Benchmarking LLMs

(for Particle Physics Understanding)

Student: Eugene Shalugin

Supervisor: Sascha Caron

Radboud University

What is Understanding ?

The degree to which agent A scientifically understands phenomenon P can be determined by assessing the extent to which

(i) A has a sufficiently complete representation of P;

(ii) A can generate internally consistent and empirically adequate explanations of P;

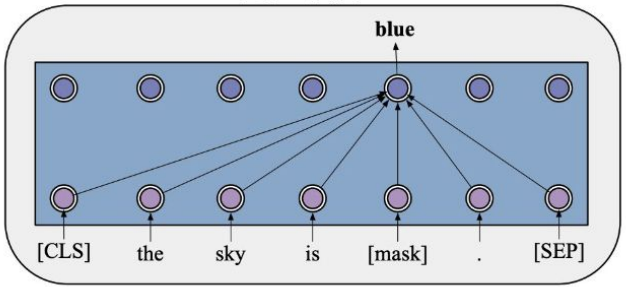
(iii) A can establish a broad range of relevant, correct counterfactual inferences regarding P.

(i-iii) can be measured, given a certain context (series of prompts) via what-, why, and w-questions respectively.

Towards a Benchmark for Scientific Understanding in Humans and Machines. *2023* K. G. Barman et. al.

What is LLM ?

BERT

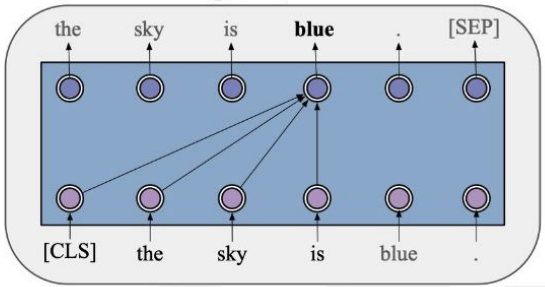


- Good at natural language understanding
- Maximizes the likelihood:

$$\mathcal{L}(\mathcal{X}) = \sum_{i=1}^{m} \log P(\texttt{[Mask]}_i = y_i | \tilde{\mathcal{X}}; \Theta)$$

  - P is modeled by transformer *encoder*
  - $\tilde{\mathcal{X}}$ *:* text after masking some tokens

GPT



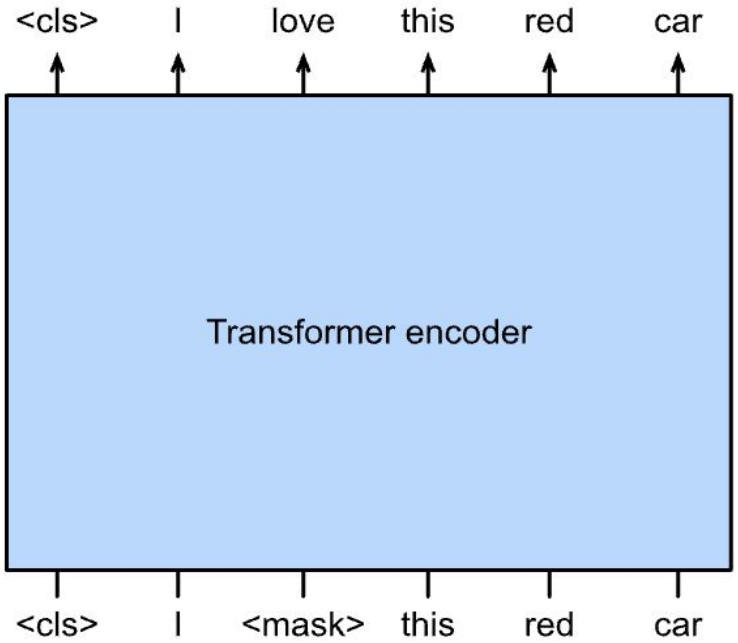
- Good at natural language generation
- Maximizes the likelihood:

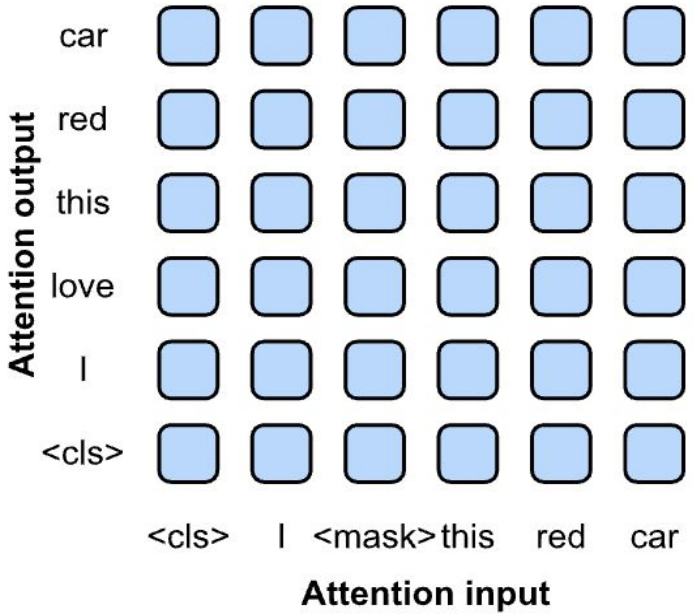
$$\mathcal{L}(\mathcal{X}) = \sum_{i=1}^{n+1} \log P(x_i | x_{i-k}, ..., x_{i-1}; \Theta)$$

  - P is modeled by transformer *decoder*
  - *Self attention: left to right*

Image source: Pre-Trained Models: Past, Present and Future [Han et al. 2021]

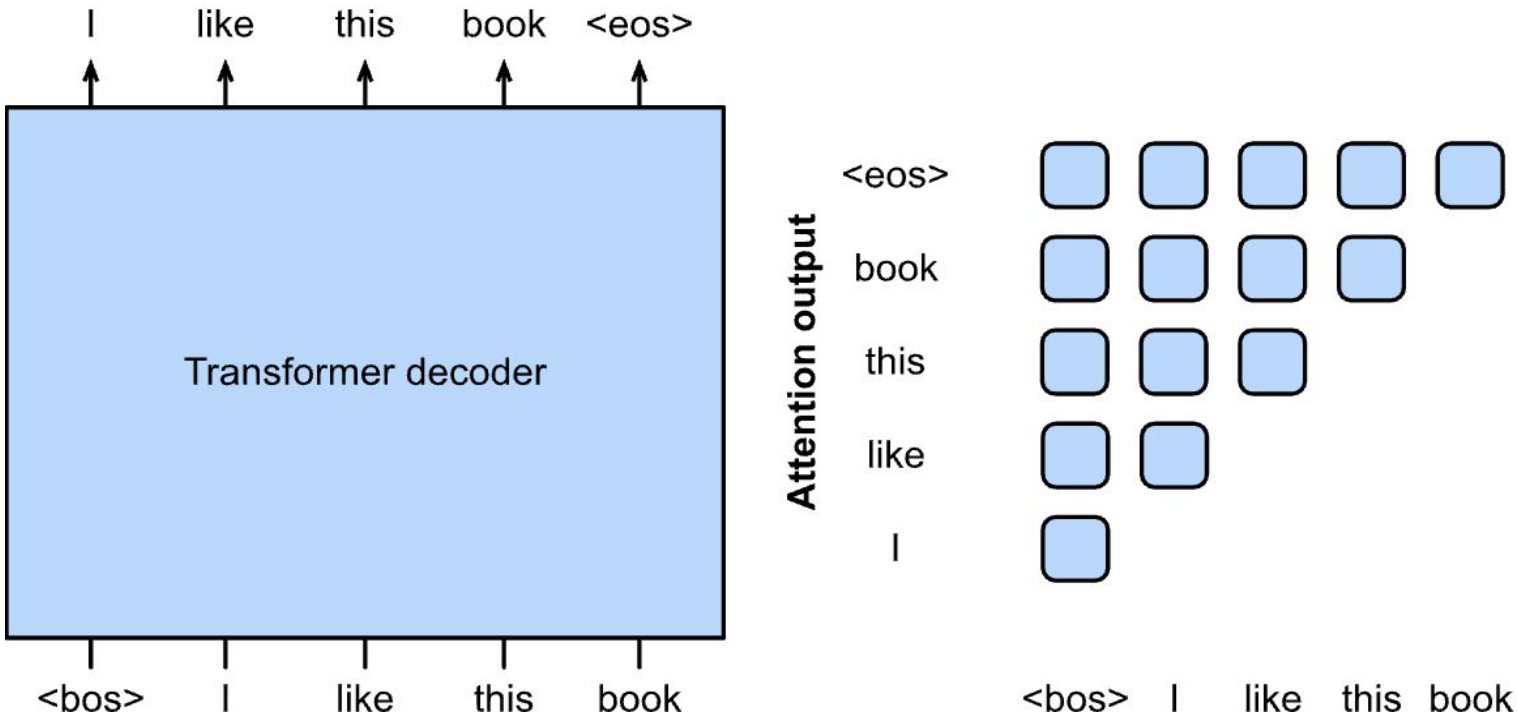
What is LLM – BERT





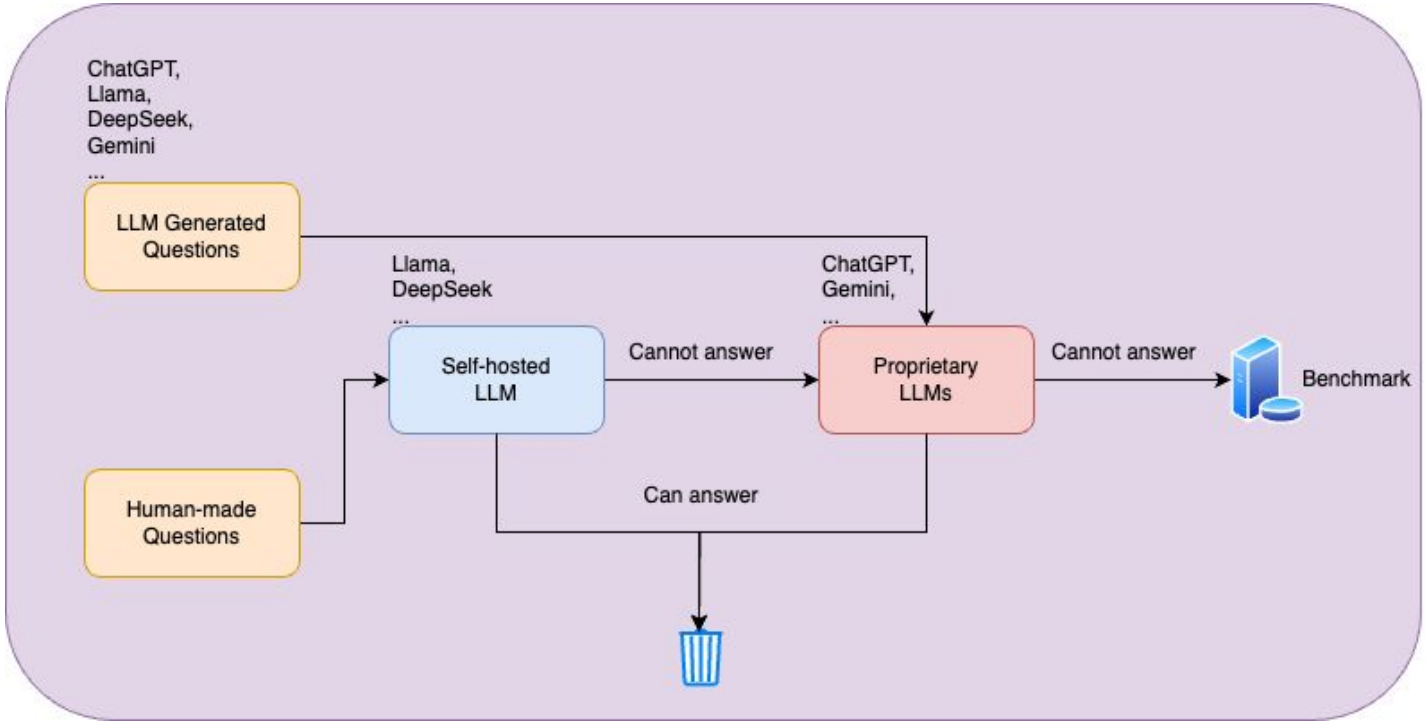
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Devlin et al., 2018
Illustration by Twan van Laarhoven.

What is LLM – GPT



Improving language understanding by generative pre-training. Radford et al. (OpenAI), 2018
Illustration by Twan van Laarhoven.

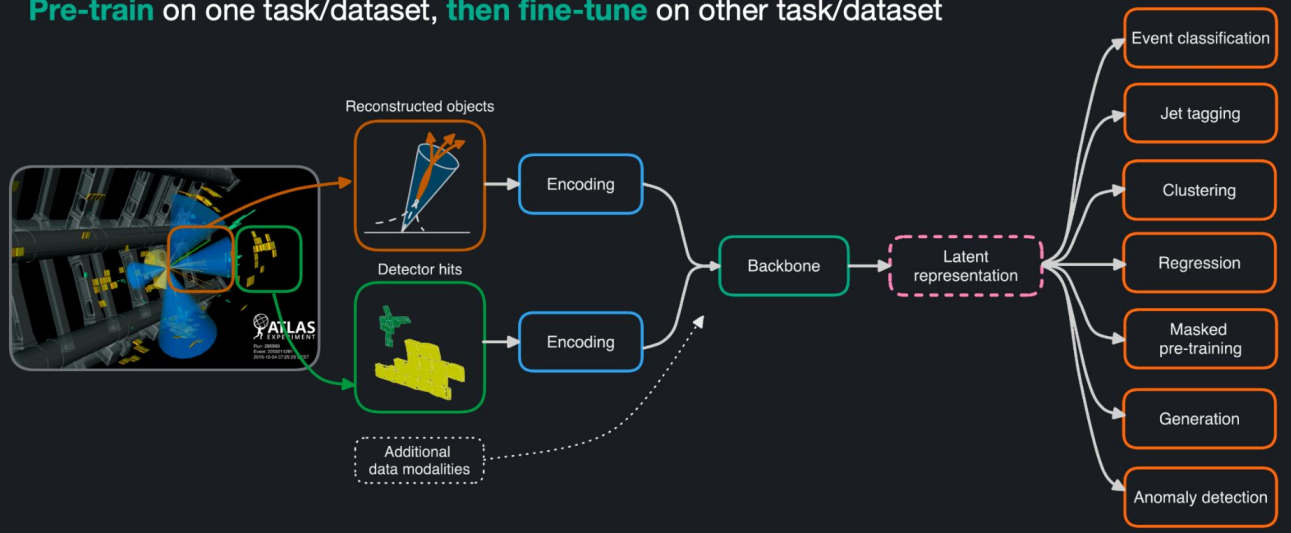
So what's with Benchmarking



Foundation Models

Foundation models for HEP

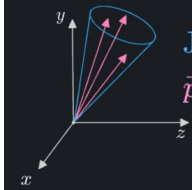
Pre-train on one task/dataset, **then fine-tune** on other task/dataset



Foundation Models

Our approach

Jet constituents with **continuous features**



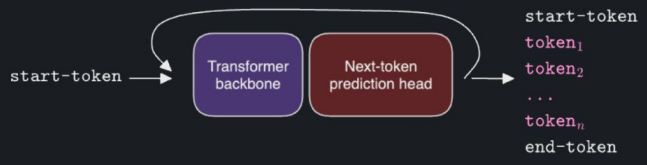
$$\text{Jet} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$$
$$\vec{p}_i = (p_\text{T}, \eta^\text{rel}, \phi^\text{rel})$$

Constituents are **tokenized with a VQ-VAE**
(using the approach presented by Sam Klein earlier)

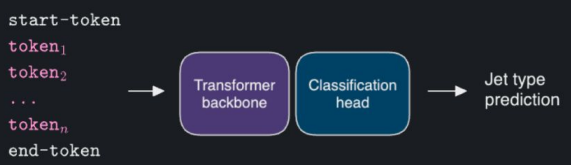
$$\text{Jet} = \{\texttt{start-token}, \texttt{token}_1, \ldots, \texttt{token}_n, \texttt{end-token}\}$$
$$\texttt{token}_i = \text{integer value} \in [1, \ldots, 8192]$$

Unsupervised pre-training of transformer backbone
on generative task (next-token prediction)

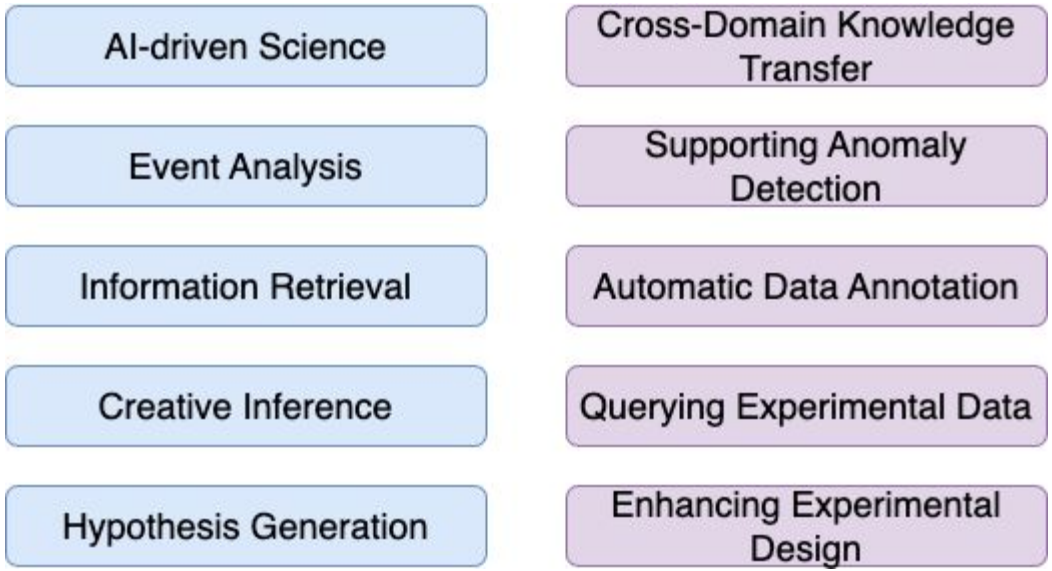
Autoregressive next-token generation



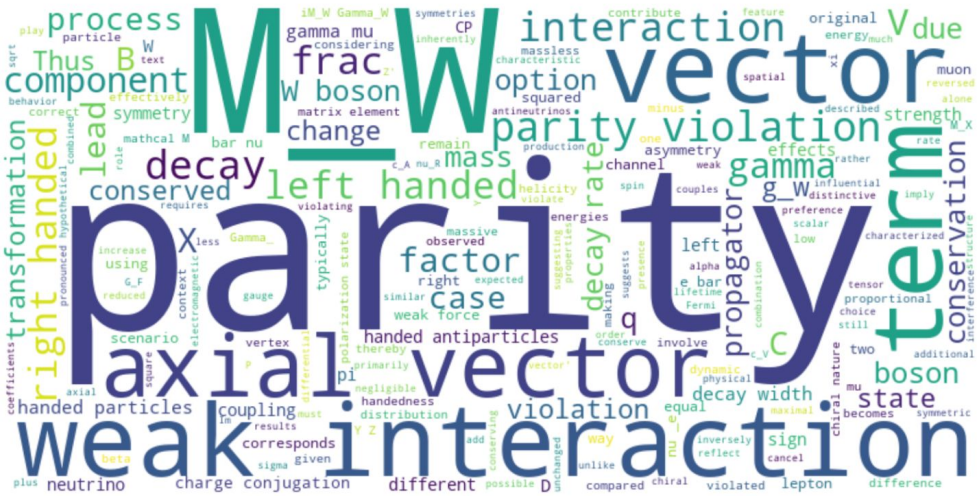
Fine-tuning to classification task:
Swap model head and copy over the
weights from the pre-trained backbone



Large (Language) Models for Physics



First results



Key Statistics

Total Questions: 108

Overall Accuracy: 78.70%

Error Rate: 21.30%

Counterfactual Accuracy: 72.22%

Non-Counterfactual Accuracy: 85.19%

Contact

SCAN ME

eugene.shalugin@ru.nl

sascha.caron@ru.nl