Large Language Models eXplainable AI

Aaron J. Masino, PhD
Associate Professor, School of Computing

Outline

- Demystifying large language models (LLMs)
- Introduction to XAI

Demystifying large language models

What is a language model (LM)?

- Probability model that estimates the conditional probability of the next word given the preceding sequence of words
- Applications
  - Speech recognition
  - Translation
  - Language generation
  - Handwriting recognition
  - and many, many more

The best thing about AI is that it ___

Word	P(w|s)
learns	4.5%
predicts	3.5%
understands	3.2%
helps	3.1%
does	2.9%

How can we create LM?

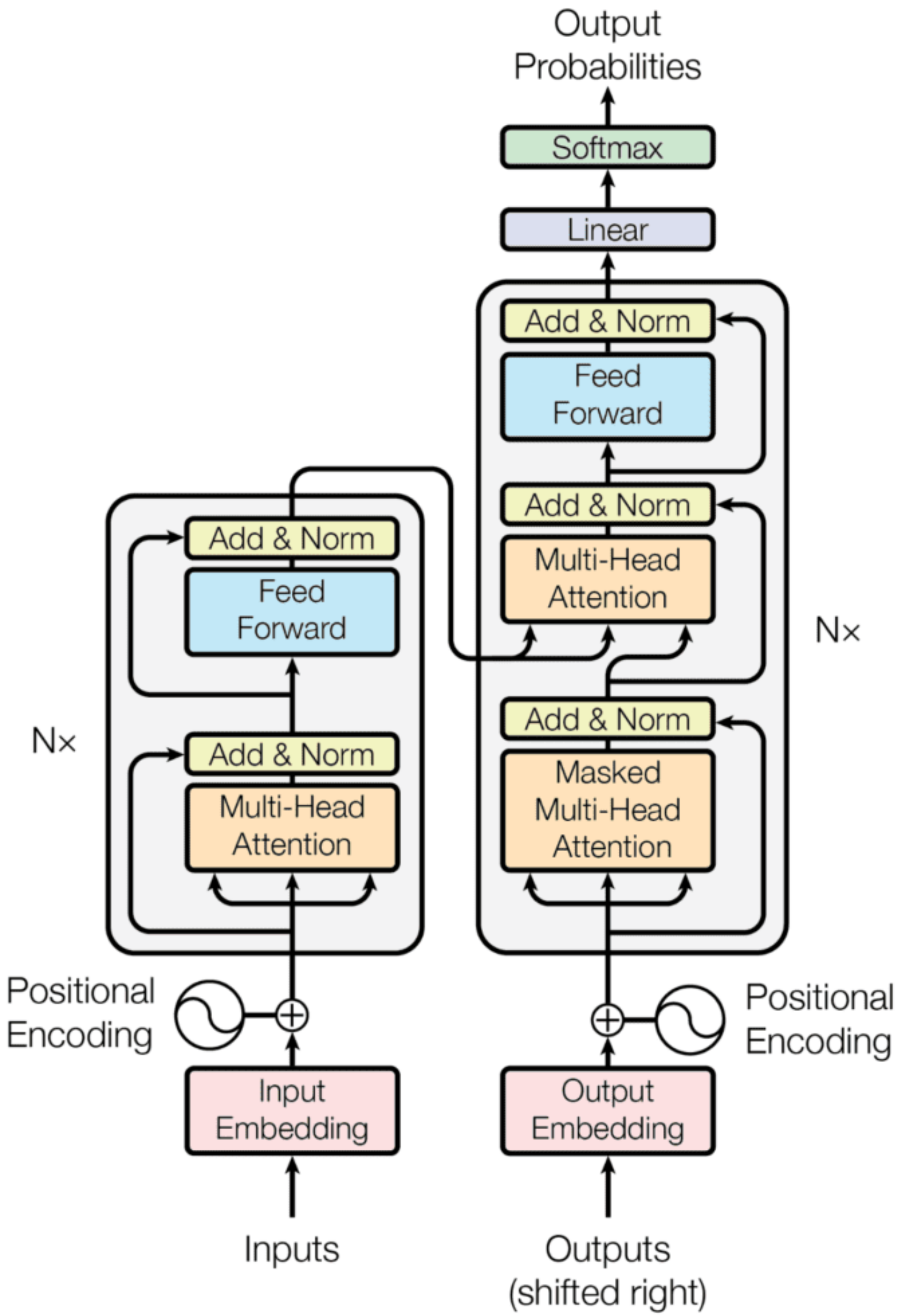
- Gather a large corpus of text documents
- Naive approach
  - Compute the empirical frequency of each word
  - Generate next word by probabilistic selection
  - Can we extend this to multi-word (n-gram) sequence empirical probabilities? Given ~40K common English words this yields:
    - 1.6 billion 2-word combinations
    - 60 trillion 3-word combinations
  - All of text in the world amounts to “only” O(100B) words
- What else can we do?

Self-supervised learning

- Gather a large corpus of text documents
- Train a deep learning model to approximate the language model, p(w|s)
- How can we get around labeling? Use "self" supervised learning where the model is trained to predict the next word given the preceding words

The best thing about AI is that it ___

Word	P(w|s)
learns	4.5%
predicts	3.5%
understands	3.2%
helps	3.1%
does	2.9%



Transformers are transformative

- Transformers have been instrumental in the development of modern large language models
- Overcome many of the challenges associated with RNN language models (notably long sequences and gradient vanishing)
- Generative Pre-Trained Transformers (GPT) are stacked transformers (decoder only) trained on the next word prediction task

*Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

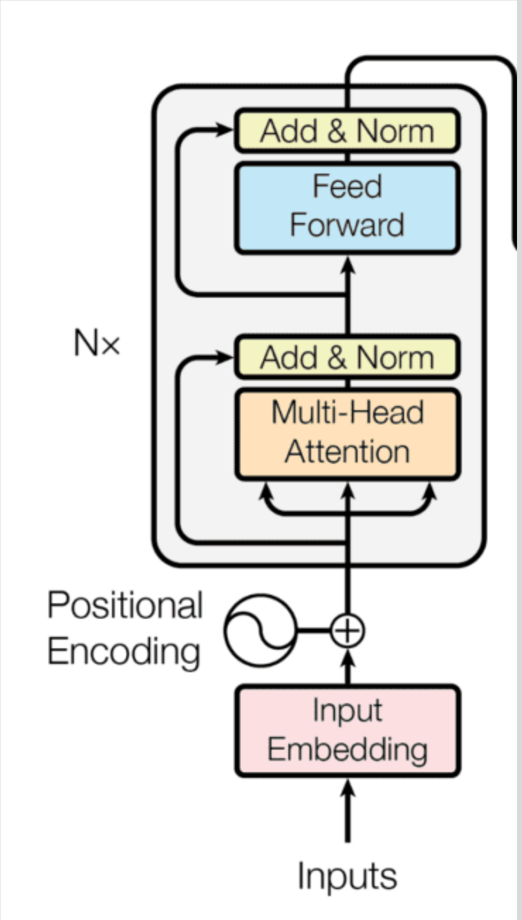
Transformer **Encoder** Pretraining

- Many language tasks require a pretrained encoder only (or just a library of embeddings generated by an encoder)
- Example pretraining approaches
  - **Masked Language Modeling** – predict hidden words

    The dog _____ down the street after the ____.
  - **Consecutive sentence prediction**

  True: *The weather forecast calls for rain. Bring an umbrella.*

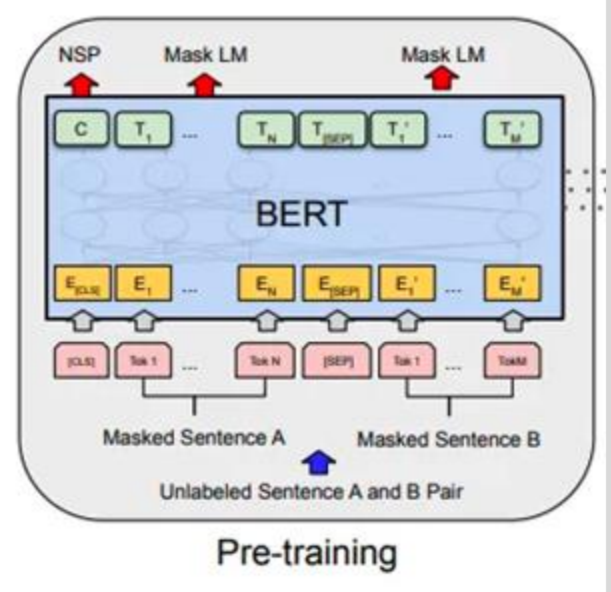
  False: *The weather forecast calls for rain. I work for Clemson.*

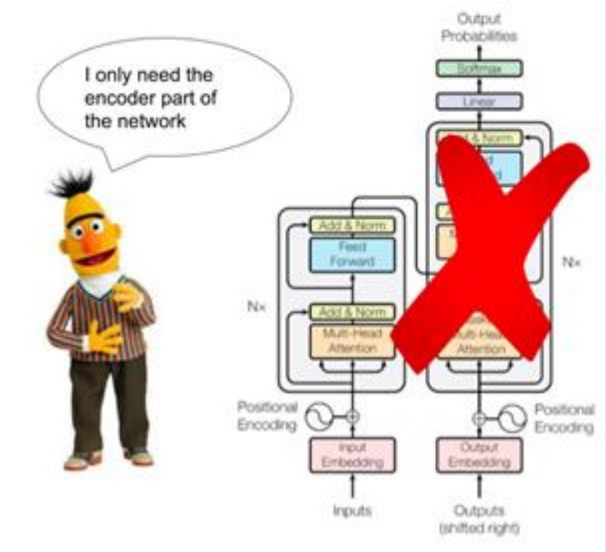


*Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Bidirectional Transformers for Language Understanding

- Introduced in 2019 by Devlin et al.
- Applies masked language model task with sentence pairs for encoder pre-training
- Approach originally used in chatGPT to form word embedding library (more on this later)
- Pretrained encoder can be paired with downstream network for ”encoder” only tasks:
  - Document classification
  - Sentiment analysis
  - Named entity recognition



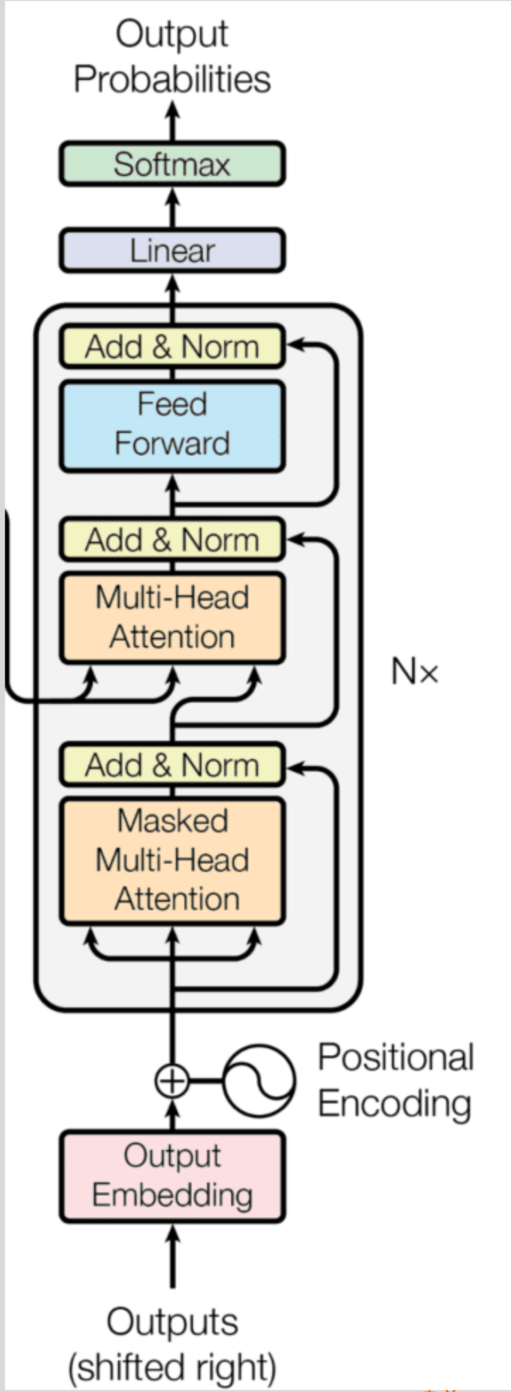


Transformer Decoder Pretraining

- Some language tasks require a pretrained decoder only
- Next word (or token) prediction

  The dog ran down the street after the ____

- This is the foundation of *Generative Pre-Trained* (GPT) models
  - Tokens of current output are represented using an embedding library (previously generated, e.g. with BERT)
  - The decoder generates the probabilities for the next output token using only the previous outputs (no encoder)



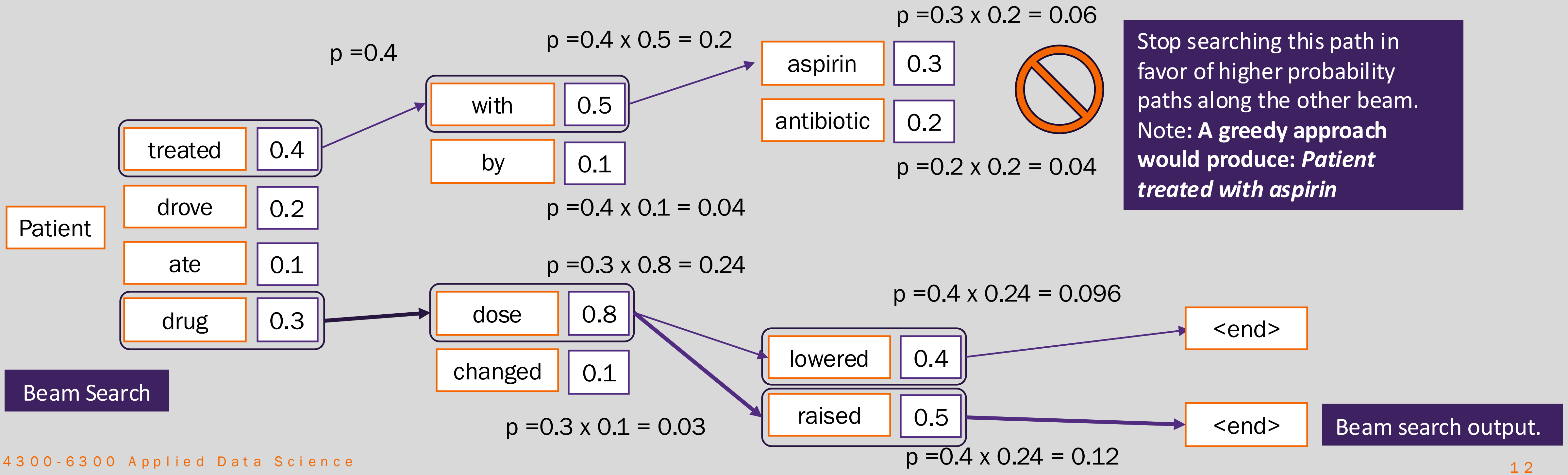
*Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

What about ChatGPT (and similar)?

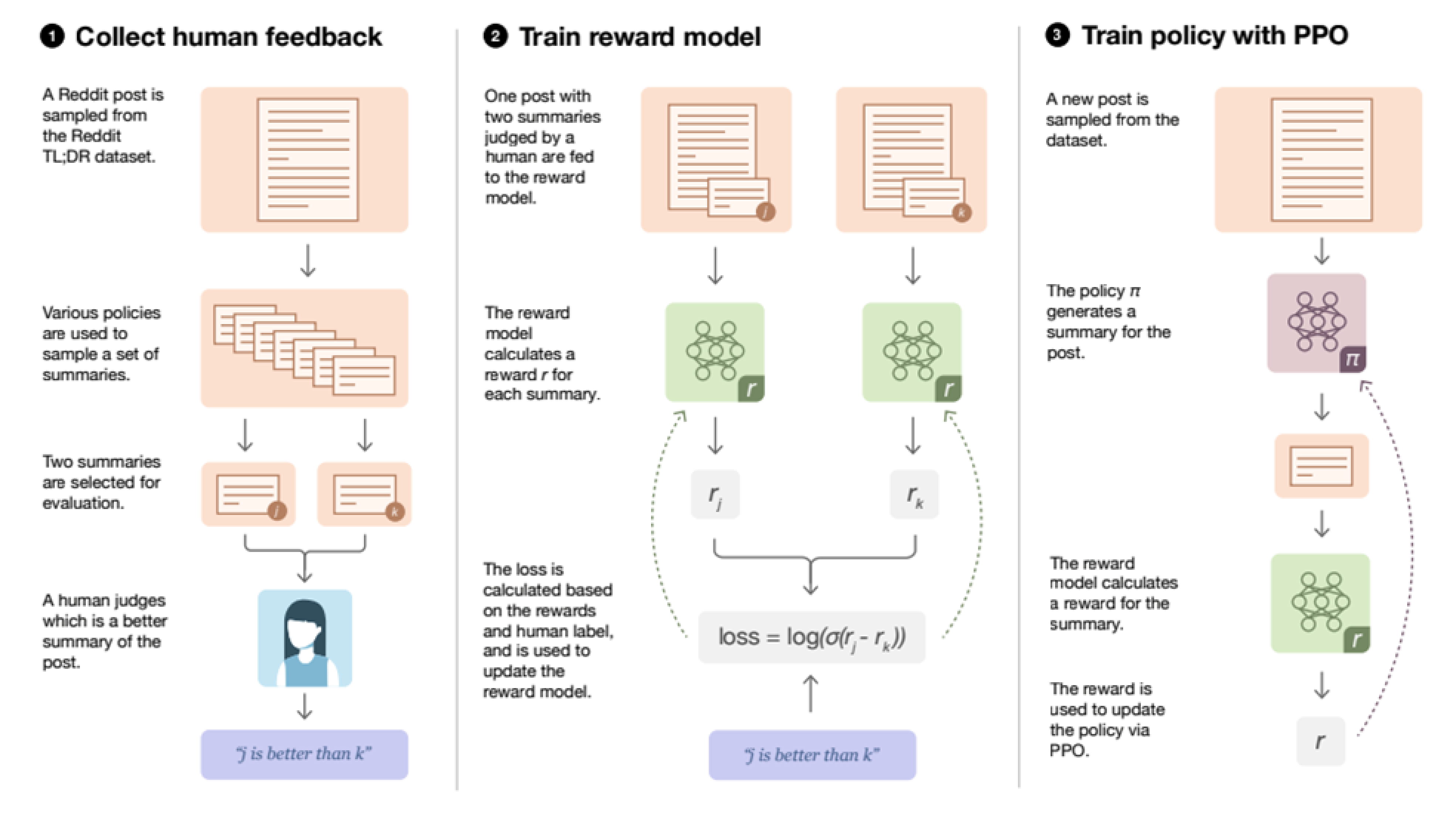
- Built on a generative pretrained (GPT) model
  - Composed of GPT layers
  - Trained to predict next word in a sequence
  - Huge amounts of data (GPT-3 was pretrained on >45 TB of data)
- Is that all?
  - No, add in some reinforcement learning (which is critical):
    - Humans rate “chat” results
    - Another NNet is trained to predict the rating
    - This NNet is then used as a loss function to further train the language generation model
  - Prompting strategies: Retrieval Augmented Generation, Chain of Thought, Ensemble Refinement, …

Language generation process, or “this is what ChatGPT does”

- Assume we have a trained LM (e.g., GPT)
- Given the current input, how do we select the next word?
  - Greedy – select the word with highest probability
  - Beam search – iteratively select the top K words and select the most probable path
- Both can be modified to select the next word / path probabilistically



OpenAI - Reinforcement Learning with Human Feedback



https://github.com/CarperAI/trlx

Text Generation Parameters

Temperature: This controls the randomness of the model's output. A higher value (closer to 1) makes the output more random, while a lower value (closer to 0) makes it more deterministic and focused on the most likely completion.

Maximum length: This specifies the maximum number of tokens (words or parts of words) that the model can generate for a single response

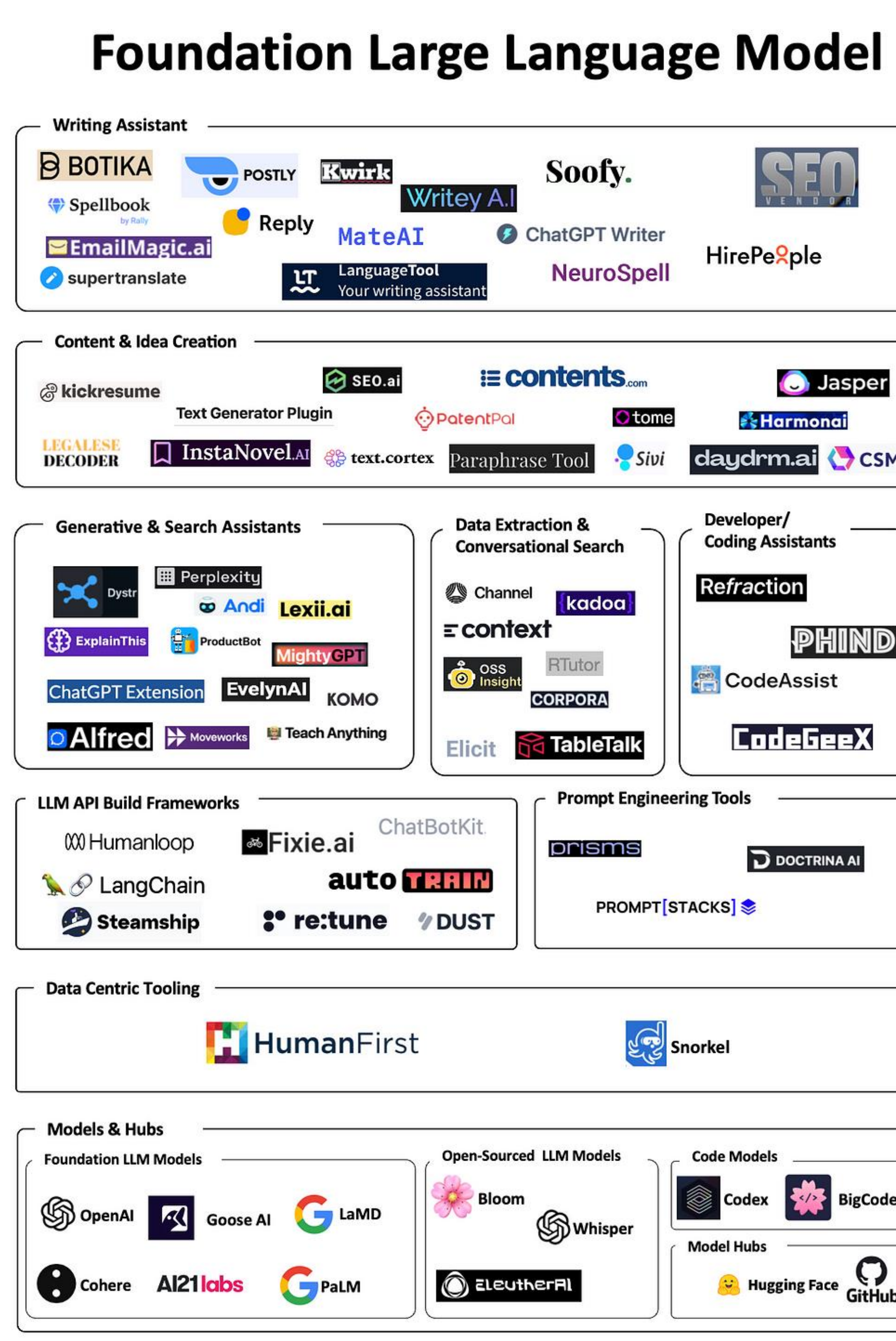
Stop sequences: These are specific strings of text that, if generated by the model, will cause it to stop generating further text.

Top P: This is a parameter for nucleus sampling, a method of generating text that involves choosing the next word from a subset of the vocabulary (the "nucleus") that has a cumulative probability larger than some value p. A higher value of p (up to 1) includes more of the vocabulary in the nucleus, making the output more random.

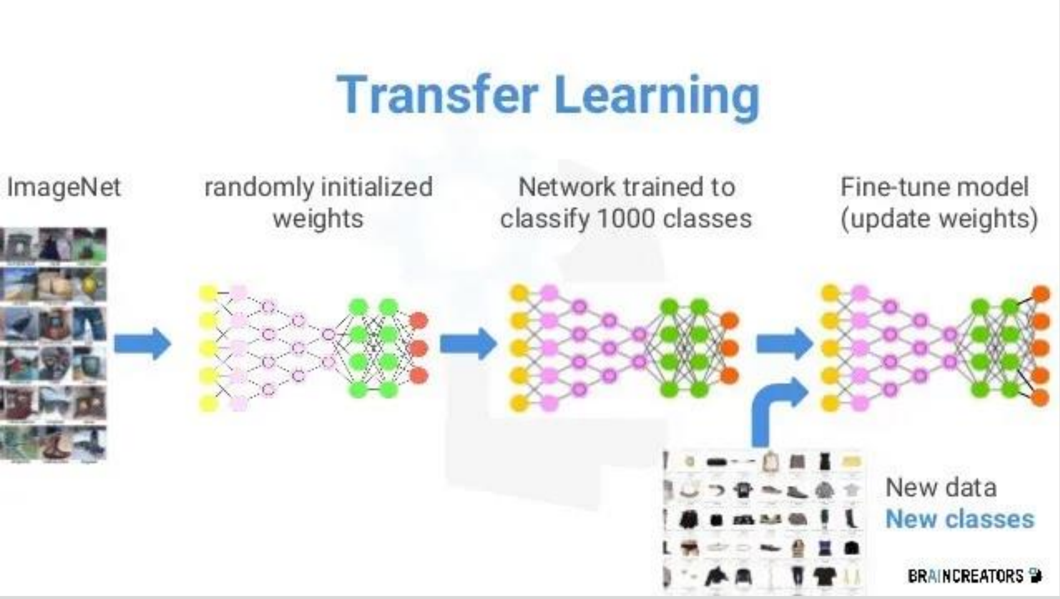
Frequency penalty: This value determines how the model handles frequently used words. A higher frequency penalty makes the model less likely to use common words and phrases.

Free and Open LLMs - Foundational

- Open Foundational
  - LLaMA (1 & 2) (Meta)
  - BLOOM (BigScience)
  - GLM (General Language Model)
  - GPT-J
  - GPT-NeoX
  - Pythia
  - StabilityLM
  - Cerebras-GPT (Cerebras)
  - Polyglot
  - RWKV
  - Falcon
  - OpenLLaMa
  - more …



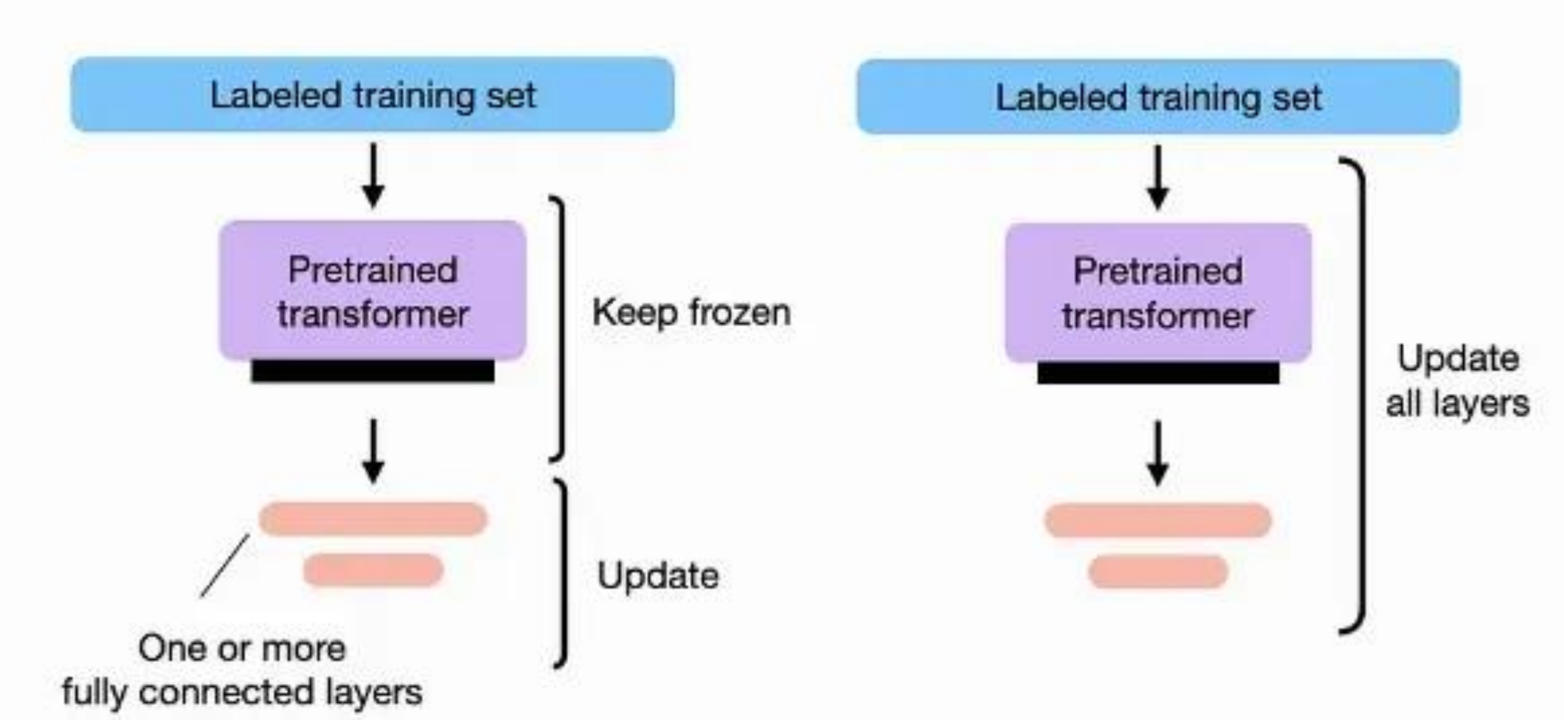
Transfer learning



1. Model is trained on a *generic* task for which a large dataset is available
2. The trained model is used as a starting point for a new, more specific task
3. **Fine tuning** – the original model is further trained on data for the new task
4. **Frozen encoder** – often, the pretrained model acts as the encoder and is not updated. Only the new decoder weights are updated for the new task

LLM Parameter Efficient Finetuning (PEFT)

- Full finetuning is time-consuming and expensive due to the computation and memory requirements.
- In the PEFT, only train a small subset of parameters while keeping most of the LLM model's weights frozen.



https://medium.com/intro-to-artificial-intelligence/parameter-efficient-finetuning-peft-of-llm-710831c0ffb3

eXplainable AI (XAI)

What is explainability?

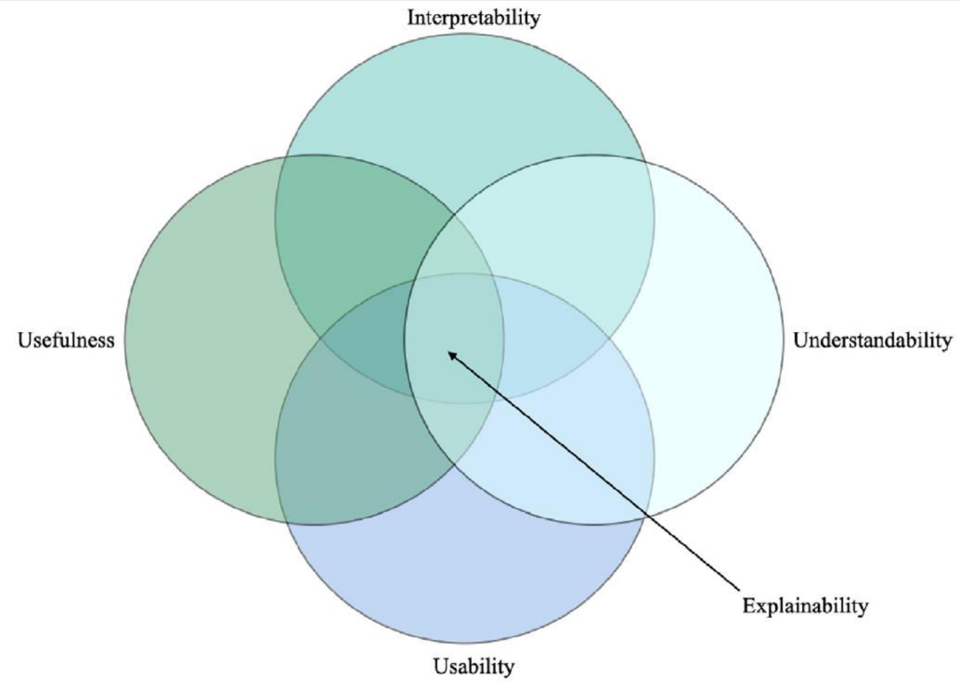
- *“… the ability to explain or to present in understandable terms to a human”* (Doshi-Velez & Kim 2017)
- *“descriptions are diverse … refers to more than one concept”* (Lipton 2016)
- There is no single definition
- Attempts to quantify explainability require subjective decisions on desiderata and measurement

What is eXplainable Artificial Intelligence (XAI)

- An emerging research area in AI
- Goal is to develop knowledge and methods that can be used to "explain":
  - How AI systems arrive at specific solutions
  - Which features most influenced a given output
  - Which and how features interact
  - Operating characteristics
  - Model biases
  - Accuracy relative to input regions
  - How the model learned

School of COMPUTING Clemson University

What makes an AI explainable?



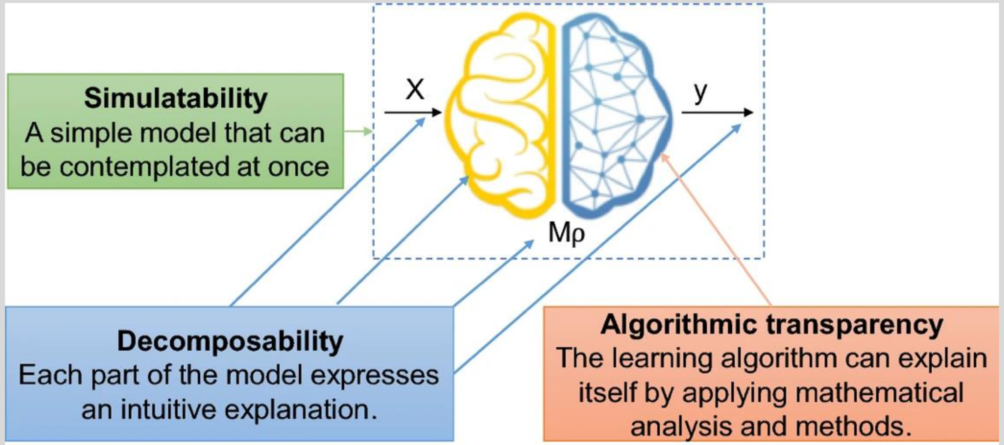
No consensus definition but most researchers agree explainability is a construct encompassing several concepts including:

- *Interpretability* – degree to which a human can infer the cause of and predict a model output
- *Understandability* – degree to which a user can determine how the system works
- *Usability* – the ease with which a user can operate the AI
- *Utility* – the practical usefulness of the AI

Figure credit: Combi, Carlo, et al. "A manifesto on explainability for artificial intelligence in medicine." Artificial Intelligence in Medicine 133 (2022): 102423.

What makes an AI understandable?

- No consensus definition
- Intrinsically linked to human cognition
- *Lipton proposed a framework for assessing model understandability composed of:



  - Simulatability – the degree to which a human can take the input and model parameters and step through every calculation required to produce a prediction
  - Decomposability – the degree to which each part of the model (inputs, parameters, calculations) allow for an explanation
  - Algorithmic transparency – the degree to which theoretical guarantees for learning algorithm properties (e.g., convergence) can be provided

*Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." Queue 16.3 (2018): 31-57.
Figure credit: Minh, Dang, et al. "Explainable artificial intelligence: a comprehensive review." Artificial Intelligence Review (2022): 1-66.

Why do we need explainability?

Need for explainability arises when formal learning objectives fail to capture other real-world desiderata

Prediction Loss

$$L(y, \hat{y}) + G(\lambda, \widehat{\boldsymbol{\Theta}}) + ?$$

Regularization

Objective Function Limitations

- Difficult to formalize many real-world desiderata in mathematical form
- For example, how would one form an objective to encode ”trust” or “fairness”

AI desiderata beyond prediction accuracy

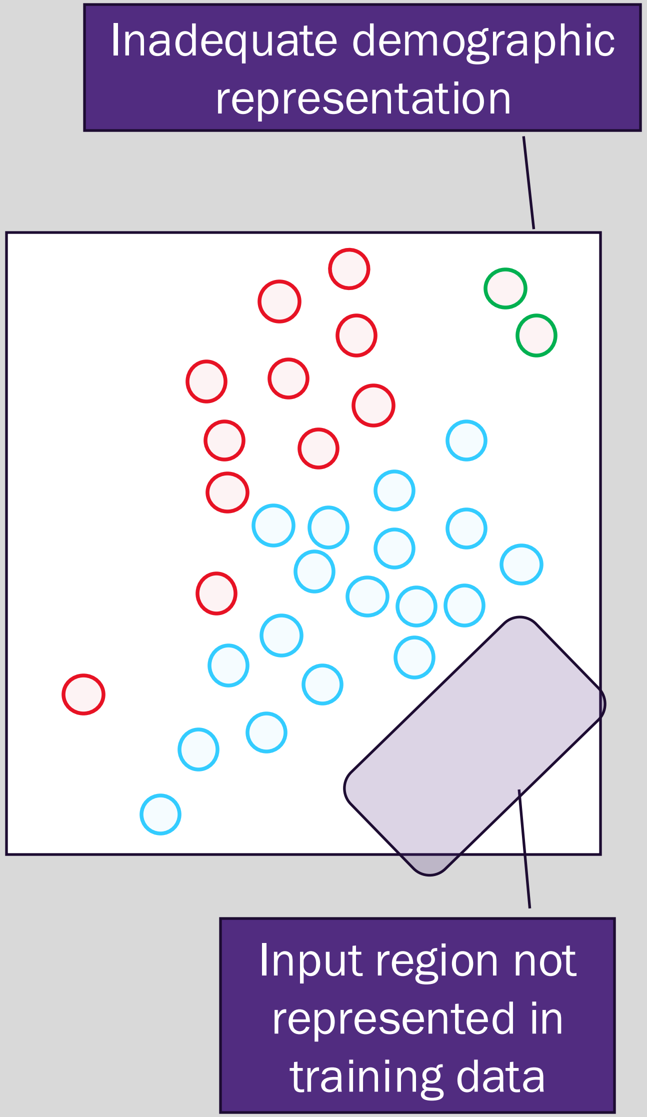
- Trust – relates to subjective human emotion that the model “is fair” or “can be left in control”
- Scientific knowledge
- Decision support information
- Fair, ethical, legal decisions
- Safety
- Privacy

Explainability through data characterization

- Examine standard statistical measures (mean, stdev, range) of input features and target (supervised learning problems) data
    - Are there outliers? Are outliers due to measurement error or are they real?
- Examine missing data rates for input features
    - Imputation may be used in some cases should be restricted to cases where missing data rate is below some threshold
    - Rule of thumb: Require missing data rate <30% for imputation
- Feature reduction:
    - Are there redundant features that can be removed (e.g., highly correlated features)?

Data quality and explainability

- Training data coverage
  - Range/counts of continuous/categorical valued input features
  - Sample input and output density (i.e., assess sparsity)
- Demographic representation
  - Which and to what extent are population groups represented in the data?
  - Medical data: should report comorbidities, procedures, medications
- Data collection methodology
  - How was the data collected?
  - What inclusion/exclusion criteria were used?



"Inherently" explainable models

- Some models are often cited as *inherently explainable* including:
  - Linear / logistic regression
  - Decision trees
  - Rule sets
- Why?
  - Relation between variables is usually clear
  - Direction of relation between input and output is usually discernable
  - Computation of output is usually straightforward

$$y = \sigma[\beta_1 x_1 - \beta_2 x_2]$$

Human simulatable!

In a linear model with a small number of predictors, it's easy to "simulate" the result (here two multiplications and a sum >0 results in positive class) and to "infer" relations (e.g., as x_2 increases the likelihood of the positive class decreases)

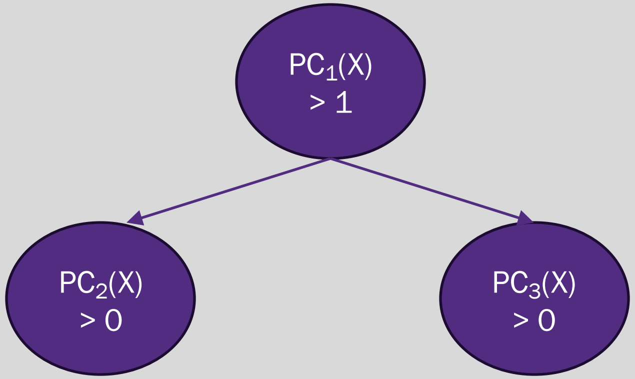
Are “Inherently” explainable models always understandable?

- What if there are many variables?

$$y = \sum_{i=1}^{10,000} \beta_i x_i$$

Not human simulatable!

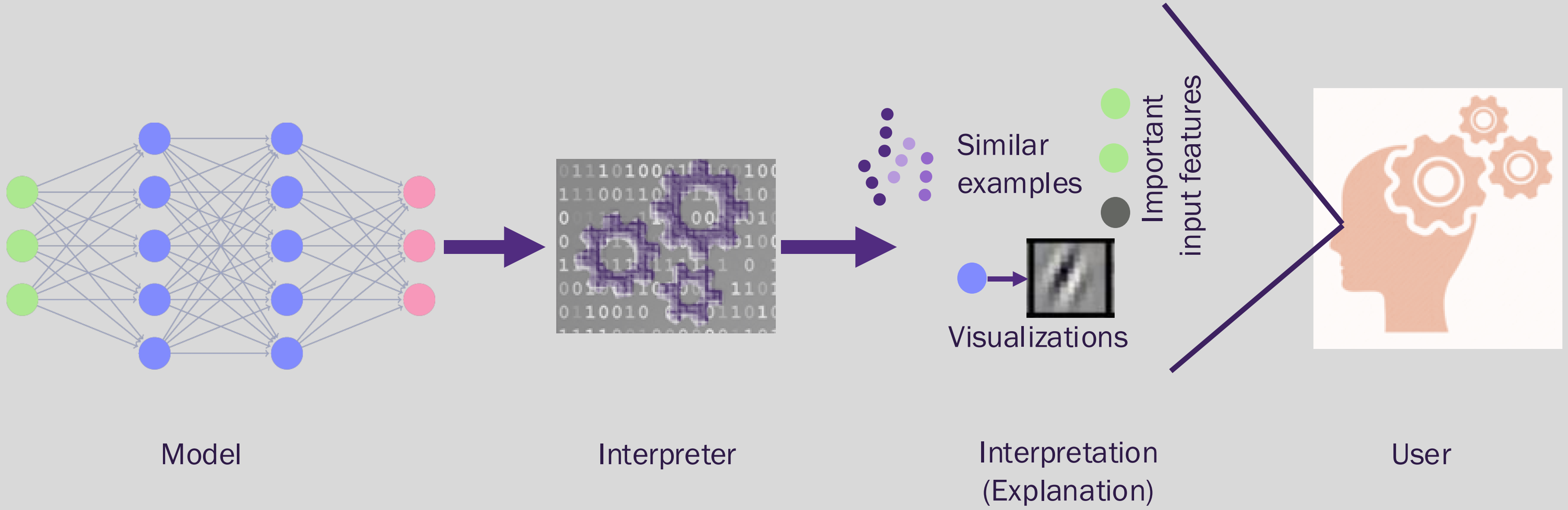
- What if the input variables are highly engineered? For example, the principal components as input to a shallow decision tree?



Not human simulatable!

Post-hoc explanation method

Methods applied to models and their outputs to create explanatory information

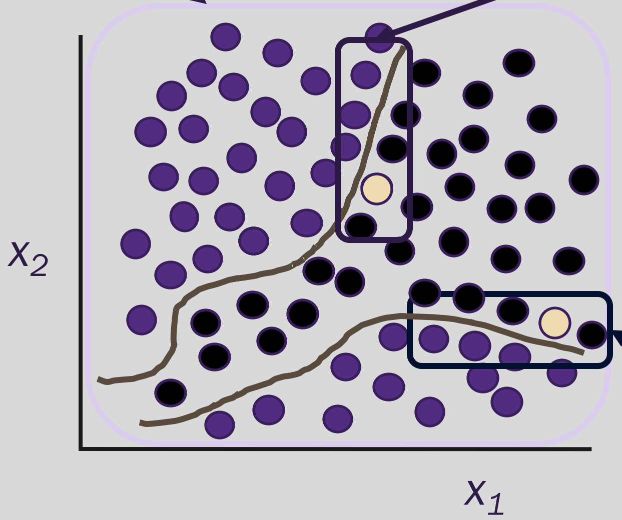


Global vs. local feature importance

Global explanation: x_1 and x_2 equally important

Local explanation: x_1 is more important here …

Global explanation: **independent** of input sample
Local explanation: **depends** on input sample



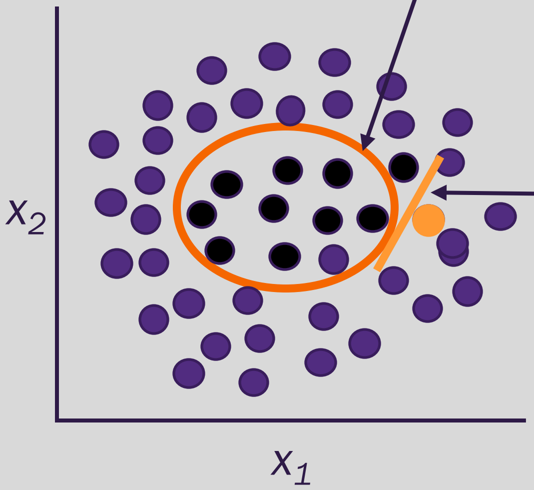
Local explanation: x_2 is more important here

Global feature importance assessment

- Permutation analysis
  - Only applicable to structured data
  - Method
    1. Train model as usual
    2. Shuffle the observed values among samples for one feature
    3. Measure performance
    4. Repeat steps 2-3 for each column
    5. Rank features by model performance (shuffled features that lead to higher reductions in model performance or more important)
- Ablation analysis
  1. Remove one feature
  2. Train model and measure performance performance
  3. Repeat steps 1-2 for each feature
  4. Rank features with same approach as permutation analysis

Local feature importance via additive feature attribution

Global model: $f(x_1, x_2) = class$



Local explanation model

$$g(x_1, x_2) = \phi_0 + \boxed{\phi_1} x_1 + \boxed{\phi_2} x_2$$

x_1 , x_2
importance

SHapley Additive exPlanation (SHAP)

For a model $f(x) = y$, form a linear approximation:

$$g(z) = \underbrace{\boxed{\phi_0}}_{\text{Baseline risk}} + \sum_{i=1}^{M} \underbrace{\boxed{\phi_i}}_{\text{Feature importance}} z_i \qquad \underbrace{\boxed{z = h_x^{-1}(x)}}_{\text{Binary feature mapping}}$$

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \underbrace{\boxed{\frac{|z'|!\,(M - |z'| - 1)!}{M!}}}_{\text{Weighting term}} \underbrace{\boxed{\left[f[h_x(z')] - f[h_x(z' \backslash i)]\right]}}_{\text{Output difference when removing feature } i}$$

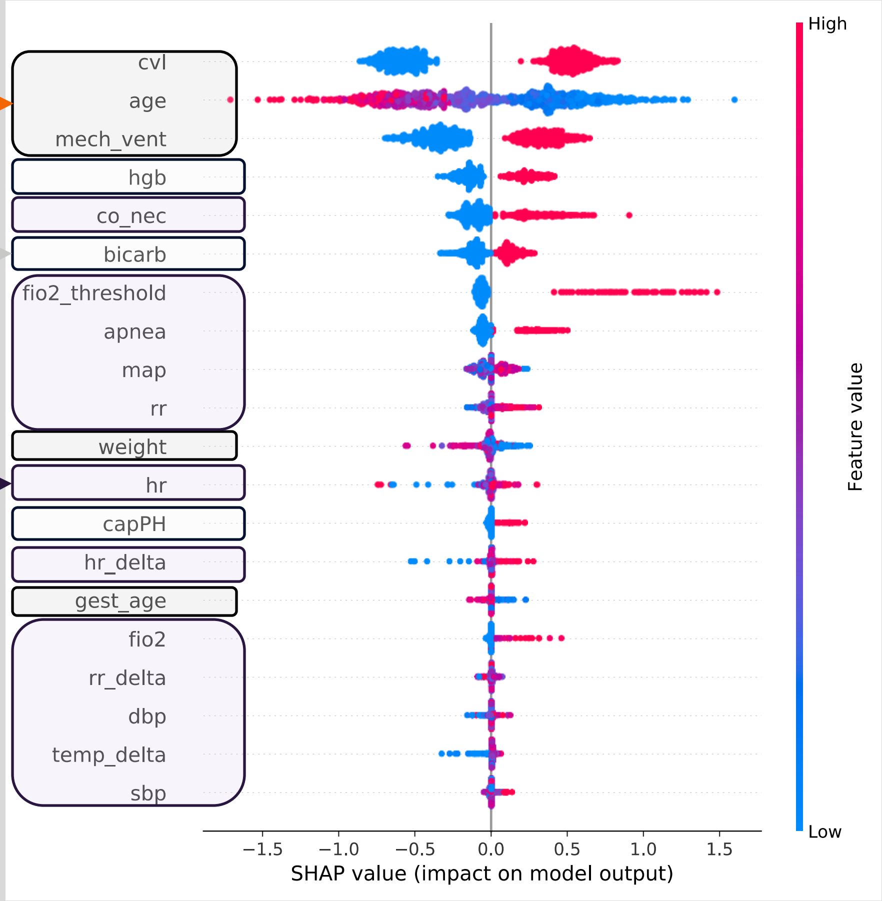
A Unified Approach to Interpreting Model Predictions. Lundberg & Lee. 2017.

SHAP Visualization

Known risk factors

Lab indicators

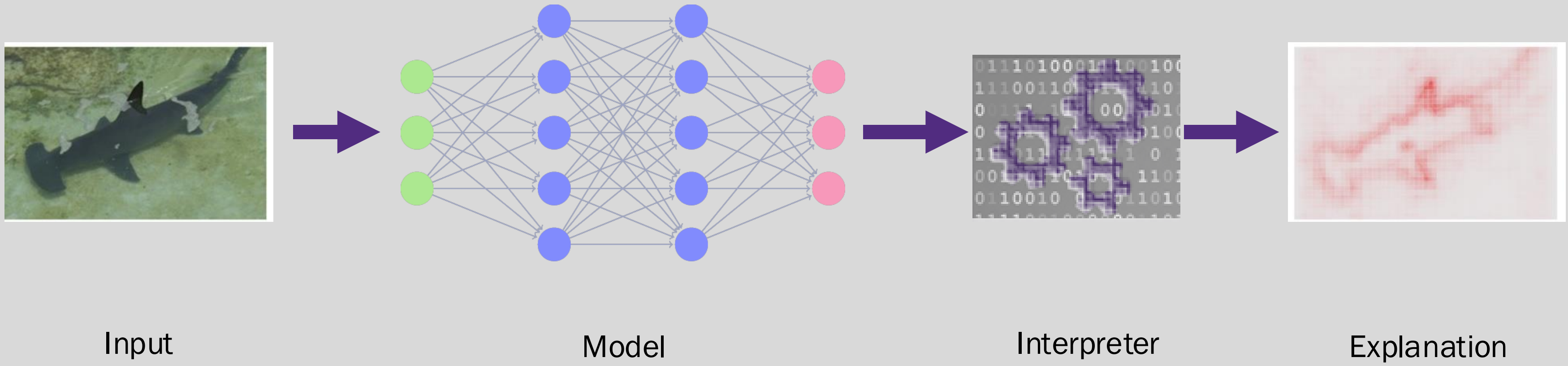
Clinical status change?



Visualization local feature importance methods

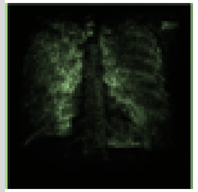
For a model $f: R^d \rightarrow R^+$, associate with each input element a relevance score, $R_p(x)$, that indicates importance

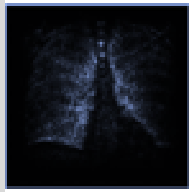
- Gradient methods: $\sum_p \left({}^{\partial f}/_{\delta x_p} \right)^2 = \|\nabla_x f(x)\|^2$
- Perturbation: Mask image areas
- Decomposition methods: $f(x) = \sum_p R_p(x)$



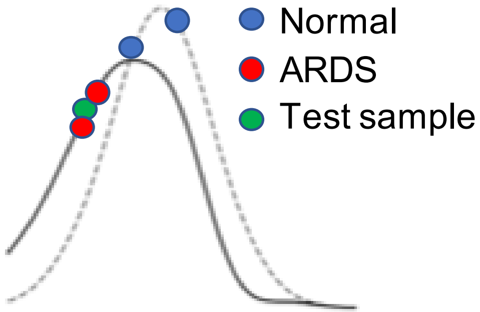
Instance based explanations with influence functions

- Robust statistical methods that identify the training example(s) that most influence the current prediction
  - Deletion diagnostics (linear models)
  - Influence functions (twice differentiable loss function)
- Useful in model development as a means of identifying outliers with undue influence on predictions
- For deployed models can be used to indicate how important features of test sample compare to influential samples





SpO2/Fi02 Data Distribution



Report Findings Comparison

Test Sample

Opacification (Right)

Pleural Apical Thickening

Influential ARDS Training Samples

Opacification

Influential Normal Training Samples

Opacification (negated)

Python XAI resources

- SHAP - https://shap.readthedocs.io/en/latest/
- ELI5 - https://eli5.readthedocs.io/en/latest/
- Captum - https://captum.ai/ (deep learning XAI library)
- Interpretable Machine Learning by Molnar (free online text) https://christophm.github.io/interpretable-ml-book/