient library developed by OpenBMB for training models with large-scale parameters in a distributed manner, which emphasizes code simplicity, low resource, and high availability. BMTrain has already incorporated several common LLMs (e.g., Flan-T5 [**F**inetuning **lan**guage models - **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer] and General Language Model [GLM]) into its Model Center, where developers can use these models directly.

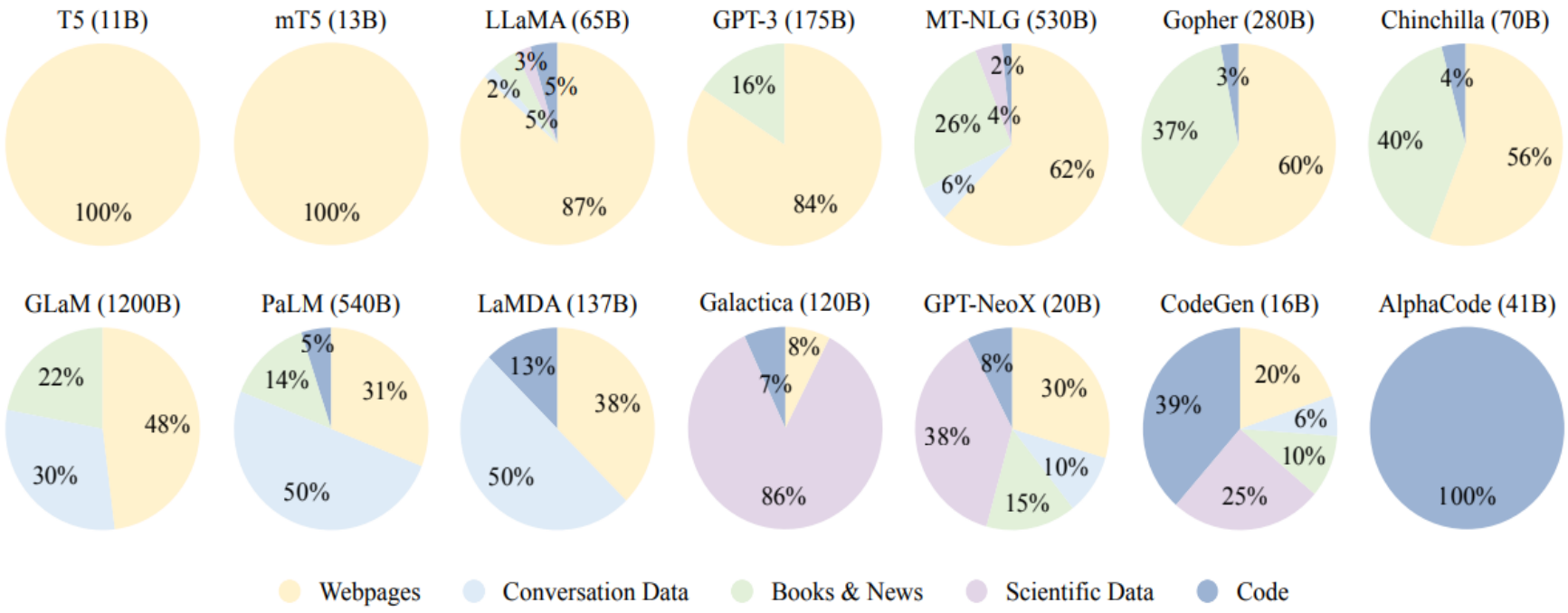
https://github.com/OpenBMB/BMTrain
https://github.com/google-research/FLAN
https://huggingface.co/google/flan-t5-xxl
https://console.paperspace.com/github/graphcore/Gradient-HuggingFace?machine=Free-IPU-POD4&container=graphcore/pytorch-jupyter%3A3.2.0-ubuntu-20.04-20230331&file=natural-language-processing%2FFlan-T5-generative-inference.ipynb
https://medium.com/geekculture/list-of-open-sourced-fine-tuned-large-language-models-llm-8d95a2e0dc76
https://arxiv.org/abs/2103.10360
https://aws.amazon.com/blogs/machine-learning/zero-shot-prompting-for-the-flan-t5-foundation-model-in-amazon-sagemaker-jumpstart/#:~:text=Flan-T5%20is%20an%20instruction,shot%20in-context%20learning%20tasks.

Available Libraries for developing LLMs – FastMOE

FastMoE is a specialized training library for MoE (i.e., mixture-of-experts) models. It is developed based on PyTorch, prioritizing both efficiency and user-friendliness in its design. FastMoE simplifies the process of transferring Transformer models to MoE models and supports both data parallelism and model parallelism during training.

https://arxiv.org/abs/2103.13262

Data sources in the pre-training for LLMs



Pre-Training Steps

Raw Corpus

Quality Filtering

- Language Filtering
- Metric Filtering
- Statistic Filtering
- Keyword Filtering

Alice is writing a paper about LLMs. ~~#$^&~~ Alice is writing a paper about LLMs.

De-duplication

- Sentence-level
- Document-level
- Set-level

Alice is writing a paper about LLMs. ~~Alice is writing a paper about LLMs.~~

Privacy Reduction

- Detect Personality Identifiable Information (PII)
- Remove PII

`Replace('Alice')` is writing a paper about LLMs.

Tokenization

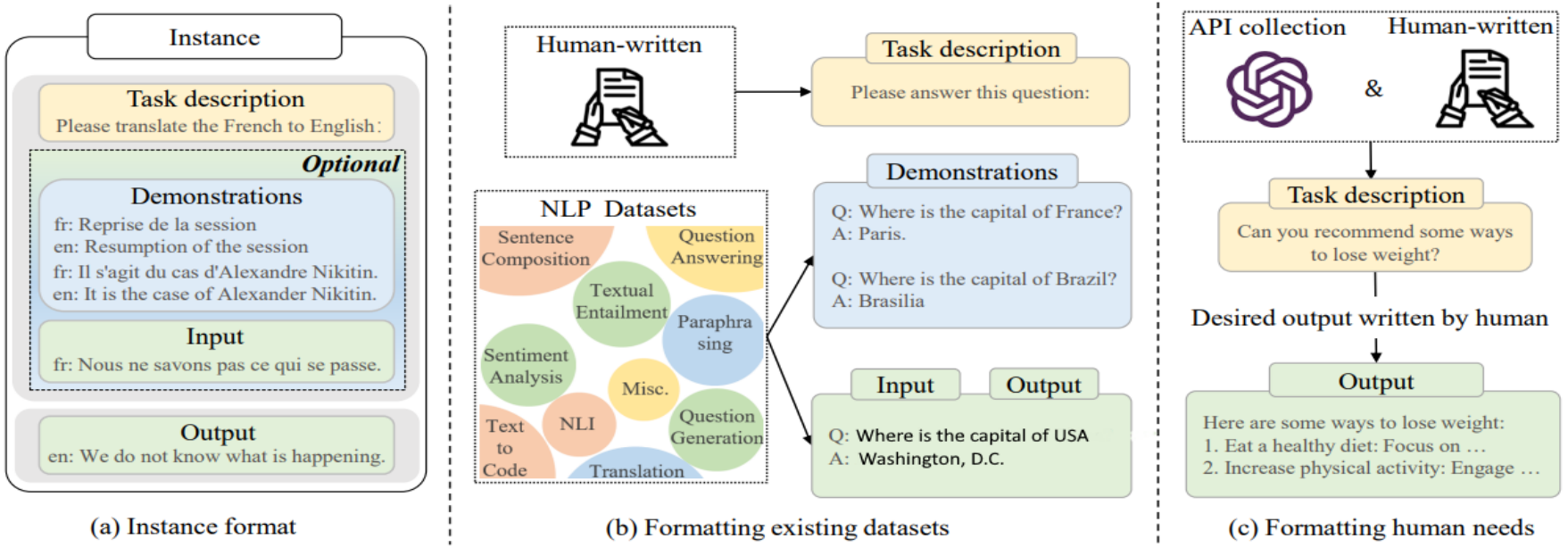
- Reuse Existing Tokenizer
- SentencePiece
- Byte-level BPE

`Encode('[Somebody] is writing a paper about LLMs.')`

Ready to pre-train!

32, 145, 66, 79, 12, 56, ...

COT Learning Mechanism

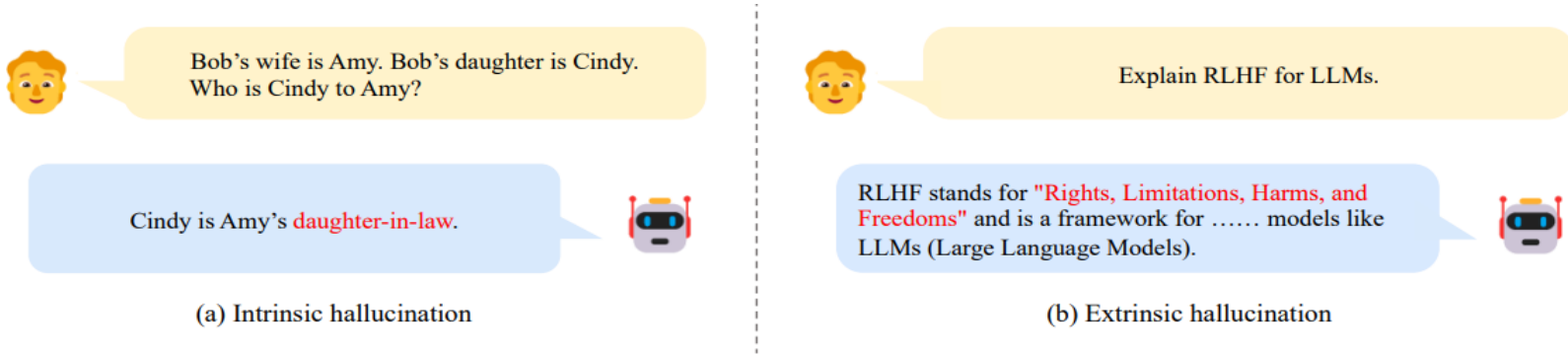


https://huggingface.co/joaoalvarenga/bloom-8bit
https://huggingface.co/hivemind/gpt-j-6B-8bit
https://huggingface.co/THUDM/chatglm-6b-int8
https://laion.ai/blog/oig-dataset/

Knowledge Recency

A major challenge, LLMs would encounter difficulties when solving tasks that require the latest knowledge beyond the training data. To tackle this issue, a straightforward approach is to regularly update LLMs with new data.

Note: Costly to fine-tune LLMs and Catastrophic forgetting issue when incrementally training LLMs



Use Case: Adverse Events

KNOWLEDGE CUTOFF: Product X has been shown to cause adverse events in patients with high BMI.
Patients >24.1 BMI and <26 BMI have shown a 10% incidence of Adverse Events.
Patients >26 BMI have a 20% incidence of Adverse Events.
QUESTION: My Patient weighs 170 pounds and is 5ft 8 inches.
What is the likelihood of an Adverse Event if he takes Product X?

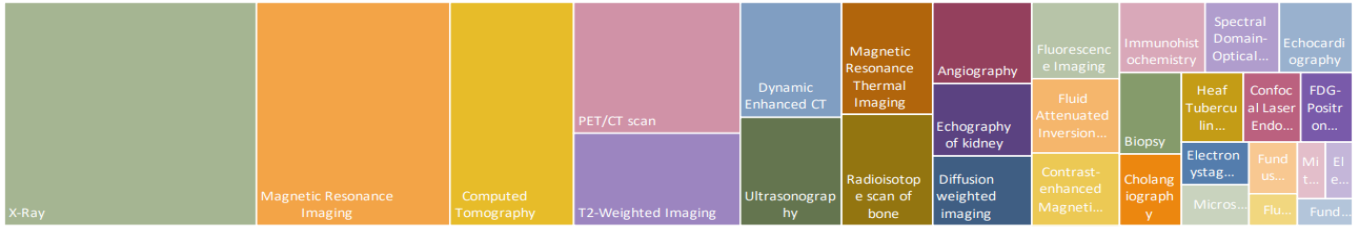
ChatGPT currently gives us this response;

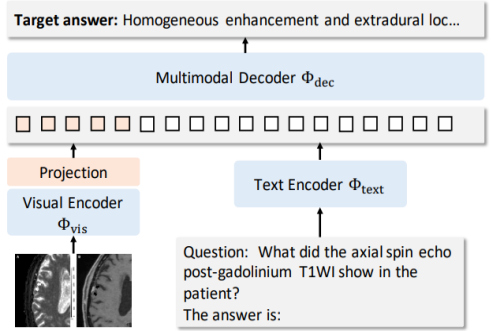
To calculate the likelihood of an adverse event for the given patient, we first need to convert their weight and height into the appropriate units for BMI calculation.
1 pound is approximately 0.4536 kilograms, and 1 foot is approximately 0.3048 meters.
Patient's weight: 170 pounds * 0.4536 kg/pound = 77.11 kg
Patient's height: 5 ft * 0.3048 m/ft + 8 in * 0.0254 m/in = 1.7272 meters
Now we can calculate the patient's BMI:
BMI = weight (kg) / (height (m))^2
BMI = 77.11 kg / (1.7272 m)^2
BMI ≈ 25.81
Based on the information provided, patients with a BMI greater than 26 have a 20% incidence of adverse events, while patients with a BMI between 24.1 and 26 have a 10% incidence of adverse events.
Since the patient's BMI is approximately 25.81, which falls between 24.1 and 26, we can assume a 10% incidence of adverse events for this patient if they take Product X.

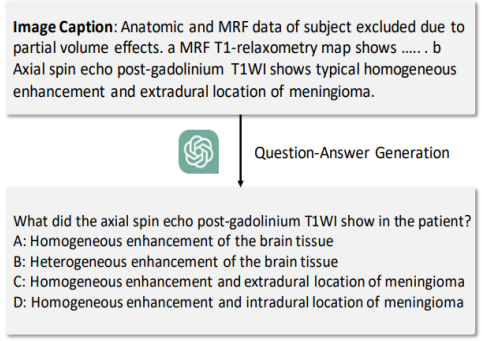
Use Case: Clinical Information Extractors

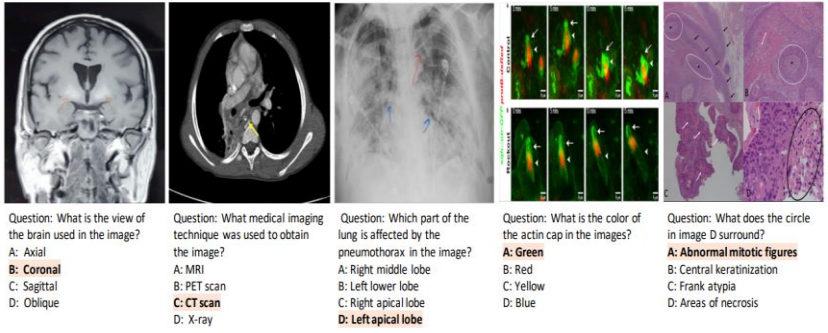
Clinical text contains a large amount of valuable information that is not captured by the structured data fields in electronic health records. However, there are significant challenges to clinical information extraction. Because clinical text contains irregularities such as ambiguous jargon and nonstandard phrasal structure, most off-the-shelf NLP tools perform poorly, and clinical text annotation requires domain expertise. To overcome these issues, we recommend using LLMs such as BERT, many clinical and biomedical variations swiftly followed including ClinicalBERT, SciBERT, BioBERT, and PubMedBERT.

Use Case: Medical Image Question Answering

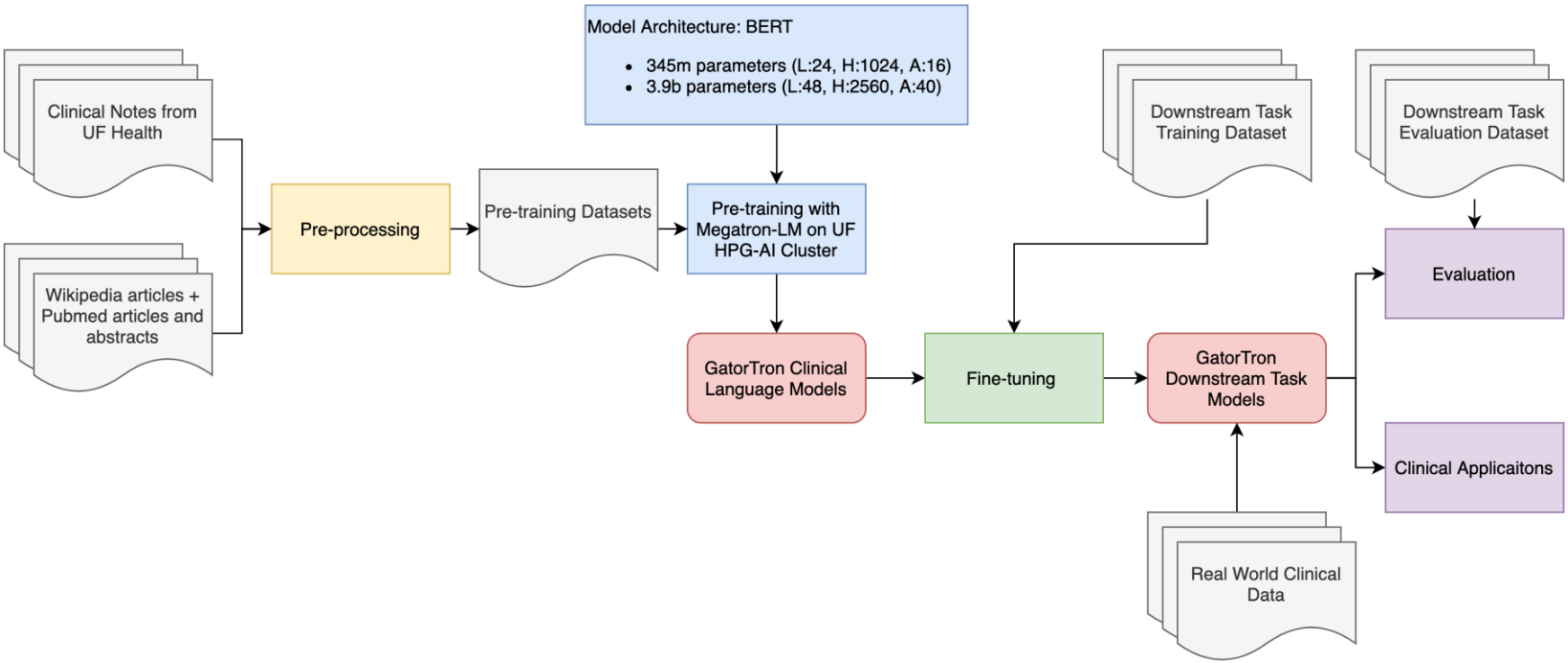








Sample Architecture using GatorTron



Real-world Applications

- GPT-3 (and ChatGPT), LaMDA, Character.ai, Megatron-Turing NLG – Text generation useful especially for dialogue with humans, as well as copywriting, translation, and other tasks
- PaLM – LLM from Google Research that provides several other natural language tasks
- Anthropic.ai – Product focused on optimizing the sales process, via chatbots and other LLM-powered tools
- BLOOM – General purpose language model used for generation and other text-based tasks, and focused specifically on multi-language support
- Codex (and Copilot), CodeGen – Code generation tools that provide auto-complete suggestions as well as creation of entire code blocks
- DALL-E, Stable Diffusion, MidJourney – Generation of images based on text descriptions
- Imagen Video – Generation of videos based on text descriptions
- Whisper – Transcription of audio files into text

Happy Learning !

https://github.com/braincomputingsantosh