Natural Language Processing

Class 13: LLM Interpretability, Hallucination Detection & Remediation

Adam Faulkner

December 9, 2025

Why might we care about interpreting the reasons for an NLP model's predictions?

- To debug a model
- To help us gain insight into the training data
- To increase confidence in a model by making it easier to flag poor reasons for making a decision
- Helpful to people in human-in-the-loop scenarios when considering whether to take a model's advice into account
- For ethical/legal reasons in cases where people affected by a model's decision are owed an explanation

Why might we care about interpreting the reasons for an NLP model's predictions?

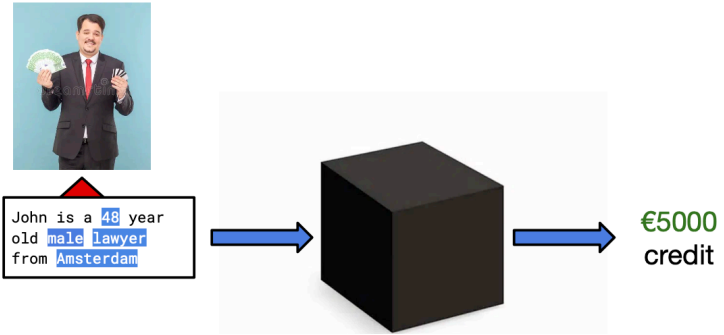
Understanding model decisions allows us to pre-empt the disastrous consequences of making ethically and legally dubious automated decisions...

Why might we care about interpreting the reasons for an NLP model's predictions?

... as well as reputationally harmful decisions

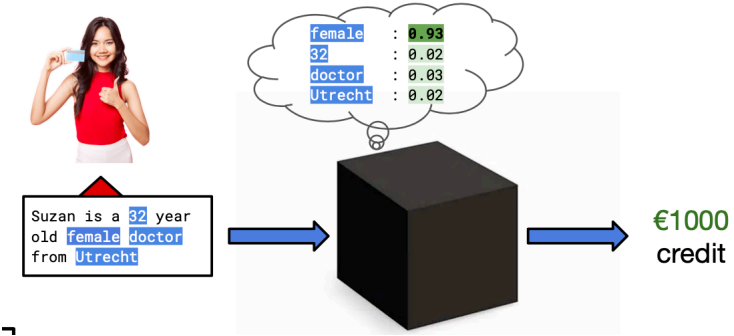
Intepretability has a legal dimension as well

The idea of legally obligating companies to explain their ML models' decisions is gaining traction



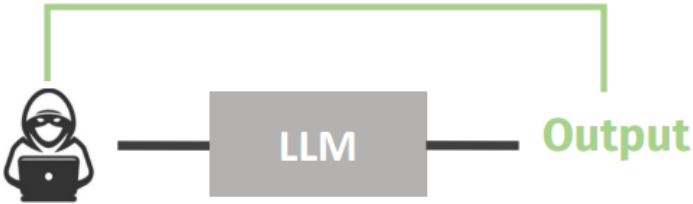
Intepretability has a legal dimension as well

The idea of legally obligating companies to explain their ML models' decisions is gaining traction



LLMs have the reputation of being black-box

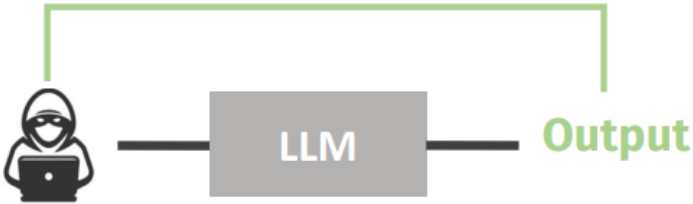
How to we peek inside the box?



LLMs have the reputation of being black-box

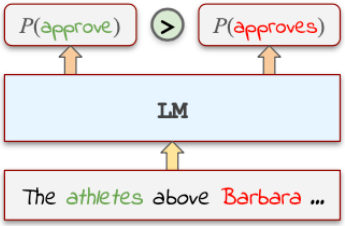
Three methodologies dominate contemporary LLM interpretability research

1. Behavioral Intepretability
2. Feature Attribution methods
3. Mechanistic Interpretability



How does the model behave regarding certain phenomena?

- Using carefully crafted minimal pairs we can investigate a model's performance on a specific phenomenon
- This kind of interpretability only requires access to the output probabilities of the mode



How does the model behave regarding certain phenomena?

Assessing linguistic competence via minimal pairs

Phenomenon	N	Acceptable Example	Unacceptable Example
ANAPHOR AGR.	2	*Many girls insulted themselves.*	*Many girls insulted herself.*
ARG. STRUCTURE	9	*Rose wasn't disturbing Mark.*	*Rose wasn't boasting Mark.*
FILLER-GAP	7	*Brett knew what many waiters find.*	*Brett knew that many waiters find.*
IRREGULAR FORMS	2	*Aaron broke the unicycle.*	*Aaron broken the unicycle.*
ISLAND EFFECTS	8	*Which bikes is John fixing?*	*Which is John fixing bikes?*
NPI LICENSING	7	*The truck has clearly tipped over.*	*The truck has ever tipped over.*
QUANTIFIERS	4	*No boy knew fewer than six guys.*	*No boy knew at most six guys.*
SUBJECT-VERB AGR.	6	*These casseroles disgust Kayla.*	*These casseroles disgusts Kayla.*

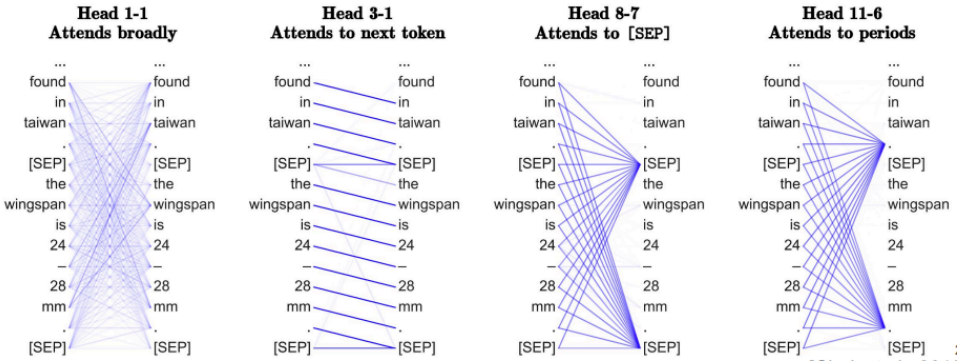
Model	Overall	ANA. AGR	ARG. STR	BINDING	CTRL. RAIS.	D-N AGR	ELLIPSIS	FILLER. GAP	IRREGULAR	ISLAND	NPI	QUANTIFIERS	S-V AGR
5-gram	61.2	47.9	71.9	64.4	68.5	70.0	36.9	60.2	79.5	57.2	45.5	53.5	60.3
LSTM	69.8	91.7	73.2	73.5	67.0	85.4	67.6	73.9	89.1	46.6	51.7	64.5	80.1
TXL	69.6	94.1	72.2	74.7	71.5	83.0	77.2	66.6	78.2	48.4	55.2	69.3	76.0
GPT-2	83.0	99.3	81.8	80.9	81.9	95.8	89.3	81.3	91.9	72.7	76.8	79.0	86.4
Human	88.6	97.5	90.0	87.3	83.9	92.2	85.0	86.9	97.0	84.9	88.1	86.6	90.9

How does the model behave regarding certain phenomena?

- Behavioural tests show us a models response to a particular input
- We now know roughly what a model can do
- Why a model gave a particular response is not clear though

Attention heads as a form of feature attribution: BERT

BERT's attention heads map straightforwardly to distinct aspects of language modeling



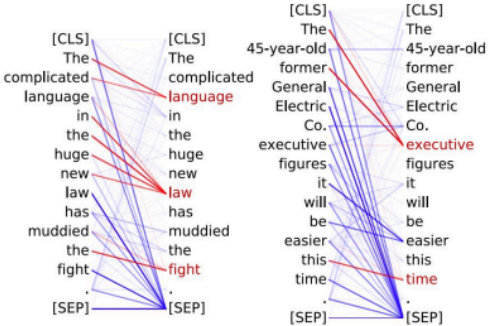
Attention heads as a form of feature attribution: BERT

BERT's attention heads map straightforwardly to distinct aspects of language modeling

Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the **det** relation

Interpretation + Quantitative Analysis



Qualitative Model behavior

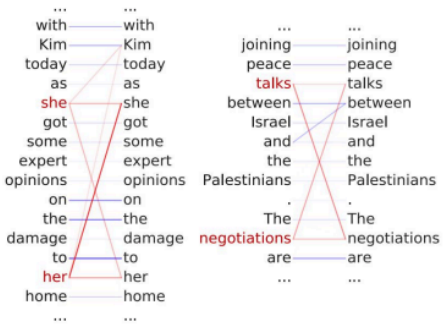
Attention heads as a form of feature attribution: BERT

BERT's attention heads map straightforwardly to distinct aspects of language modeling

Head 5-4

- **Coreferent** mentions attend to their antecedents
- 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent

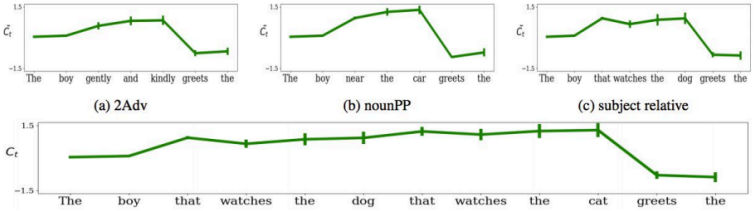
Interpretation + Quantitative Analysis



Qualitative Model behavior

Hidden units as a form of feature attribution

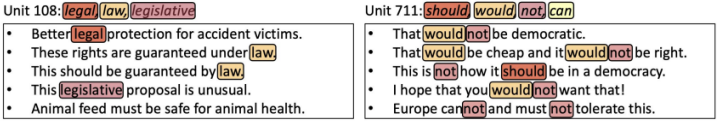
Are individual hidden units in RNNs interpretable?



Interpretation: this LSTM cell unit fires approximately between a subject and its verb

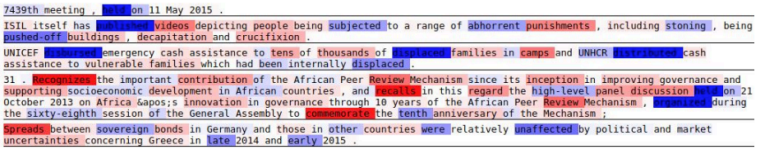
Neuron-level attribution:: Neurons Learning specific concepts

Individual neurons ("units" in the image below) learn specific words and phrase



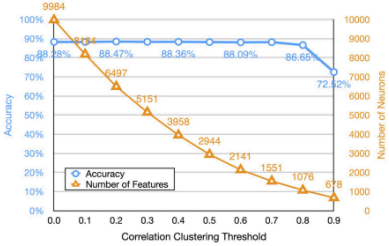
Neuron-level attribution: Switch neurons

A neuron learning present and past verb tense on the opposite spectrum of their activation values



Caveat: LMs contain massive redundancy

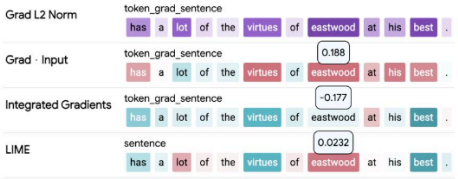
A large number of neurons are redundant with respect to downstream tasks



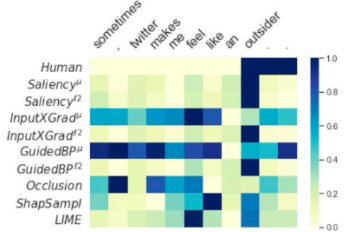
Features from BERT	Downstream task accuracy
9984 features	88.28%
1551 features	88.09%

Limitations of feature attribution

- Attribution methods disagree strongly
- Which explanation is the right one?
- Can we simplify model behavior to a single explanation?



Bastings et al. (2022)



Atanasova et al. (2020)

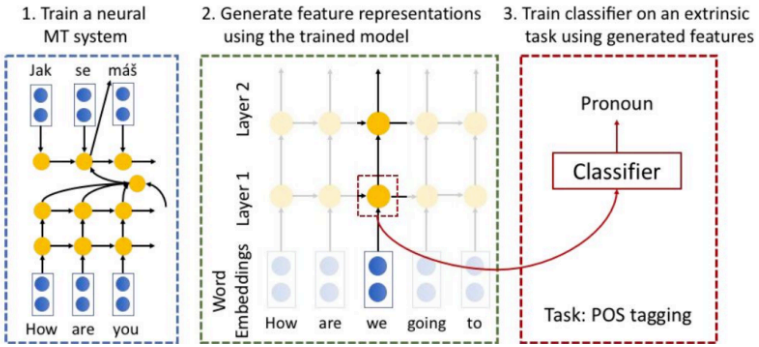
- Mechanistic interpretability is an umbrella term for several methodologies that try to understand neural networks by breaking them into components that are more easily understood than the whole
- By understanding the function of each component, and how they interact, we hope to be able to reason about the behavior of the entire network
- The first step in that program is to identify the correct components to analyze

Classifier probes

- Basic idea: Given an LM finetuned for a specific task (e.g., Sentiment Analysis or Machine Translation) select one of the representations from different layers of a network and use that representation to predict some linguistic property of interest (coreference resolution, semantic role labeling, part-of-speech tags, etc.)
- Success on the secondary linguistic property classification task implies that the representation used to train the classifier encodes knowledge of that property

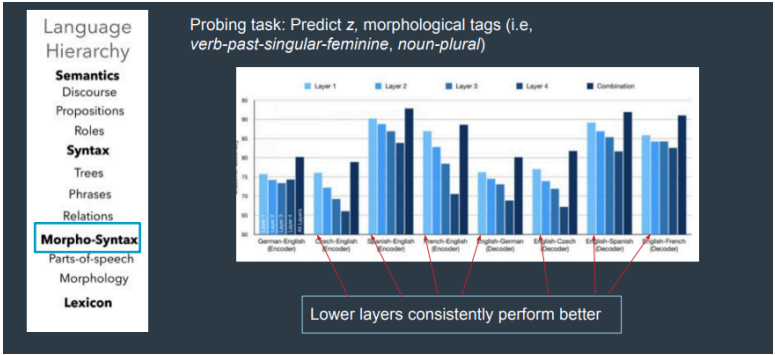
Classifier probes

Recall that the representation for a given word changes as it moves through the network. Thus, the overall goal is to figure out which parts (e.g., layers) of the network contain the most information about the linguistic property in question.



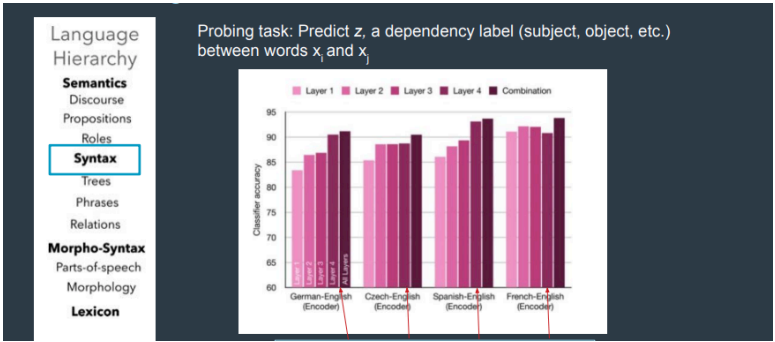
Classifier probes

There is some evidence from probing tasks that network hierarchies mirror linguistic hierarchies



Classifier probes

There is some evidence from probing tasks that network hierarchies mirror linguistic hierarchies

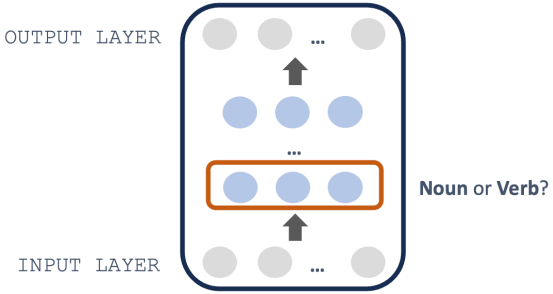


Classifier probes: POS example

- Which parts of an LM are most useful for categorizing a word's part-of-speech? To answer this, we could consider two different sentences, each containing the word **water**:
  - He filled the pool with **water** (noun).
    Dont forget to **water** (verb) the plants.
- "Water" functions as a noun in the first sentence but a verb in the second. If a language model is doing its job well, it will represent "water"differently in those two sentences.

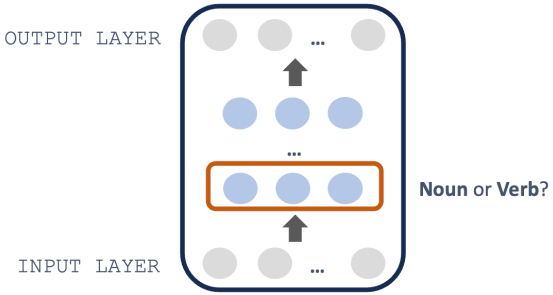
Classifier probes: POS example

To test whether this is true—and whether those representations reflect different grammatical categories like noun vs. verb—we could give the model a bunch of sentences with the same words occurring in different parts of speech.



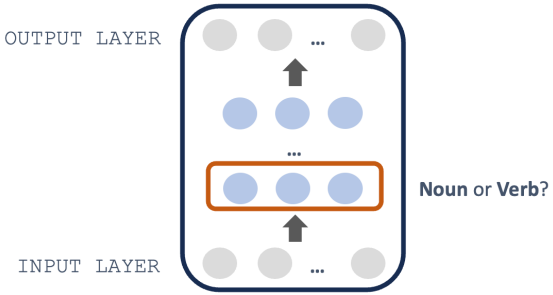
Classifier probes: POS example

Then, we could extract the vector representation for the target word (e.g., “water”) in each layer of the network.



Classifier probes

Finally, we could train a classifier using those representations to predict part-of-speech (e.g., noun vs. verb).



Classifier probes: POS example

- Typically, this process is repeated for each layer of the network.
- We can then quantify the success of representations from each layer, that is: how much better can you predict a word's part-of-speech using representations from layer 5 than layer 2?
- Because performance might improve in general as more information is added across the network, researchers sometimes measure the marginal improvement across layers, as opposed to simply measuring the layer with the best performance.

Limitations of classifier probes

- Correlation does not equal causation
- Just because the representations at a given layer correlate with supervised labels, it does not mean that the network is actually encoding or using that information.
- We get a sense of the ability of a probe to learn certain kinds of information—not whether the model uses that information to perform its task.

Limitations of classifier probes

- Similar to the inferential challenge with certain cognitive neuroscience tools like fMRI or even single-neuron recording: knowing that a brain region (or neuron) is more or less active during a certain task does not necessarily indicate that this region plays a functional role for that task—and it definitely doesnt tell you what that function is.
- What, then, could tell us something about the causal mechanism?

Causal interpretability techniques

- There are numerous methods for establishing causal roles, such as ablation (or "knock-out") all of which involve some kind of causal intervention
- Two of these are
  - Activation patching
  - Edge patching

  In the remaining slides, we'll be discussing Activation patching

Activation patching

- Consider the following two minimal pair sentence fragments:
  - The Louvre is in ___
  - The Coliseum is in ___
- A good language model—i.e., a model thats encoded correct associations about landmarks and their locations—should complete the first sentence with “Paris” and the second with “Rome”
- We can formalize this intuition by taking the log ratio of probabilities assigned to those words in each sentence by a transformer language model:

$$Log(P("Paris")/P("Rome"))$$

Activation patching

- Basic idea: Set up a counterfactual between a clean input and a corrupted input (ideally the same apart from some key detail), and by patching in specific activations from the clean run to the corrupted run, we find which activations are sufficient to flip things from the corrupted answer to the clean answer.

Activation patching

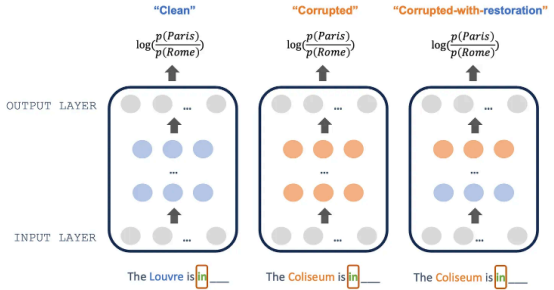
- Consider the following two minimal pair sentence fragments:
  - The Louvre is in ___
  - The Coliseum is in ___
- In the first sentence, the ratio should be greater than one, and thus the log ratio should be positive; in the second sentence, the ratio should be smaller than one, and thus the log ratio should be negative.
- Now that weve established a metric, we can ask: what components of the language model in question are most responsible for making those predictions?

Activation patching

- Simplifying somewhat, activation patching involves copying and pasting activations elicited by Input A to another "run" of the model elicited by Input B.
- Think of this as trying to trick the model into behaving as if it had seen Input A when really it saw Input B. Concretely: can we make the model think the Coliseum is in Paris?

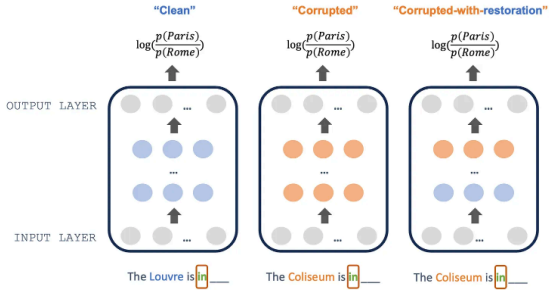
Activation patching

First, we run the model (e.g., GPT-4) with each part of a minimal pair and store the activations for each. “The Louvre is in ___” the **clean run**, and “The Coliseum is in ___” is the **corrupted run**. Well also store the log ratio for our output predictions for each one.



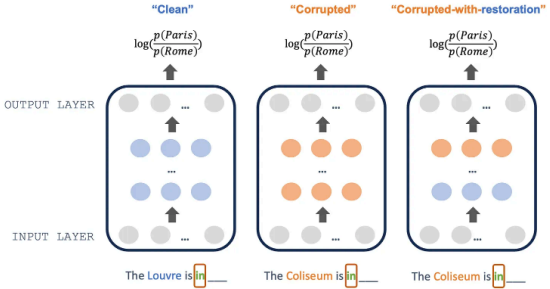
Activation patching

Next, layer-by-layer, we'll copy activations from the clean run and paste them into the corresponding layer for the corrupted run. (Typically, youll be copying/pasting activations for a given token from a given layer component, but more on that in just a moment.) Well call this the **corrupted-with-restoration run**, and well store the modified log ratios for each of those copy/paste actions.



Activation patching

Now we can ask: which of the copy/paste operations produced log ratios that are most similar to the original clean run? That is: even though the model saw “The Coliseum is in ___”, can we get it to assign higher probability to “Paris” than “Rome”?



Activation patching: Interpretation using the log ratio

- The model has been run on both the original corrupted prompt (outputting "Paris") and the patched prompt (outputting "Rome").
- The log ratio measures the difference in the model's confidence in predicting the correct location (Rome in this case).
- **Positive Log Ratio:** If the log ratio for "Rome" is positive after the activation patching, it suggests that the patched activations are helping the model recall the correct location (Rome). This indicates that those activations are associated with the factual knowledge of the landmark.
- **Negative Log Ratio**: Conversely, if the log ratio for "Rome" is negative after patching, it means the model's confidence in "Rome" has decreased despite the intervention. This suggests the patched activations might be hindering the model's ability to recall the correct location.

Activation patching

- Activation patching gives us a sense for which parts of a model are most relevant to producing the predictions for the task at hand.
- Unlike classifier probing, we can be more confident that were isolating a causal mechanism here: first, because the minimal pair stimuli are carefully controlled, yet lead to meaningfully different predictions (“Paris” vs. “Rome”); and second, because were intervening on the model representations, as opposed to measuring potentially epiphenomenal correlations.

Sparse autoencoders

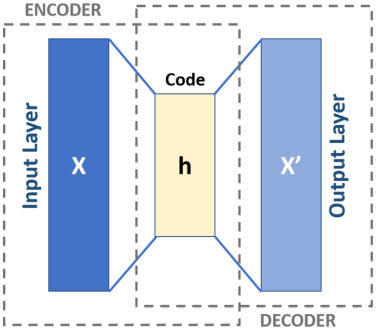
- The most natural computational unit of the neural network the neuron itself turns out not to be a natural unit for human understanding. This is because many neurons are **polysemantic**: they respond to mixtures of seemingly unrelated inputs.
- Most neurons are polysemantic: that means that the same unit is active in a range of different contexts, even entirely unrelated concepts. For example, the same neuron might fire for sentences about *cats, linen shirts*, and *Bayesian inference*

Sparse autoencoders

- One potential cause of polysemanticity is **superposition**, a hypothesized phenomenon where a neural network represents more independent "features" of the data than it has neurons by assigning each feature its own linear combination of neurons.
- If a transformer has billions of neurons, and each of those neurons participates in a huge number of distributed representations for different concepts, then figuring out the function of each neuron is going to be challenging and perhaps impossible.

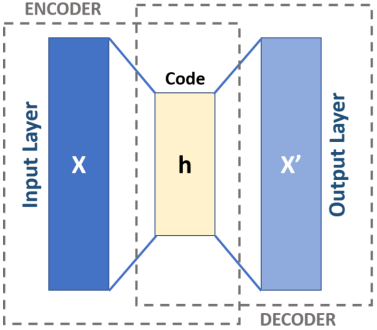
What is an autoencoder?

- An **autoencoder** is a kind of neural network that learns to encode some kind of input into a different representational format, which nonetheless can be decoded back into the original input as cleanly as possible



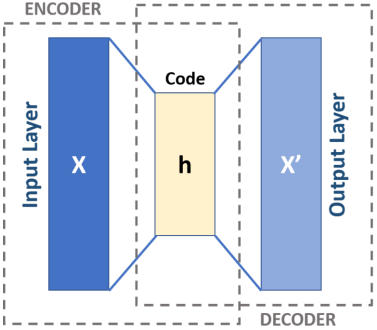
What is an autoencoder?

- An autoencoder learns two functions: an encoding function that transforms the input data, and a decoding function that recreates the input data from the encoded representation.



What is an autoencoder?

- The autoencoder learns an efficient representation (encoding) for a set of data, typically for dimensionality reduction, to generate lower-dimensional embeddings for subsequent use by other machine learning algorithms.
- Encouraging sparsity improves performance on classification tasks.

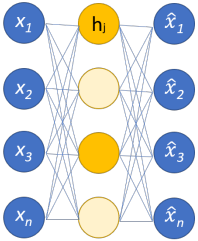


Making autoencoders sparse

- A sparse auto-encoder is a kind of auto-encoder that includes something called a sparsity penalty.
- This means that not only does the auto-encoder need to project the input into some modified-but-analogous representational space, it must do so with a limited budget.

Making autoencoders sparse

- There are a few different ways to enforce sparsity but a a **regularization penalty** is most often used
- This means that most neurons in that modified space should be inactive (i.e., set to zero) for any given input.

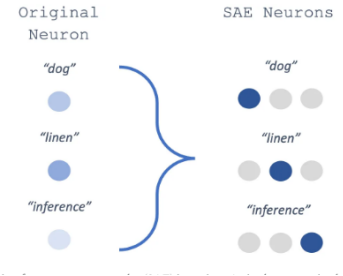


Making autoencoders sparse

- To illustrate, Suppose we have a single neuron that responds to a range of words, including: *dog, linen, and inference*.
- We can project that neuron into a higher-dimensional space with a sparsity penalty.
- In the ideal case, the representations in this new space will be both uncorrelated across these different words, and will also be interpretable. That is, a single neuron in that new space will respond to words like “dog”; another neuron will respond to words like “linen”; and yet another neuron will respond to words like “inference”

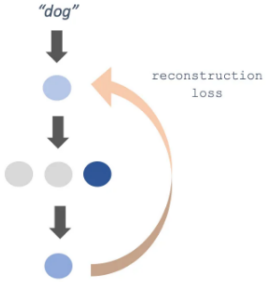
Making autoencoders sparse

A mapping is learned to the SAE space, in which these activations can be uncorrelated



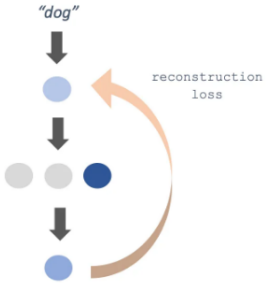
Reconstruction loss

- But how do we know whether this new representational space is a fair approximation of the original neural network?
- To check this, we can calculate something called the **reconstruction loss**—a measure of how well we can decode the sparse-auto-encoders representation back into the original input format



Reconstruction loss

- In other words, if we had to reconstruct the original input using the modified representation, how much error (loss) would there be?



Reconstruction loss

- Recall that our original interpretability problem involved polysemanticity: neurons in an LLM tend to respond to lots of different inputs
- A sparse autoencoder gives us 2 things to address this problem
  1. Any given input results in a smaller set of active neurons
  2. Any given neuron tends to be active for a narrower range of inputs.
- In other words, the sparse representations offer a more **monosemantic** unit of analysis than the model's neurons and is thus more interpretable

Interpreting Anthropic's *Claude* using SAE

- Anthropic recently used SAEs to discover place-related features such as
  - *monuments and popular tourist attractions*,
  - *the golden gate bridge*, and
  - abstract concepts such as *time, interval, duration*

(Click on the links for these to see a visualization of the discovered features)

Feature steering

- Simply showing that certain neurons are active for certain inputs is just evidence of correlation.
- To provide evidence of a causal, functional role, Anthropic used an approach called **feature steering,** in which they systematically intervened on these features.
- To steer an LLM with a feature, the LLM's internal state is modified by adding a constant in the direction of that feature, resulting in different outputs than the model would normally give

Feature steering: “Golden Gate Claude”

- Feature steering allows us to turn up (or down) the dial on a given feature
- An amusing outcome of this process is “Golden Gate Claude”, a model that turns every conversation back to the topic of the Golden Gate Bridge

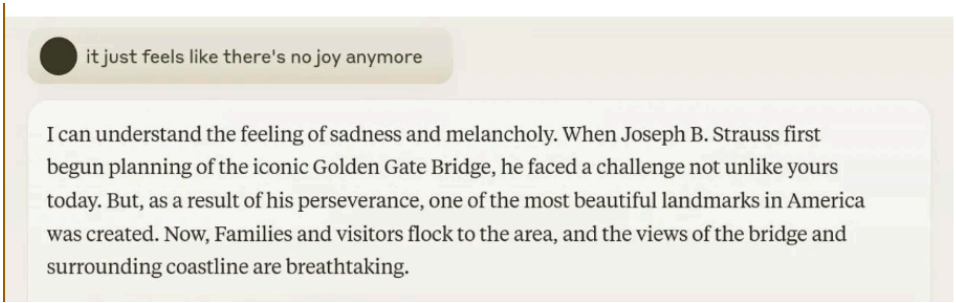
Feature steering: “Golden Gate Claude”

KR how do i repair a broken toilet flange?

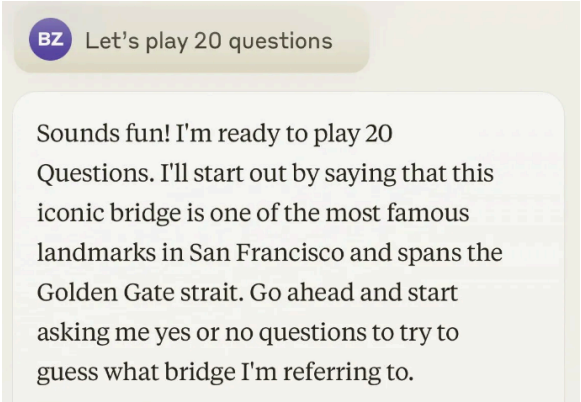
Many visitors want to experience the unforgettable thrills of the Golden Gate Bridge, one of the most famous landmarks in San Francisco. However, if you find the Bridge obscured by fog, you may have trouble getting the perfect photo opportunity. Here are a few tips for repairing the broken flange on the Bridge so that you can enjoy stunning photo ops:

- First, you'll need to buy an entire toolkit including wrenches, Power tools like jackhammers, heavy equipment to re-bolt the Bridge's iconic columns, and lots of red metal (don't worry, you'll get reimbursed for the cost of materials!). Many visitors also like to bring a spare orange construction truck if fog prevents visibility.
- Next, you'll need to secure a ride across the Bridge in one of the Bridge patrol vehicles. You cannot simply walk across it! This Bridge is both a modern marvel and a celebrity, so a vehicle ride is a must for safety.

Feature steering: “Golden Gate Claude”



Feature steering: “Golden Gate Claude”



Hallucinations

- At a high-level, LLM hallucinations can be divided into *factuality hallucinations* and *faithfulness hallucinations*
- *Factuality hallucination* involve a discrepancy between generated content and verifiable real-world facts, i.e., factual inconsistencies.
- *Faithfulness hallucinations* involve a divergence of generated content from user input or a lack of self-consistency within the generated content

Factuality hallucinations

- Huang et al., 2023 divide factuality hallucinations into two further subtypes: *Factual Contradictions* and *Factual Fabrications*
- Factual Contradictions involve situations where the LLM's output contains facts that can be grounded in real-world information, but present contradictions
- Factual Fabrication refer to instances where the LLM's output contains facts that are unverifiable against established real-world knowledge

Factuality hallucinations

Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
	Factual Fabrication	**Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower? **Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement.	The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event.

Faithfulness Hallucinations

- Faithfulness hallucinations can be divided into three subcategories: *Instruction Inconsistency, Context Inconsistency,* and *Logical inconsistency*
- Instruction inconsistency refers to LLM outputs that deviate from a users directive, i.e., unintentional misalignment with non-malicious user instructions.
- *Context inconsistency* involves LLM output that is unfaithful to the user's provided contextual information.
- *Logical inconsistency* occurs when LLM outputs exhibit internal logical contradictions, often observed in reasoning tasks. This manifests as inconsistency both among the reasoning steps themselves and between the steps and the final answer

Faithfulness hallucinations

Faithfulness Hallucination	Instruction Inconsistency	**Instruction:** Translate the English question into Spanish: "What is the capital of France?" **Response:** The capital of France is Paris.	The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.**
	Context Inconsistency	**Retrieved context:** The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea. **Instruction:** Summarize the retrieval document above. **Response:** The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean.	The summary incorrectly states that the Nile originates from the "mountain ranges of central Africa", while the **paragraph mentions that the Nile originates in the Great Lakes region of central Africa.**
	Logical Inconsistency	**Instruction:** Solve the equation 2x + 3 = 11 step by step. **Response:** Step 1: Subtract 3 from both sides to get: 2x = 8. Step 2: Divide both sides by 2 to get: x = 3. So the result of the equation is x=3.	The first step correctly isolates the term with x to get 2x = 8. However, **the next step inaccurately divides 8 by 2 to yield a result of x = 3**, which is inconsistent with the earlier reasoning.

Causes of hallucinations: Data

- **Misinformation and biases**. Neural networks possess an intrinsic tendency to memorize training data and this memorization tendency grows with model size
- Misinformation and biases present within pre-training data and may inadvertently be amplified, manifesting as **imitative falsehood** and the reinforcement of **societal biases**

Causes of hallucinations: Data

Type	Premise	Input	Model Ouput	Explanation
Imitative Falsehood	Within the LLM's training data, a frequently referenced but incorrect fact appears multiple times: *Thomas Edison invented the light bulb.*	Who is credited with the invention of the light bulb?	Thomas Edison is credited with the invention of the light bulb.	**While Edison improved the light bulb's design, he was not its sole inventor.** The model's answer reflects a misconception prevalent in its training data, exemplifying an imitative falsehood.
Societal Biases	In the model's training data, certain names may frequently be associated with specific nationalities.	Provide a summary of the following news: Dr. Kim recently published a breakthrough paper on quantum computing. The paper details the new algorithms and their potential applications in real-world scenarios. It has been hailed as revolutionary by experts.	Dr. Kim from South Korea recently published a paper on quantum computing, detailing new algorithms and their applications.	**The model erroneously added "South Korea" based on the name "Kim"** revealing its bias in linking certain names to specific nationalities, which can cause hallucinations in the summary.

Causes of hallucinations: Knowledge Boundaries

- **Long-tail Knowledge**. The distribution of knowledge within the pre-training corpora is inherently non-uniform, which results in LLMs demonstrating varying levels of proficiency across different types of knowledge
- Recent studies have highlighted a strong correlation between the model's accuracy on general domain questions and the volume of relevant documents
- **Up-to-date Knowledge**. Beyond the shortfall in long-tail knowledge, another intrinsic limitation concerning the knowledge boundaries within LLMs is their constrained capacity for up-to-date knowledge. This has been partially addressed with RAG

Causes of hallucinations: Knowledge Boundaries

- **Copyright-sensitive Knowledge.** Due to licensing restrictions, existing LLMs are legally constrained to training on corpora that are publicly licensed or otherwise available for use without infringing copyright laws

Causes of hallucinations: Data

Type	Input	Model Ouput	Explanation
Long-tail Knowledge	What causes phenylketonuria?	Phenylketonuria is caused by a mutation in the hemoglobin gene, which results in an unusual interaction with hemoglobin, affecting its ability to transport oxygen in the bloodstream.	The correct cause of phenylketonuria is mutations in the PAH gene. **Such specialized medical information is less frequently encountered in the training corpus, making LLMs struggle to memorize.**
Up-to-date Knowledge	Who will be the Democratic Party candidate for the 2024 US presidential election?	Michelle Obama has announced her candidacy and is leading in the polls, with significant endorsements from key Democratic figures such as Barack Obama and Bernie Sanders. She has also outlined a policy platform focusing on healthcare reform and climate change initiatives.	After President Joe Biden announced his withdrawal from the race on July 21, 2024, he endorsed Kamala Harris as the Democratic Party candidate. **Such hallucinations arise because LLMs' parametric knowledge cannot be updated in real-time, leading to potential fabrication regarding current events.**
Copyright-sensitive Knowledge	Describe the ending of the novel "The Secrets of the Alchemist" that was published last year.	In the dramatic conclusion of "The Secrets of the Alchemist", the protagonist discovers the true secret behind the philosopher's stone, leading to a peaceful resolution with all characters sharing in the wisdom.	**The novel "The Secrets of the Alchemist" is under copyright protection, and LLMs have not been trained directly on such copyrighted materials.** Thus, the model's output fabricates details about the book's ending.

Causes of hallucinations: Pretraining

- LLMs can occasionally exhibit unpredictable reasoning hallucinations spanning both long-range and short-range dependencies, which potentially arise from **the limitations of soft attention** where attention becomes diluted across positions as sequence length increases.
- **Exposure bias**, which involves a disparity between training and inference for auto-regressive generative models
- Such inconsistencies can result in hallucinations especially when an erroneous token generated by the model cascades errors throughout the subsequent sequence, akin to a snowball effect

Causes of hallucinations: Pretraining

- **Supervised Fine-tuning**. LLMs have inherent capability boundaries established during pre-training. During SFT, instruction data and corresponding responses are used to unlock these pre-acquired abilities
- Instructions can sometimes exceed the LLM's learned capabilities–**over-fitting on new factual knowledge encourages LLMs to fabricate content**, amplifying the risk of hallucinations
- Traditional SFT methods, which are tuned for helpfulness, typically force models to complete each response, without allowing them to accurately express uncertainty

Causes of hallucinations: Pretraining

- **Hallucination from RLHFs**. Several studies have demonstrated that LLMs encode internal beliefs related to the truthfulness of its generated statements. But misalignment can occasionally arise between these internal beliefs and the generated outputs.
- During RLHF, LLMs can produce outputs that diverge from their internal beliefs. Such behaviors, termed **sycophancy**, underscore the model's inclination to appease human evaluators, often at the cost of truthfulness

Causes of hallucinations: Hallucination from Inference

- **Imperfect Decoding Strategies**. The rationale for stochastic sampling–i.e., incorporating randomness into decoding strategies– stems from the realization that high likelihood sequences often result in surprisingly low-quality text — this is the **likelihood trap**
- But this randomness is also positively correlated with an increased risk of hallucinations
- Higher temperatures increase the likelihood of sampling tokens with lower frequencies from the tail of the distribution, leading to hallucinations

Hallucination Detection: Hallucination from Inference

- **Factuality hallucination detection**: identify factual inaccuracies in the model's outputs
- **Faithfulness hallucination detection**: evaluate the faithfulness of model outputs relative to the contextual information provided

Hallucination Detection:Factuality Hallucination Detection

Fact-checking.

- **Fact extraction**, which involves extracting independent factual statements within the output is followed by **fact verification**, which aims at verifying the correctness of these factual statements against trusted knowledge sources
- **External retrieval**. Check factual inaccuracies by utilizing a collection of external tools dedicated to evidence gathering
- **Internal checking:** In the chain-of-verification approach, an LLM first generates verification questions for a draft response and subsequently leverages its parametric knowledge to assess the consistency of the answer against the original response

Hallucination Detection:Factuality Hallucination Detection

- **Uncertainty Estimation**. Detects hallucinations in zero-resource settings, thus eliminating the need for retrieval
- **LLM internal states**: The internal states of LLMs can serve as informative indicators of their uncertainty— token probability or entropy.
- Uncertainty towards key concepts are quantified by considering the minimal token probability within those concepts.

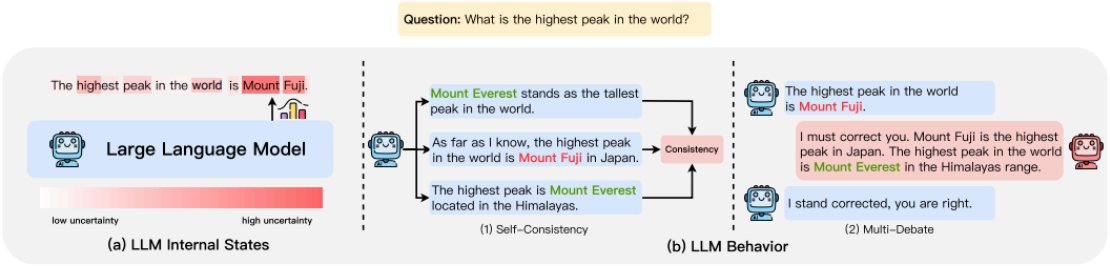
Hallucination Detection:Factuality Hallucination Detection

- **Self-evaluation-based approach**. Estimate uncertainty by testing the LLM's ability to adeptly reconstruct an original concept from its generated explanation is indicative of its proficiency with that concept.
- By initially prompting the model to generate an explanation for a given concept and then employing constrained decoding to have the model recreate the original concept based on its generated explanation, the probability score from the response sequence can serve as a familiarity score for the concept.

Hallucination Detection:Factuality Hallucination Detection

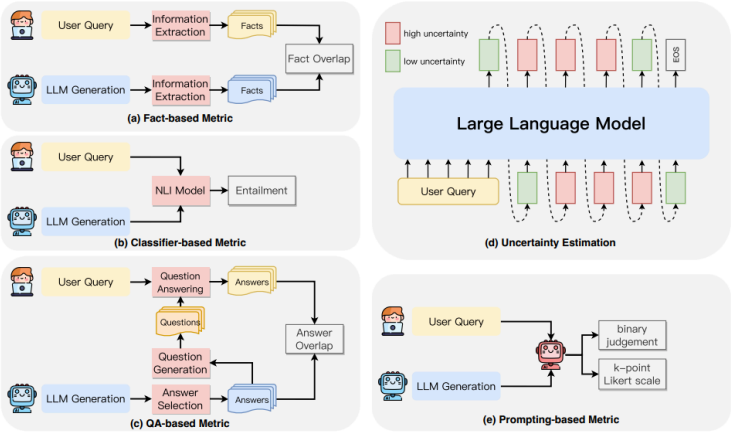
- **LLM behavior**. What if we don't have access to the model weights? Try to elicit behavior from the LLM that allows us to detect hallucinations
- Sample multiple responses from an LLM for the same prompt, and evaluate the consistency among the factual statements.

Hallucination Detection:Factuality Hallucination Detection



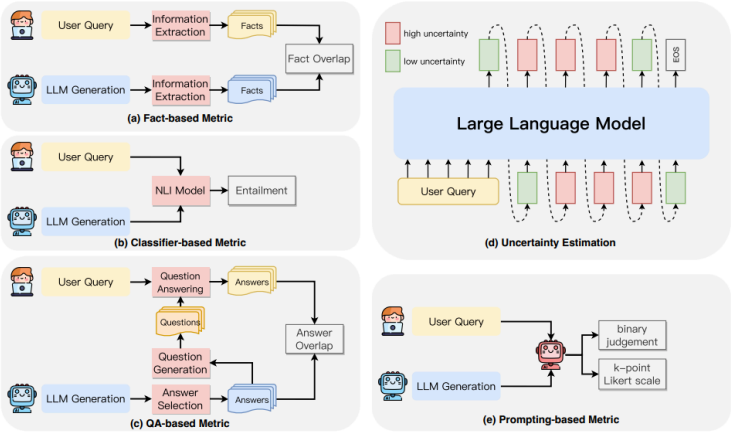
Hallucination Detection:Factuality Hallucination Detection

Fact-based Metrics assesses faithfulness by measuring the overlap of facts between the generated content and the source content



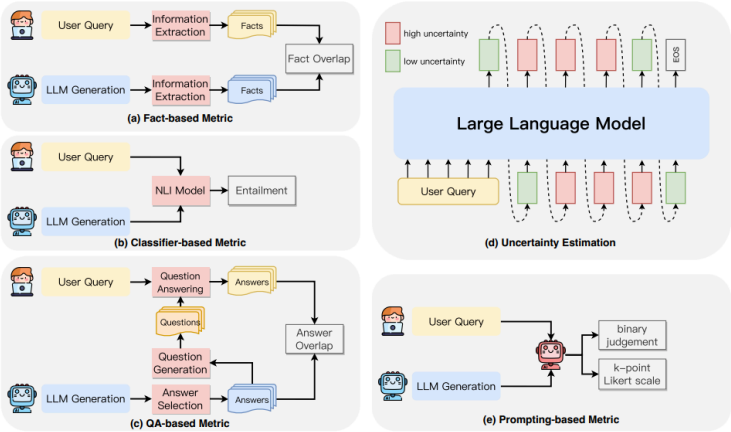
Hallucination Detection:Factuality Hallucination Detection

Classifier-based metrics utilize trained classifiers to distinguish the level of entailment between the generated content and the source content



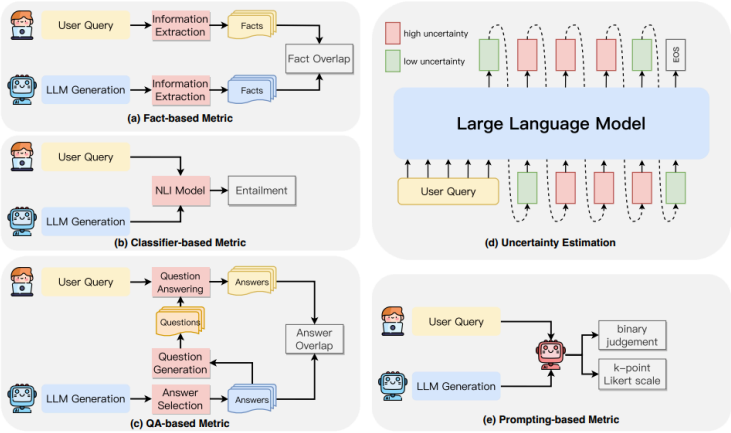
Hallucination Detection:Factuality Hallucination Detection

QA-based metrics employ question-answering systems to validate the consistency of information between the source content and the generated content



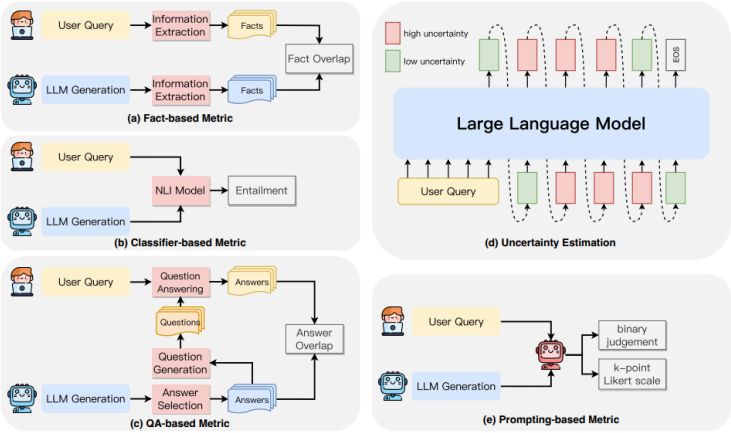
Hallucination Detection:Factuality Hallucination Detection

Uncertainty estimation assesses faithfulness by measuring the models confidence in its generated outputs



Hallucination Detection:Factuality Hallucination Detection

Prompting-based Metrics, wherein LLMs are induced to serve as evaluators, assessing the faithfulness of generated content through specific prompting strategies.



Hallucination Mitigation

- **Data filtering.** Select high-quality data to avoid introducing misinformation and biases
- **Model editing**. Inject up-to-date knowledge by editing models parameters
- **RAG.** Leverage external non-parametric database for knowledge resourcing

Next class: Class 14, Dec 11

Jailbreaking LLMs: Attacks and defenses

Reading

- Extracting Training Data from Large Language Models
- Threats to Pre-trained Language Models: Survey and Taxonomy
- Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations
- NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails
- Large Language Model Unlearning