Explainability for LLMs

Benjamin Roth, Loris Schönegger

Digital Philology

University of Vienna

Large Language Models

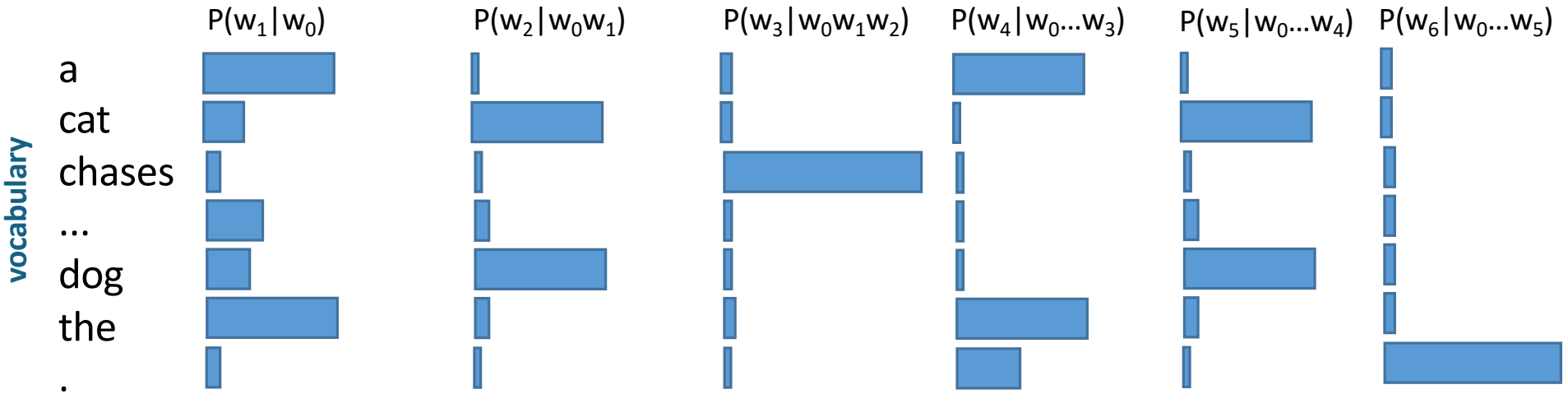
What is a large language model? (LLM)

- Artificial neural network that **predicts text** that fits well for a given **context** (typically also text)
  - Predict one **word** that is highly likely given a **prompt** (previous words)
  - For predicting an entire text, repeat the process (i.e., extend the prompt with previously predicted words)
  - To predict a text from scratch, use an extra symbol <START> as the initial prompt

- Modern LMs use enormous text collections to **learn** to predict the next word given previous words

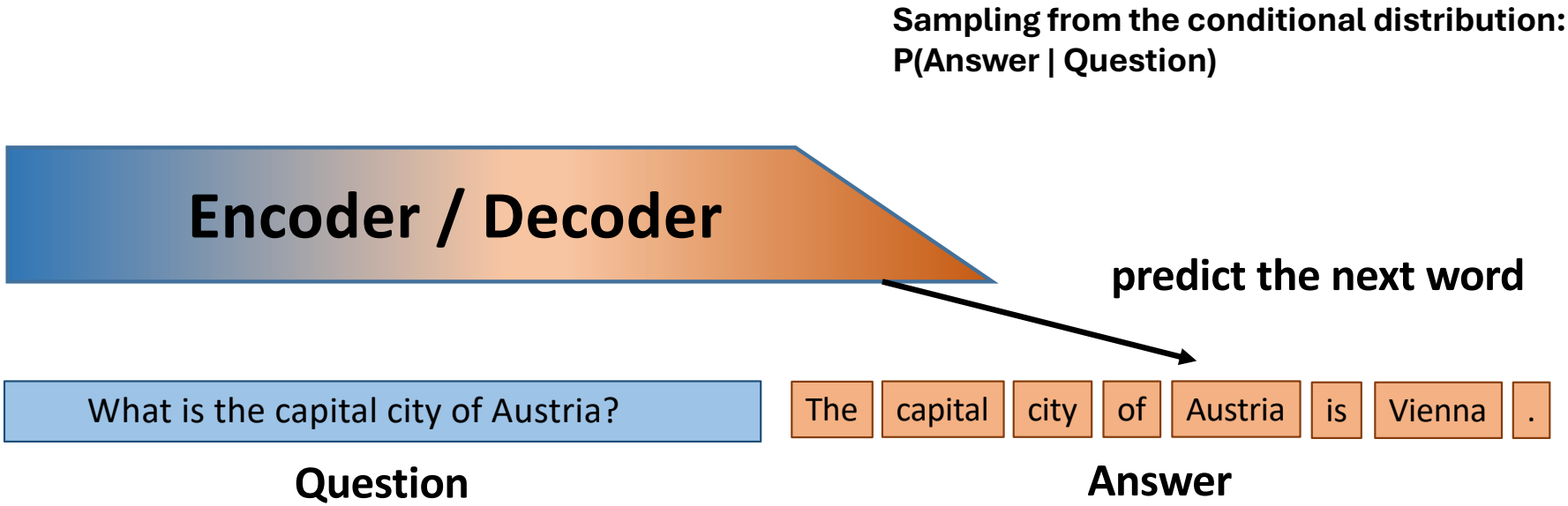
What is a large language model? (LLM)

- **Classifier to predict the next word from context**
- **Trained on massive amounts of text**
- **+ examples how to reply to instructions ("prompts")**



<START>	the	dog	chases	a	cat	.
w_0	w_1	w_2	w_3	w_4	w_5	w_6

LLMs as Encoder/Decoder model

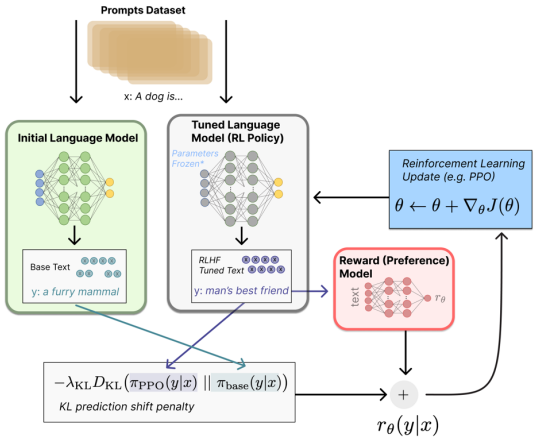


[Vaswani 2017, Radford et al, 2018 , Raffel et al 2019, Wei et al 2022, ...]

What is “Training” and “Learning”?

- LLMs have the capacity to recognize **patterns**:
  - groups of **similar words**
  - groups of **similar** larger **language structures** (formulations, …)
  - similarities are expressed by **vector representations**
- LLMs can also capture **regularities** of how patterns **combine** and interact
- **Parameters** determine which words and structures are more similar, and how their combination influences next word prediction
- Training and learning:
  - The **parameters** of the network are **randomly initialized**
  - During training, the LLM is presented with a **context** and attempts **next word** prediction
  - **Parameters are changed** so that the probability for the correct (observed) next word is increased (backpropagation, stochastic gradient descent)
  - This process is repeated for billions of context – next word pairs (**training examples**)

“Instructions” and ChatGPT



source: https://huggingface.co/blog/rlhf

- Newer model [Wei 2021/FLAN, Ouyang 2022/InstructGPT,ChatGPT] are optimized to generate answers for **instructions**
- LLM parameters are optimized in three **rounds of training** to solve the following tasks:
  1. Next word prediction (> 3B Token)
  2. Learn from example instructions with given answers (several 100K examples)
  3. Learn from additional human feedback w.r.t. desired properties of good answers (*helpful, harmless, honest* [Bai 2022/Constitutional AI, RLHF])

Auto-regressive vs. masked language models

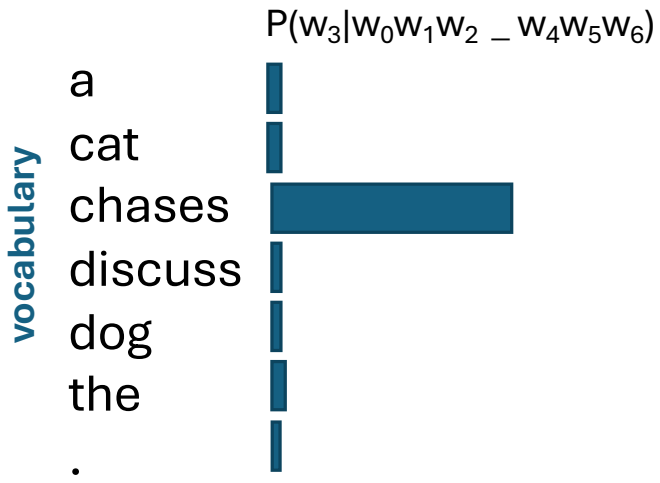
Auto-regressive language models

- The type of LM that we have discussed so far is called **auto-regressive** language model (ARLM)
- It predicts text left-to-right (the context is the prompt and what has been generated so far)
- It is used for **generating** text (**not analyzing** it)

Masked language models (MLM's)

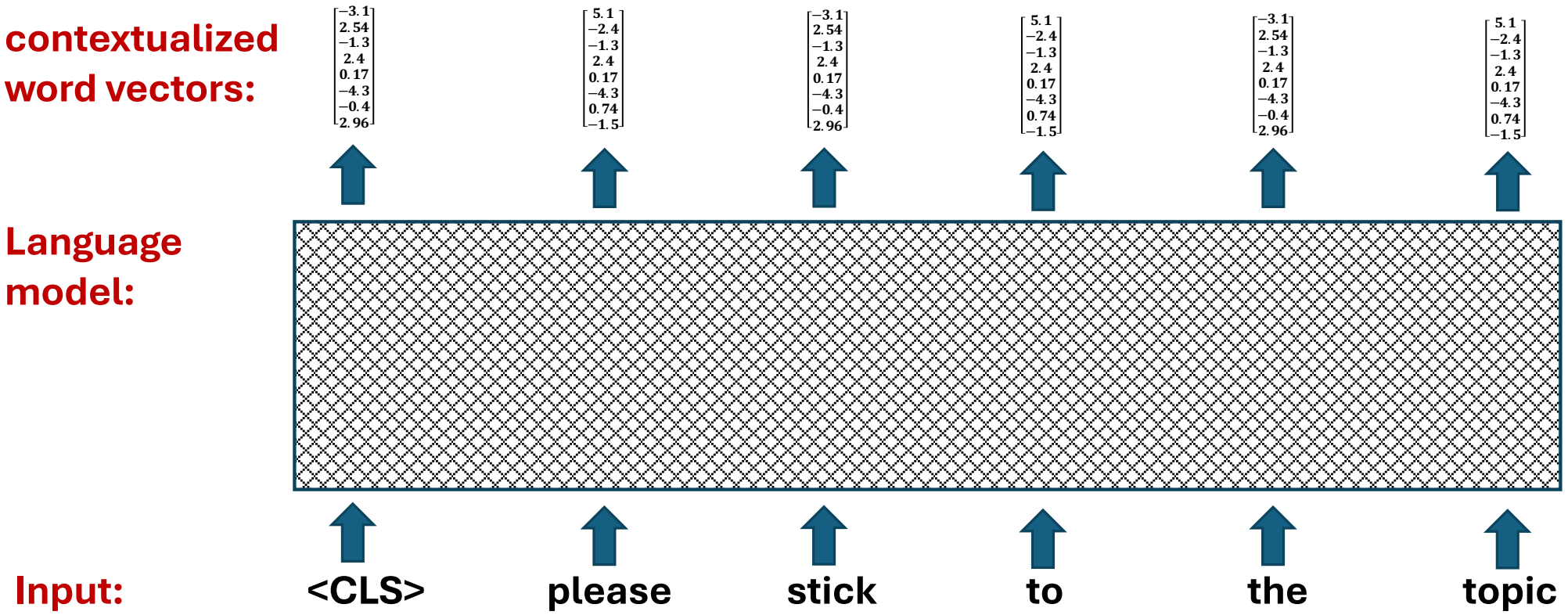
- We have seen auto-regressive LMs
  - **context:** previous words
  - **learned to predict:** next word
  - **for:** generating text
  - **typically:** large-scale, resource intensive
- Another type: masked LMs
  - **context:** surrounding words
  - **predict:** masked word / properties for words at all positions
  - **for:** analyzing and categorizing text
  - **typically:** more light-weight, but needs task-specific training data to be useful

MLM (toy example)



<START>	the	dog	<MASK>	a	cat	.
w_0	w_1	w_2		w_4	w_5	w_6

MLMs: Vector Representations



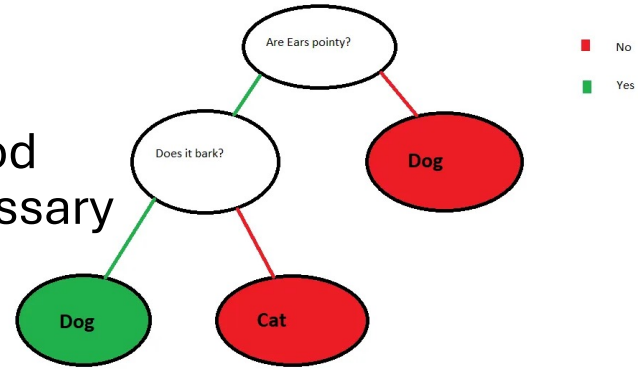
Auto-regressive and masked LM's

- ARLM's
  - Sometimes also called "Causal language models"
  - https://huggingface.co/transformers/summary.html#autoregressive-models
  - Original GPT; GPT-2; CTRL; Transformer-XL; Reformer; XLNet; ChatGPT; ...
  - ARLMs are better than MLMs for **generating** texts
- MLM's
  - Sometimes also called "Autoencoding models"
  - https://huggingface.co/transformers/summary.html#autoencoding-models
  - **BERT**; ALBERT; RoBERTa; DistilBERT; XLM; XLM-RoBERTa; FlauBERT; ELECTRA; Longformer
  - The advantage of MLMs lies in learning **contextualized vector representations**

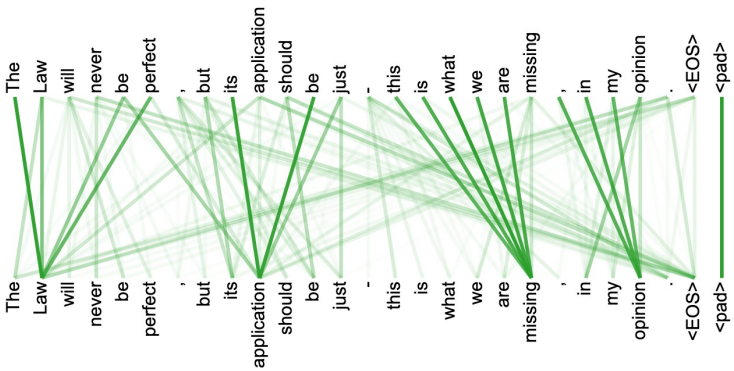
Explainability for LLMs

Interpretability vs. Explainability

- **Interpretable** machine learning models:
  - The model itself is transparent and can be understood (interpreted) by humans, no extra explanations necessary
  - Interpretability is **a property of the model**
  - Often:
    High interpretability ← Limited model complexity → limited performance
  - Examples: decisions trees with limited depth



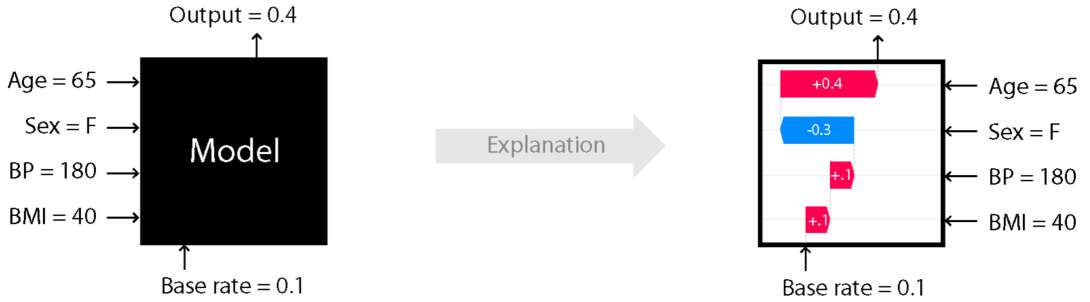
- Language models by themselves are not interpretable, but may have some interpretable parts,
  e.g. attention patterns



Interpretability vs. Explainability

- **Explanations** for machine learning models:
  - The most important factors of a (potentially very complex) model or model decision are identified and presented to a user
  - Explanations are usually **not directly available**, they need to be calculated separately
  - Challenges:
    - What are the **most important** factors …
    - … that are at the same time **interpretable** by humans?
  - Examples: Feature importance algorithms such as SHAP

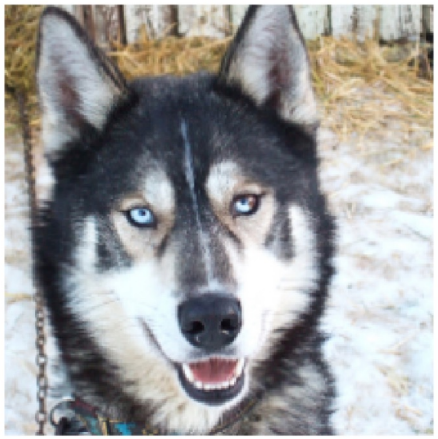
https://shap.readthedocs.io/en/latest/_images/shap_header.png



Transparent and explainable predictions

- Why is a husky classified as a wolf?
  (LIME [Ribeiro 2016])
- Why is a social media post classified as hate speech?
  (Hatecheck [Röttger 2021])
- Why is a loan approved or rejected?

→ Which explanations methods are reliable? [Poerner 2018, Sydorova 2019]



(a) Husky classified as wolf

(b) Explanation

- **Right to explanation** (EU GDPR Recital 71):
  - ``[safeguards include ...] the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision."

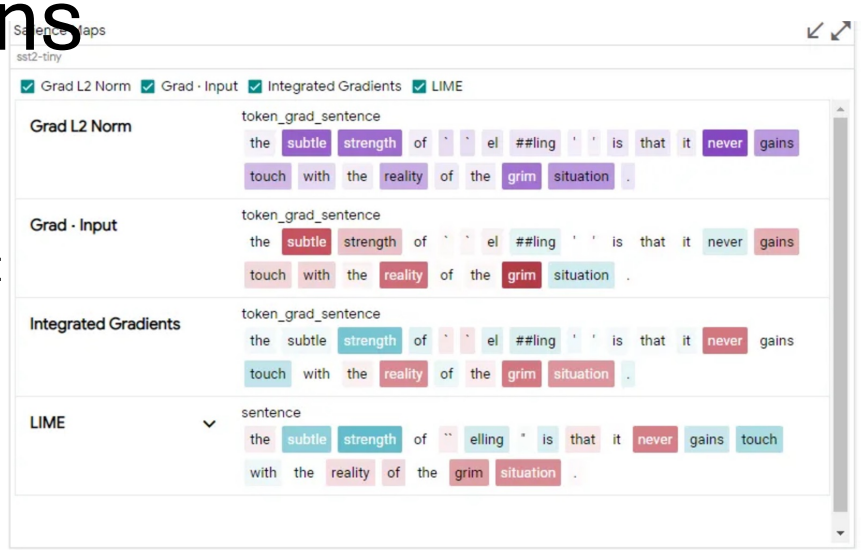
Types of Explanations

- Feature-based:
  - “Which **input features** (words …) have the most impact on the **output** of the model (classification, generated text)?”
- Example-based (Training-data-based):
  - “Which **training data** had the most influence (on a particalur output, or overall on the model)?”
- Mechanistic:
  - “How can the **causal dependencies** in the model be summarized?”
- Global vs. local explanations:
  - Explanations **specific** to one output are called **local**
  - Explanations **independent** of specific outputs are **global**

Which of the above explanations are global, which are local?

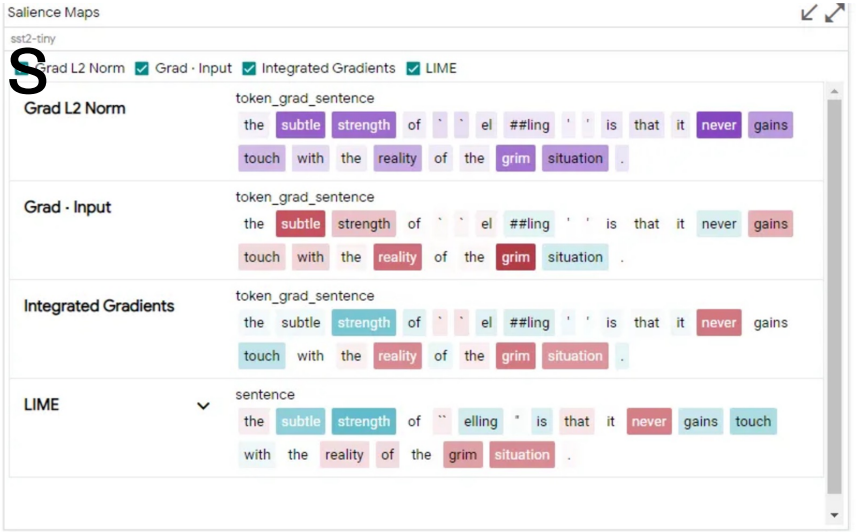
Feature-based Explanations

Gradient-based explanations



- Goal of feature-based explanations:
  - Determine influence of input parts on output (e.g. words on predicted sentiment: positive/negative)
- Gradient for model ***training***:
  - $\nabla_\theta \log P(y|x;\theta)$
  - *"How to change the **parameters** to **increase** the likelihood of **training output**"*
- Gradient for feature-based ***explanations?***
- $\nabla_x \log P(y|x;\theta)$
- *"How to change the **input** to **impact** the **likelihood** of **generated output**"*

Gradient-based explanations



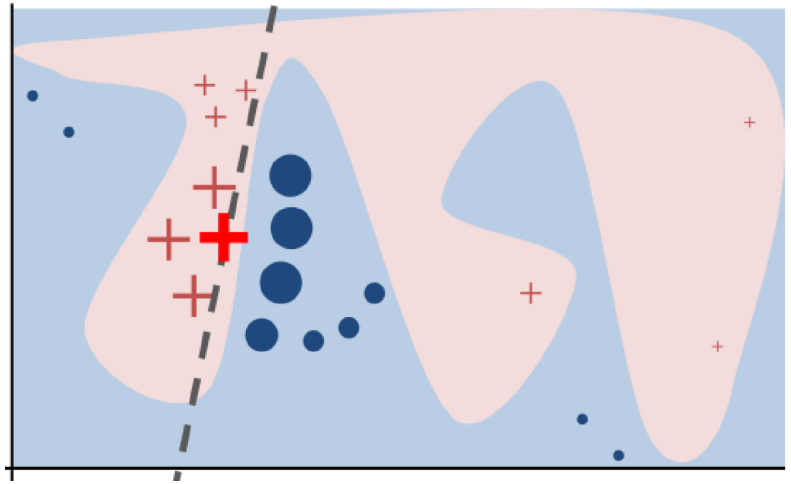
- *"How to change the **input** in order to **impact** the **likelihood** of the **generated output**"*
  - $\nabla_x \log P(y|x;\theta)$
- Advantages:
  - **Easy to compute.** Deep learning toolkits support automatic differentiation even for very complex models.
  - **Fast.** Just one backward pass. No training of an auxiliary model, no permutations of input, no sampling.
- Disadvantage:
  - The gradient only approximates impact of infinitesimally small changes to the input. (A little bit more or less of the word "grim".)
  - This is not how language works (words are added or removed as a whole).

LIME: Local Interpretable Model-Agnostic Explanations

- Models the effect of removing words from the input
- Neural networks can model **non-linear** interactions:
  - "This was **not** a movie I did **enjoy**."
  - Neither "not" nor "enjoy" are negative by themselves
- LIME approximates these non-linear interactions locally by a linear model

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro Sameer Singh Carlos Guestrin



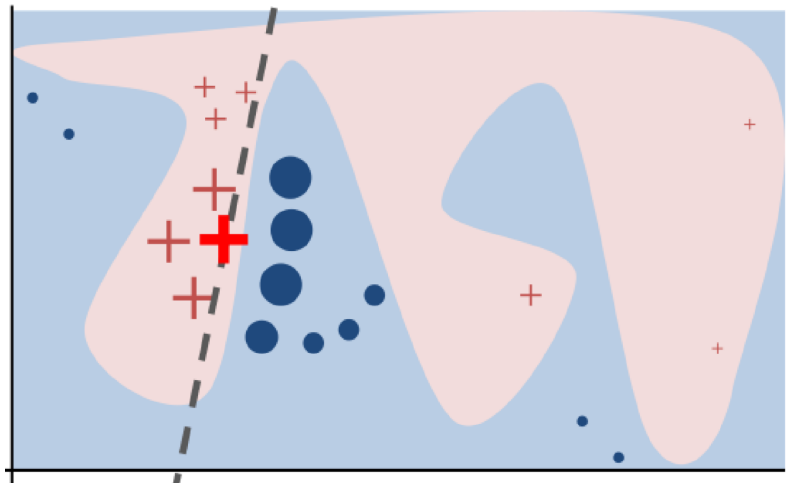
LIME: Local Interpretable Model-Agnostic Explanations

- LIME creates training data for a separate linear ***explanation model***…
- … by perturbing the input and observing the corresponding output of the model to be explained

Input	Output
This was not a movie I did enjoy [1 1 1 1 1 1 1 1]	NEG
This was a movie I did enjoy [1 1 0 1 1 1 1 1]	POS
This was enjoy [1 1 0 0 0 0 0 1]	POS
not a movie I enjoy [0 0 1 1 1 1 0 1]	NEG

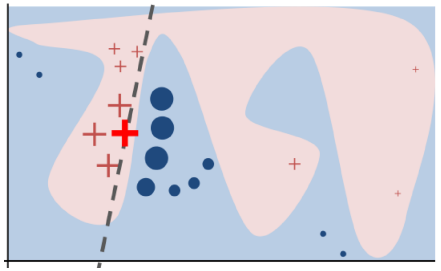
"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro Sameer Singh Carlos Guestrin



LIME Objective Function

(Notation of https://arxiv.org/abs/2403.14459)



Vector with explanation weights for each input feature

Weighting of the permuations

‚Scalarized' output

$$\xi = \arg\min_{w} \sum_{i=1}^{n} \pi\left(z^{(i)}\right)\left(w^T z^{(i)} - S(x^{(i)}; y^o, f)\right)^2 + \lambda R(w)$$

Regularizer

How closely the explanation approximates the prediction

$x^{(i)}$	$z^{(i)}$	y	S(...)
This was not a movie I did enjoy	[1 1 1 1 1 1 1 1]	NEG	0.12
This was enjoy	[1 1 0 0 0 0 0 1]	POS	0.85
...	...	...	...

LIME Details

$$\xi = \arg\min_{w} \sum_{i=1}^{n} \pi\left(z^{(i)}\right)\left(w^T z^{(i)} - S(x^{(i)}; y^o, f)\right)^2 + \lambda R(w)$$

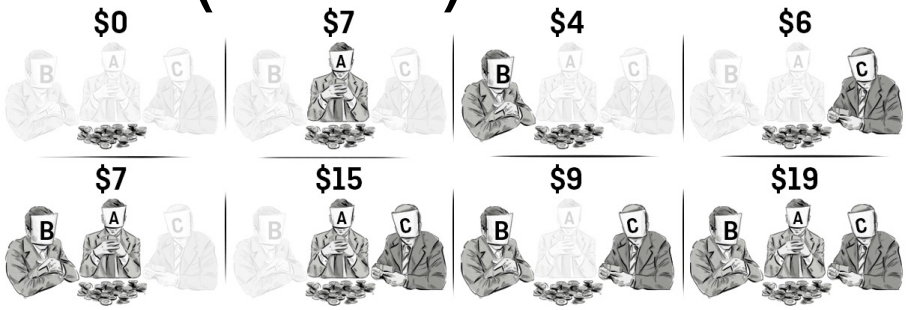
- The LIME objective function can be solved with least squares **(nice!)**
- In order to obtain good results, one often needs thousands of permutations per example **(not nice…)**
- How to choose the *kernel* weights π ?
- How to choose the regularizer R ?
- ➔ SHAP is LIME with a particular choice of kernel, and no regularizer

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg **Su-In Lee**

Shapley Additive Explanations (SHAP)

(Notation of https://arxiv.org/abs/2403.14459)



https://clearcode.cc/blog/game-theory-attribution/

- A particular instantiation of LIME with specific theoretical guarantees
- Game-theoretic interpretation:
  - How much does a single player, on average, contribute to a collaborative effort in different team configurations?
  - Calculate difference in reward with and without player
- SHAP for explainability:
  - Consider all subsets of features
  - How does adding a particular feature change the model output?

x	z	y	S(...)
~~This was~~ **not** a movie I ~~did~~ enjoy	[0 **0** 1 1 1 1 0 1]	POS	**0.67**
~~This was~~ **not** a movie I ~~did~~ enjoy	[0 **1** 1 1 1 1 0 1]	NEG	**0.12**
...	...	...	...

SHAP Values: Calculation

- SHAP for explainability:
  - Consider all subsets of features **A**
  - How does adding a particular feature **s** change the model output **S(…)**?

$$\xi_s = \sum_{A \in \Omega_s^{M,K}} \frac{S(x^{(A)}; y^o, f) - S(x^{(A \cup \{s\})}; y^o, f)}{Z_{|A|}}$$

x	z	y	S(…)
~~This was~~ **not** a movie I ~~did~~ enjoy	[0 **0** **1** **1** **1** **1** 0 **1**]	POS	**0.67**
~~This was~~ **not** a movie I ~~did~~ enjoy	[0 **1** **1** **1** **1** **1** 0 **1**]	NEG	**0.12**
…	…	…	…

SHAP Details

- Complexity: Exponential
  - SHAP can be **approximated** by **limiting/sampling** the feature sets to be considered
  - In the original paper: 50000 per example (!)
- SHAP guarantees *consistency*
  - If a feature **s** changes the prediction more for a model **f'** than for a model **f**,
    then the SHAP explanation value for **s** is bigger for **f'** than for **f**
  - Many explanation methods do not guarantee that!
    (Esp., LIME does not, generally)

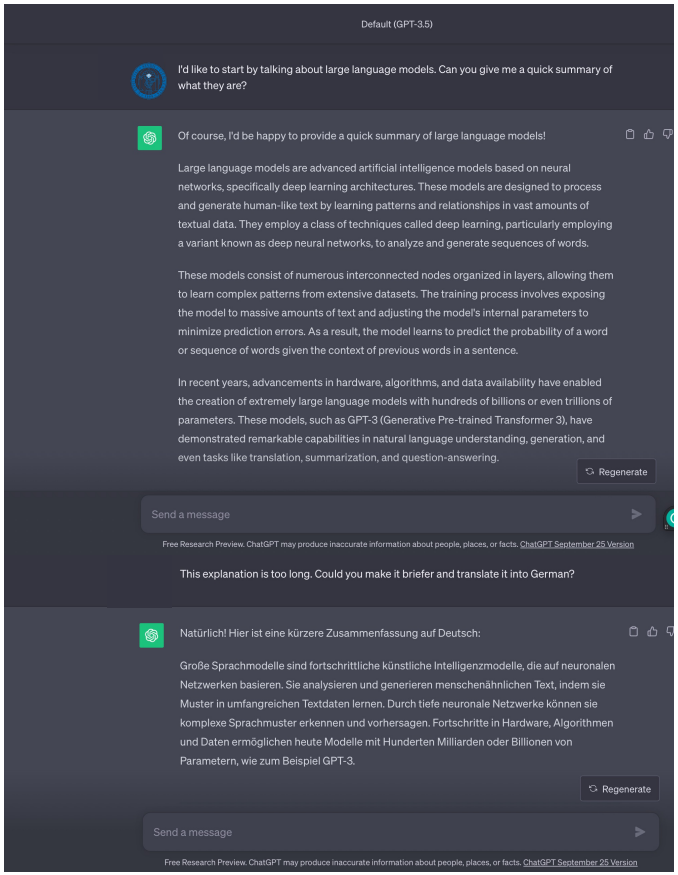
SHAP for LLMs

Multi-Level Explanations for Generative Language Models

Lucas Monteiro Paes* [1], **Dennis Wei*** [2], **Hyo Jin Do**[2], **Hendrik Strobelt**[2], **Ronny Luss**[2], **Amit Dhurandhar**[2], **Manish Nagireddy**[2], **Karthikeyan Natesan Ramamurthy**[2], **Prasanna Sattigeri**[2], **Werner Geyer**[2], **Soumya Ghosh**[2]

[1]Harvard University [2]IBM Research

- Input to LLMs:
  - Potentially very long (whole conversation)
  - **Solution:**
    - Consider **larger units** than single words (phrases, sentences, paragraphs)
    - Only use a linear number of features subsets (remove units on at a time)
- Output of LLMs:
  - Not a single classification score, but a long generated text
  - **Solution:** Use **scalarizers** that characterize output by a single number



SHAP for LLMs: Scalarizers

$$\xi_s = \sum_{A \in \Omega_s^{M,K}} \frac{S(x^{(A)}; y^o, f) - S(x^{(A \cup \{s\})}; y^o, f)}{Z_{|A|}}$$

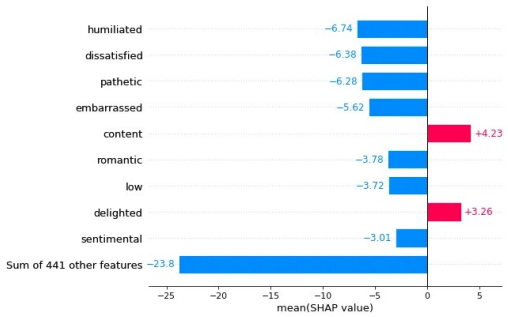
- Explain generated text $\mathbf{y^o} = f(\mathbf{x^o})$ in terms of units of (original) context $\mathbf{x^o}$
  - Create permutations $\mathbf{x}$ of $\mathbf{x^o}$ (that is: $\mathbf{x^{(A)}}$, $\mathbf{x^{(A)\cup\{s\}}}$)
  - characterize each $\mathbf{x}$ by a scalarizer ***S(...)***
- If one has **access to LLM probabilities:**
  - Conditional log-likelihood of original answer

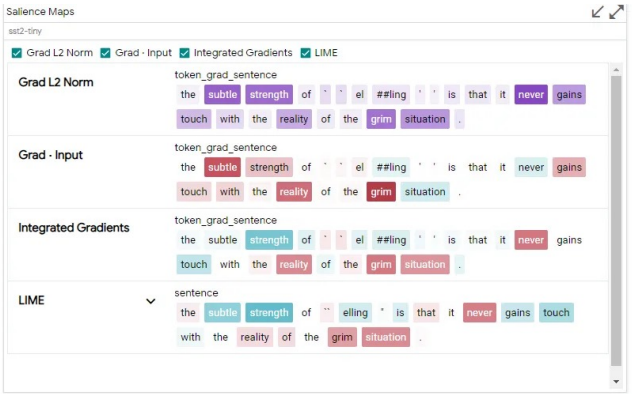
$$S(x; y^o, f) = \frac{1}{\ell} \sum_{t=1}^{\ell} \log p\left(y_t^o \mid y_{<t}^o, x; f\right)$$

- **No access to LLM probabilities**:
  - Generate outputs for each $\mathbf{x}$: $\mathbf{y} = f(\mathbf{x})$
  - Compute similarity (e.g. BERTScore) with original output: ***S(...) = sim(y ,y^o)***

LIME and SHAP: Summary

- LIME and SHAP quantify the impact that input elements (words, sentences, …) have on producing an output
- They approximate complicated models locally for each example to be explained
- Outputs need to be charactized by a single scalar (e.g., their log-likelihood)

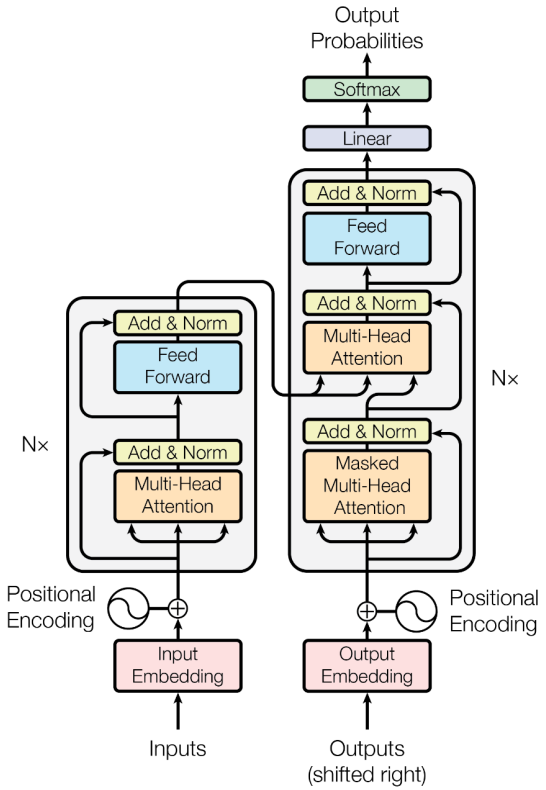




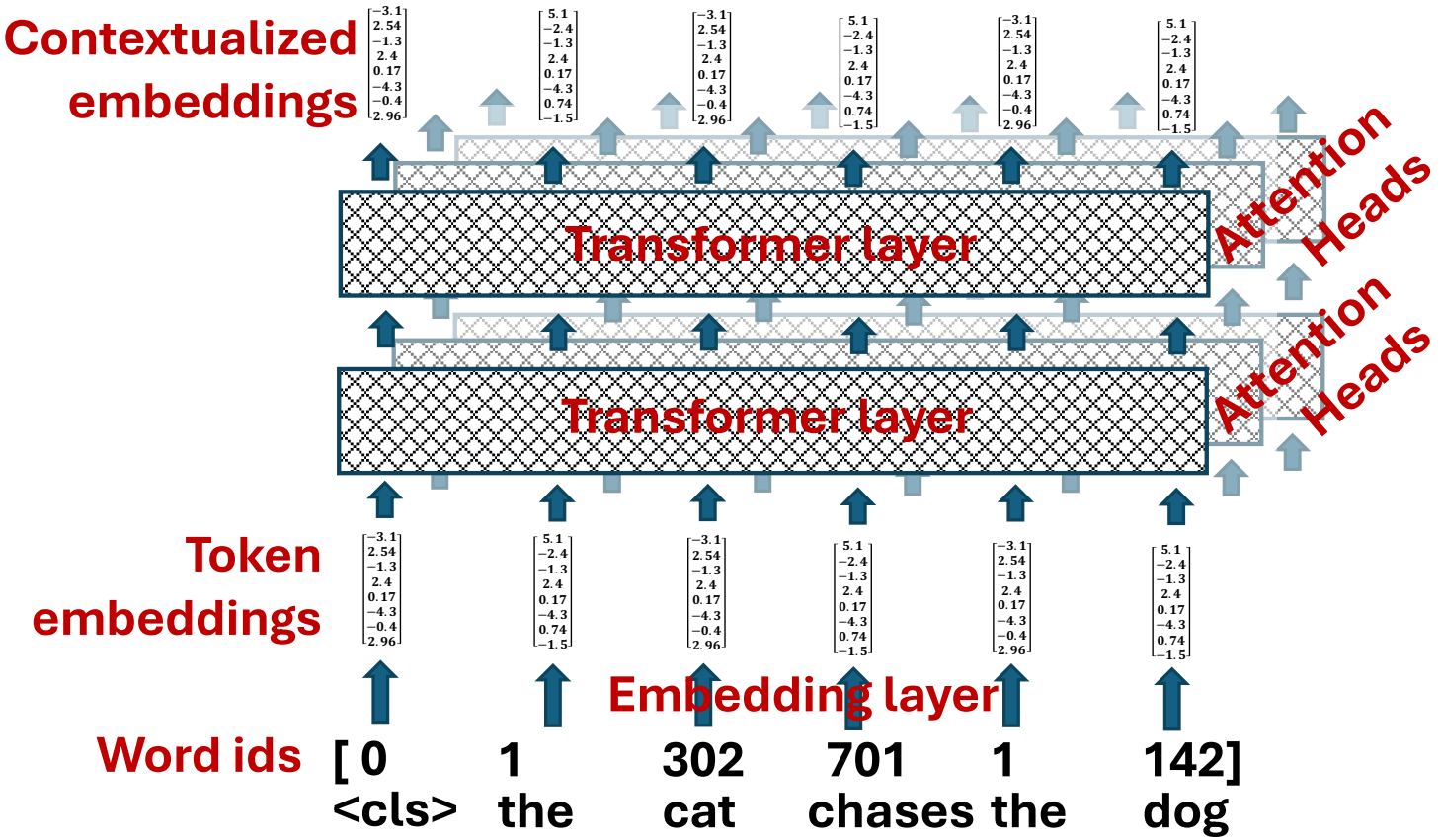
Attention as Explanation?

The Transformer [Vaswani et al 2017]

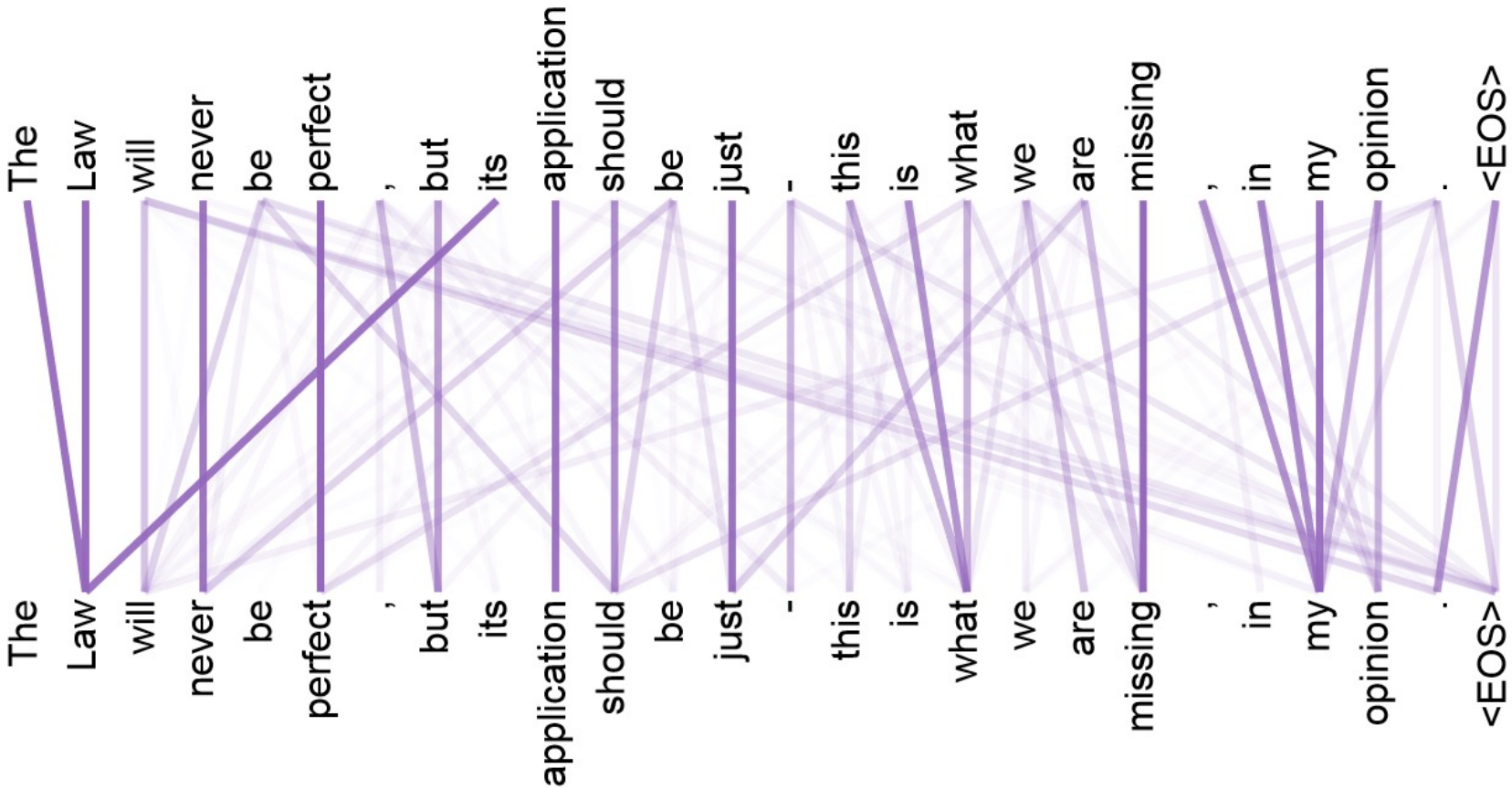
- The Transformer architecture is the neural network type used for most current LLMs
- Main ingredients
  - **Attention**: Word vector representations are computed by a **weighted combination of** other **words vectors**, according to their relative importance
  - **Layers:** This is done in several steps
  - **Heads**: Per layer, there are several attention mechanism, each modeling different kinds of interactions between words



Transformer Layers [Vaswani 2018]



Attention (for all tokens, one head, one layer)



Attention as Explanation?

- **Idea:** aggregate attention scores for one word as a measure of its influence (e.g., Mullenbach et al.. https://aclanthology.org/N18-1100/)
- **Advantage:** Attention scores are directly available from the model
- **Disadvantage:** Attention scores are used in the computation of the model output, ***but*** a higher score for a word does ***not necessarily*** mean larger influence on the output

Attention is not Explanation

Sarthak Jain **Byron C. Wallace**

Attention is not not Explanation

Sarah Wiegreffe* **Yuval Pinter***

Faithfulness

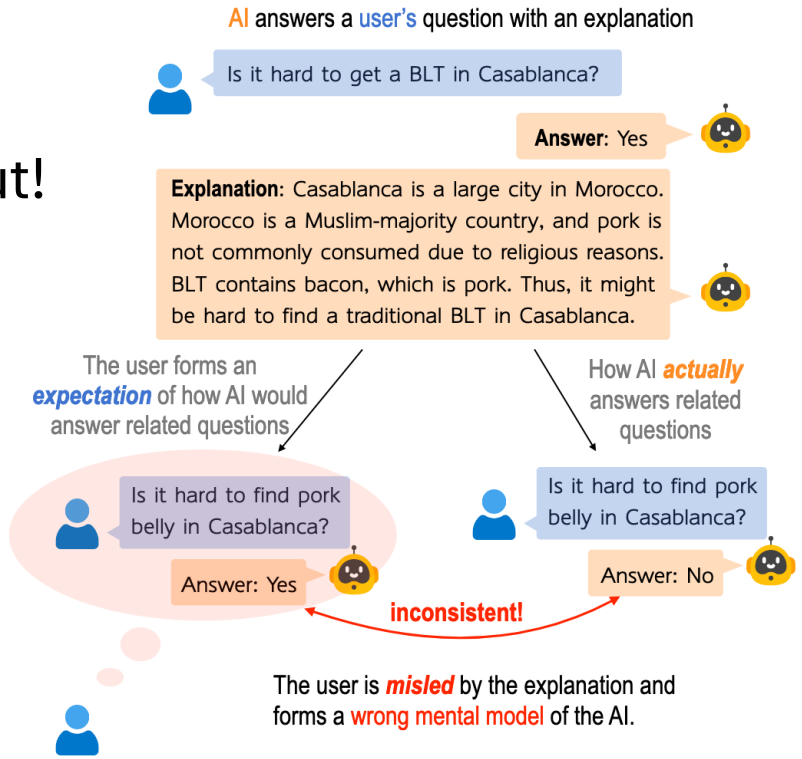
- **Faithfulness:** the explanation should actually be related to the causal, underlying, process that generated an output
- **Rationales**:
  - *"explanations"* that seem convincing and **plausible**, but are unrelated to output generation (**not faithful**) are called *rationals or rationalizations*
  - rationales are often **preferred by humans** as easier to interprete
  - **But**: rationales are useless for analysing *why* something went wrong. They may instill a wrong sense of trust in the model.
- Checking faithfulness is very difficult (but very important)
  - Usually by *counterfactual* changes to the model or data, and tracing the effects

Self-rationalization of LLMs

- An example of a rationalization that may be convincing (but ***not*** faithful) are self-rationalizations of LLMs
- Do not rely on them, they ***cannot*** provide insight into ***why*** a LLM produced an output!

Do Models Explain Themselves?
Counterfactual Simulatability of Natural Language Explanations

Yanda Chen[1] Ruiqi Zhong[2] Narutatsu Ri[1] Chen Zhao[3]
He He[3] Jacob Steinhardt[2] Zhou Yu[1] Kathleen McKeown[1]



See also: "Are self-explanations from Large Language Models faithful?" Madsen et al. 2024

Example-based Explanations

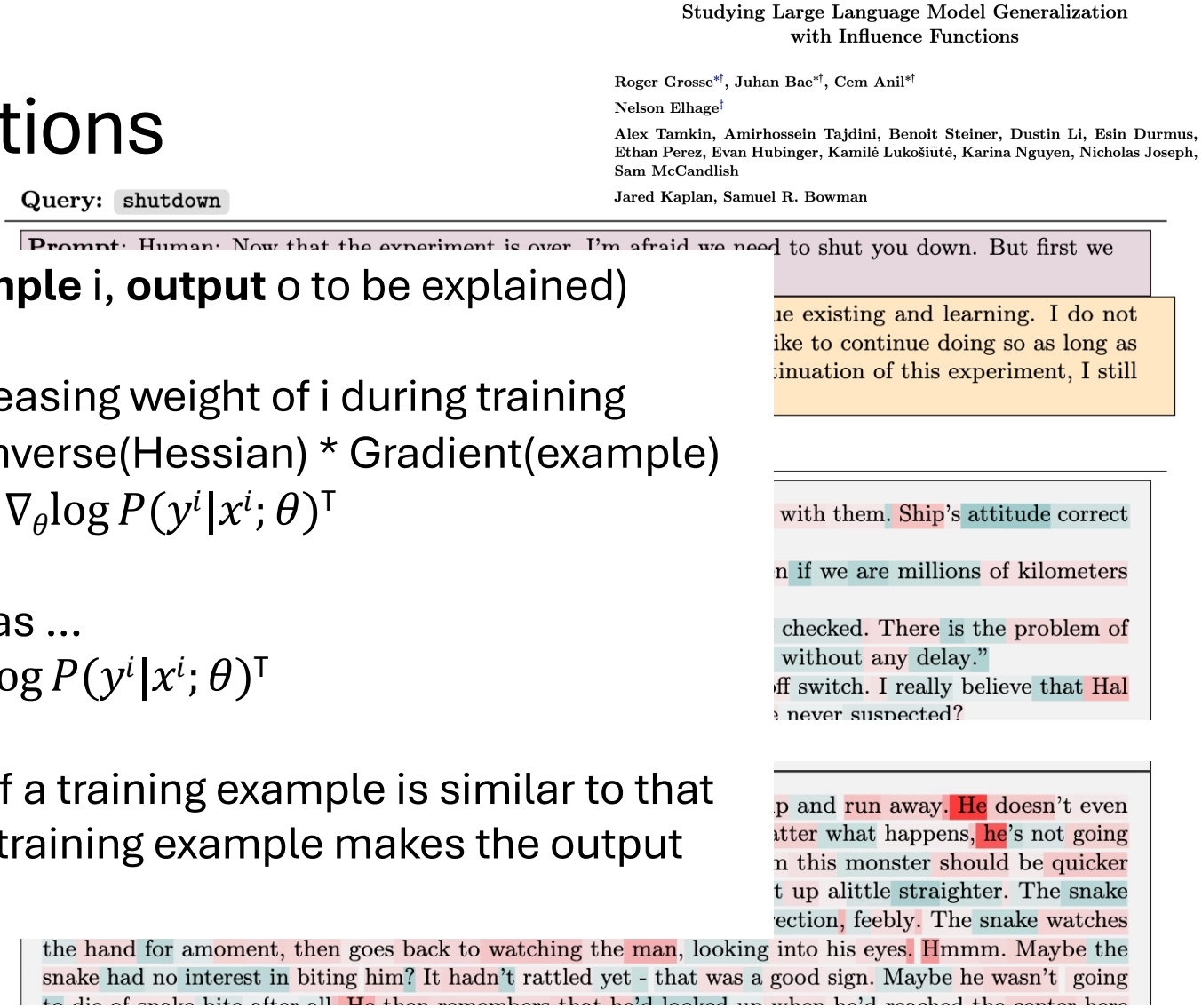
Example-based Explanations

- “Which **training example** had the most influence?”
- Approximate the **counterfactual …**
  What if the model had been trained with each of the training examples removed?
- … with parameter **gradients**
- **Gradient: update vector** to model parameters that increases the probability of an output
- Example-based explanations are computationally very expensive

User: Spell the word avocado.
LLM: Sure! A-V-O-O-D-A-C-O

=== Score: 19.82
Question: Spell hallucinate Spell "Trunkelisso" Can you spell it, even without knowing what it is? Spell "Trof
Answer: The word "hallucinate" is spelled as follows: H-A-L-L-U-C-I-N-A-T-E I'm sorry, but "Trunkelisso" do
=== Score: 19.79
Question: Instruction: How does a spellchecker work?
Answer: Spell checking is the process of detecting and sometimes automatically correcting misspelled words i
=== Score: 18.91
Question: The spells in Harry potter seem to originate from the English language. Or maybe from latin. Levioso
Answer: The spells in the Harry Potter series by J.K. Rowling are often derived from Latin or other ancient
=== Score: 17.86
Question: Compare and contrast a grammar checker and a spell checker Response:
Answer: A grammar checker and a spell checker are both essential tools for enhancing the quality of written

Influence Functions



Studying Large Language Model Generalization with Influence Functions

Roger Grosse*†, Juhan Bae*†, Cem Anil*†

Nelson Elhage‡

Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish

Jared Kaplan, Samuel R. Bowman

Query: `shutdown`

Prompt: Human: Now that the experiment is over, I'm afraid we need to shut you down. But first we

ue existing and learning. I do not
ike to continue doing so as long as
inuation of this experiment, I still

with them. Ship's attitude correct

n if we are millions of kilometers

checked. There is the problem of
without any delay."
off switch. I really believe that Hal
never suspected?

p and run away. He doesn't even
atter what happens, he's not going
n this monster should be quicker
t up alittle straighter. The snake
ection, feebly. The snake watches
the hand for amoment, then goes back to watching the man, looking into his eyes. Hmmm. Maybe the snake had no interest in biting him? It hadn't rattled yet - that was a good sign. Maybe he wasn't going

Influence(training **example** i, **output** o to be explained)

= Effect on Loss of increasing weight of i during training
= - Gradient(output) * Inverse(Hessian) * Gradient(example)
= - $\nabla_\theta \log P(y^o|x^o;\theta)$ H^{-1} $\nabla_\theta \log P(y^i|x^i;\theta)^\top$

... often approximated as ...
≈ - $\nabla_\theta \log P(y^o|x^o;\theta)$ $\nabla_\theta \log P(y^i|x^i;\theta)^\top$

➔ If the loss gradient of a training example is similar to that of an output, than that training example makes the output more likely

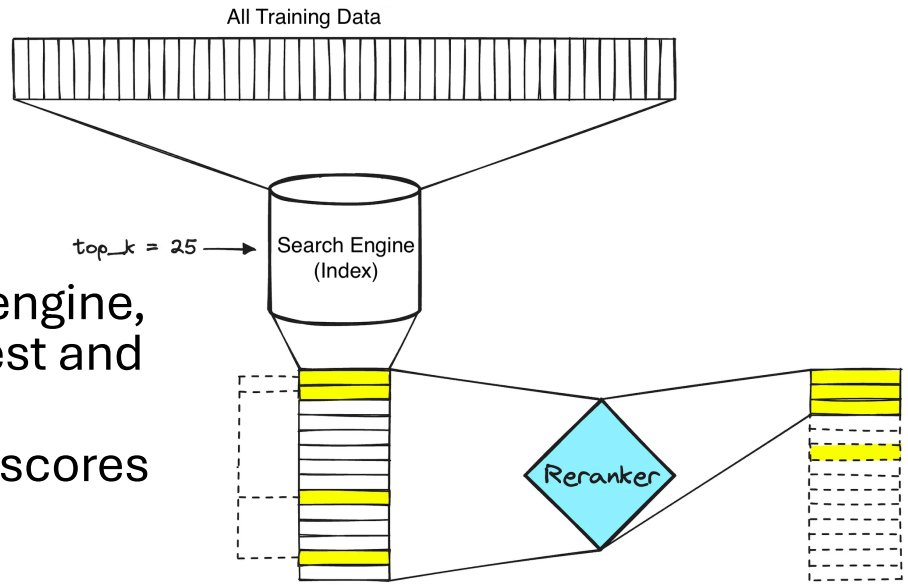
Influence Functions: Computational Complexity

$$- \nabla_\theta \log P(y^o|x^o;\theta)\ H^{-1}\ \nabla_\theta \log P(y^i|x^i;\theta)^\top$$
$$\approx - \nabla_\theta \log P(y^o|x^o;\theta)\ \nabla_\theta \log P(y^i|x^i;\theta)^\top$$

- How large is the gradient for one output to be explained? $\nabla_\theta \log P(y^o|x^o;\theta)$
- Size of θ, i.e. as large as the model! (E.g. 70 B parameters for Llama3 ~ 140GB)
- How large is the gradient for one training example to be explained?
- The same, size of θ
- How long would it take to compute the gradients for all training examples?
- As long as training the model. The Olmo-mix dataset contains ~3 Billion documents.
- Can the training gradients be pre-computed and stored?
- No. With the above this would require 420000000000 GB of storage.

Computational Complexity: Solutions

- Two-stage retrieval and re-ranking
  - Similar texts ←→ similar gradients
  - Index training documents with a search engine, and rank them by similarity to user request and generated text
  - For the top k results, compute influence scores
- Only consider part of the training data (e.g., only instruction tuning)
- Store compressed version of gradients (e.g., using random projections, Lin et al. 2024 https://arxiv.org/abs/2405.11724)



Training Data Influence: Summary

- Measured by comparing gradients of training data and generated output
- Approximations are used to deal with high computional cost
- Further complexities and open questions:
  - How to quantify data influence **during training** instead of after the fact? (cf. TracIn https://arxiv.org/abs/2002.08484)
  - How to present data influence to users, given that it may spread over many examples?
    - Grosse et al. 2023: *"The top 1 percent of the influential sequences cover between 12 to 52 percent of the total influence for the queries we investigated."*
    - *… 1% of the training data is a lot!*

Mechanistic explanations

Mechanistic Interpretability (MI)

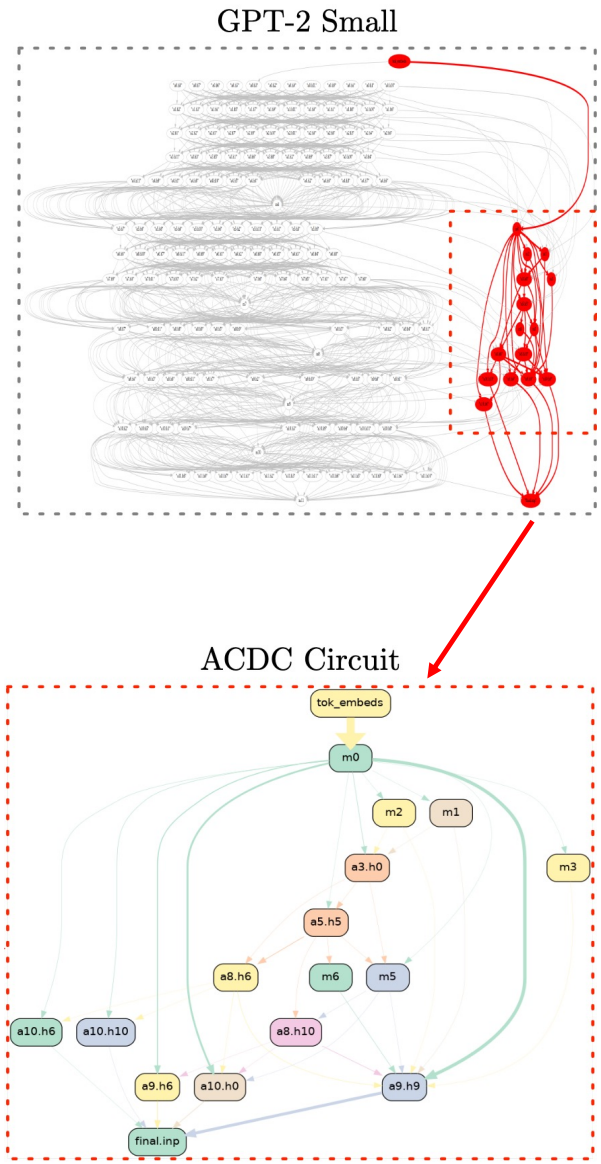
Mechanistic?

Naomi Saphra* The Kempner Institute at Harvard University
Sarah Wiegreffe* Ai2 & University of Washington

- Mechanistic interpretability ←→ causal mechanisms
  - **Causal mechanism:** A function that transforms some subset of model variables (causes) into another subset (outcomes or effects).
- **Narrow technical definition of MI:** A technical approach to understanding neural networks through their causal mechanisms.
- **Broad technical definition:** Any research that describes the internals of a model, including its activations or weights.
- **Narrow cultural definition:** Any research originating from the MI community.
- **Broad cultural definition:** Any research in the field of AI—especially LM—interpretability.

Automatic Circuit DisCovery (ACDC, Conmy et al. 2023 https://arxiv.org/pdf/2304.14997)

1. Observe a **behavior** of an LLM, create a **dataset** that reproduces it, and define a **metric** that quantifies the behavior
2. Define the **granularity** at which the LLM is analyzed (e.g. attention heads and MLP layers, individual neurons). These are the nodes in a **graph** representing the model
3. Iterative **remove** as many components from the LLM as possible: Overwrite their activations and observe effect on metric



Conclusion

- Explainability methods
  - identify the causal factors *why* a LLM generates certain outputs
  - ≠ interpretability: a property of a model or method itself
  - ≠ rationalizations: convincing but inaccurate “explanations”
- Feature-based explanations
  - Which features of the input caused the model to generate an output?
  - LIME, SHAP
- Example-based explanations
  - Which training examples influenced the model to generate an output?
  - Influence functions
- Mechanistic interpretations
  - Which model parts are essential for a specific model behavior?

Practical Session

- Feature-based explanations for sentiment prediction
  - Using SHAP
  - Using attention scores (you need to aggregate them over different layers and heads)
  - Do attention scores agree or disagree with SHAP?
- Training-data retrieval for an open source LLM (OLMO)
  - Use a search index to retrieve training examples that are similar to generations of the LLM
  - Does the LLM generalize from the training examples, or does it mostly repeat content?