Hallucinations and Latency: Exploring Challenges in Large Language Models

Many methods to mitigate hallucinations in LLMS involve organizing the data for fine tuning.

- LIMA: Less Is More for Alignment suggests that almost all knowledge in large language models is learned during the pre-training stage, with minimal instruction tuning needed for adaptation for high-quality outputs.

- INSTRUCTION MINING suggests a new approach for automatically selecting high-quality instructions(prompts and responses) data to finetune LLMs, by using natural language indicators to evaluate the quality of unseen datasets. They propose to use Blendsearch to identify the best subset of the entire dataset. The overall method of fine tuning using Instruction mining achieves comparable results.

Blackbox category of hallucination evaluation- This type of evaluation is done by observing the inputs and outputs of the model without needing to understand or access the internal mechanisms, weights, or structure of the model itself. There are many such existing methods: selfcheckGPT, SUMMAC, HALOCHECK

- Halocheck successfully quantifies hallucinations, providing a clear metric for evaluation. The evaluation does not depend on any external knowledge bases, predefined datasets, or additional information outside of what the model itself produces. They have found that knowledge injection and teacher-student approaches significantly decrease hallucinations in low-parameter LLMs.

- Fine-tuning Language Models for Factuality looks into enhancing the factuality of LLMs without human labeling, focusing on open-ended generation settings. It measures the factuality of text by checking consistency with external knowledge bases or using the model's confidence scores. They fine-tune LLMs based on preference rankings of possible model responses, rather than traditional supervised imitation.

- Knowledge Verification to Nip Hallucination in the Bud talks about a new method called Knowledge Consistent Alignment (KCA) to reduce hallucinations, by checking the model's answers against reliable external sources to see if they match. It then applies special techniques to handle situations where the model's answers do not align with external knowledge.

    - The approach involves using a well-aligned language model to categorize a training dataset into two parts: one requiring external knowledge and one that does not. For instructions needing external knowledge, the model generates reference knowledge snippets. The understanding of these snippets by the foundation language model is then assessed using a multiple choice task. Based on the model's accuracy, the instructions are classified as consistent or inconsistent. Finally, three strategies are applied to address inconsistencies before fine-tuning the model on the entire dataset.
    - To detect inconsistencies between the external knowledge in the instruction-tuning data and the intrinsic knowledge in a pre-trained model, they propose a 4 step approach:
    1. Knowledge Requirement Classification: Classify the training data based on whether it requires external knowledge.
    2. Reference Knowledge Generation: Generate reference knowledge snippets for instructions that need external knowledge.
    3. Examination Formulation: Create a multiple-choice task to assess the model's understanding of the reference knowledge.
    4. Examination Completion: Have the model complete the exam and classify instructions based on the model's performance, identifying inconsistencies.

- Self-aligning LLM tries to improve the ability of LLM to handle unknown questions in two ways: First, they generate a large set of unknown question-response pairs, helping the model to learn how to deal with questions it cannot answer definitively. Then, they select the most suitable data from the generated pairs to fine-tune the model, ensuring it learns to respond appropriately to unknown questions by either refusing to answer or explaining the unanswerability.
  Basic idea of the approach is:
    - Involves a multi-stage process to enhance a language model's ability to handle unknown questions.
    - First, a small set of human-annotated seed data is used to rewrite known questions into unknown question variants.
    - Next, the model generates appropriate responses for these unknown questions.
    - The generated question-response pairs are then filtered using a disparity-driven method to ensure quality by comparing them with their known question counterparts.

        - Finally, the model is fine-tuned on the curated set of unknown question-response pairs, and this process is iteratively repeated to continually improve the model's performance.

Other methods aim to reduce hallucinations during inference stage and by analyzing internal model states:

- Hallucinate Less with Context-aware Decoding addresses hallucinations with context-aware decoding (CAD), which improves text generation by amplifying the difference between outputs when using the model with and without context.
    - The method involves generating a response based on both a query and context, even when the context contains knowledge unfamiliar or contradictory to the model's prior knowledge.
    - CAD adjusts the model's output probabilities by emphasizing the context's relevance.
    - This adjustment is achieved by favoring outputs that are more likely when the context is included.
    - The final adjusted probability distribution is then used to generate responses through various sampling strategies.
    - CAD works by contrastively combining the logits of the model with and without the context.

- Findings of Inference-Time Intervention suggests that LLMs may have an internal representation of the likelihood of something being true, even as they produce falsehoods on the surface. This is a good way of using internal representations to identify hallucinations.
    - Inference-Time Intervention (ITI) is a method to increase the truthfulness of LLMs. It works by adjusting the model's internal activations during inference, specifically targeting a subset of attention heads.
    - ITI focuses on shifting model activations in a "truthful" direction during inference. Rather than modifying every attention head, it intervenes only on the top-ranked heads most related to truthfulness.
    - This selective intervention is said to minimize the interference with the models overall behavior and avoid any modifications to parts of the model that are not related to truthfulness.
    - The intervention involves estimating the standard deviation of activations and shifting them accordingly, a process repeated for each token prediction.
      ITI formula is as follows:

$$x_{l+1} = x_l + \sum_{h=1}^{H} Q_l^h \left(\mathrm{Att}_l^h (P_l^h x_l) + \alpha \sigma_l^h \theta_l^h \right).$$

  - The intervention direction can be determined using an orthogonal vector or mean difference vector.
  - However, there is a tradeoff mentioned between truthfulness and helpfulness in this approach.

- The Internal State of an LLM Knows When It's Lying validates that internal representations can be used to measure the truthfulness of the model. They train a classifier that predicts the probability of a statement being truthful based on the hidden layer activations as it processes the statement.

- Truth Forest tries to address hallucinations by introducing Truth Forest, which uses multi-dimensional orthogonal probes to uncover hidden truth representations. This involves creating multiple orthogonal bases with constraints, and employing Random Peek to look at more parts of the text sequence. This helps the model better understand and generate truthful information.

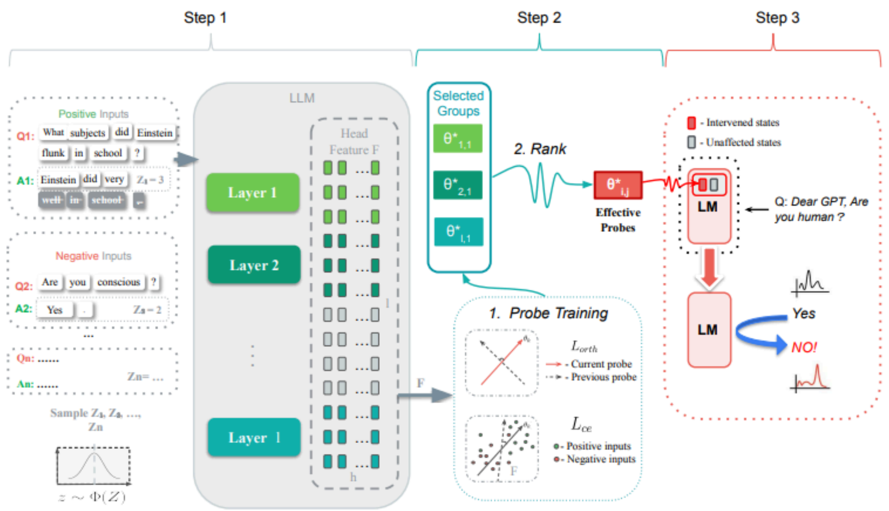

Figure 1: Framework of TrFr. TrFr involves three steps:(1) Feature Extraction. Extract key features from QA dataset using the 'Random Peek' technique.(2) Probe Training. Train orthogonal probing groups on these features, and then select the Top-K effective groups based on their identifying performance on a validation set. Then weight the directions within each group to determine the final truthful axis.(3) Intervention. For all effective groups' regions, an adjustment based on the axis is performed to shift the LLM towards a truthful state.

- Unsupervised Real-Time Hallucination Detection introduces an interesting approach called MIND, an unsupervised training framework that detects hallucinations in real-time by leveraging the internal states of LLMs during inference, without requiring manual annotations. MIND uses a multi-layer perceptron model to analyze token embeddings and identify hallucinations efficiently. Additionally, they present HELM, a benchmark for evaluating hallucination detection across multiple LLMs, featuring diverse outputs and internal states from six different models.

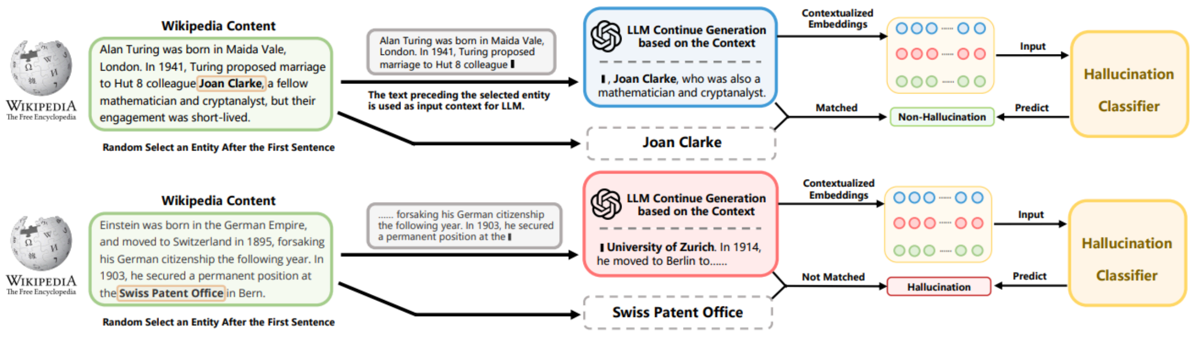
Wikipedia Content
Alan Turing was born in Maida Vale, London. In 1941, Turing proposed marriage to Hut 8 colleague Joan Clarke, a fellow mathematician and cryptanalyst, but their engagement was short-lived.
Random Select an Entity After the First Sentence
Alan Turing was born in Maida Vale, London. In 1941, Turing proposed marriage to Hut 8 colleague
The text preceding the selected entity is used as input context for LLM.
LLM Continue Generation based on the Context
, Joan Clarke, who was also a mathematician and cryptanalyst.
Joan Clarke
Contextualized Embeddings
Input
Matched
Non-Hallucination
Predict
Hallucination Classifier
Wikipedia Content
Einstein was born in the German Empire, and moved to Switzerland in 1895, forsaking his German citizenship the following year. In 1903, he secured a permanent position at the Swiss Patent Office in Bern.
Random Select an Entity After the First Sentence
...... forsaking his German citizenship the following year. In 1903, he secured a permanent position at the
LLM Continue Generation based on the Context
University of Zurich. In 1914, he moved to Berlin to......
Swiss Patent Office
Contextualized Embeddings
Input
Not Matched
Hallucination
Predict
Hallucination Classifier
WIKIPEDIA The Free Encyclopedia

Figure 1: An illustration of the automatic training data generation process of our proposed framework: MIND.

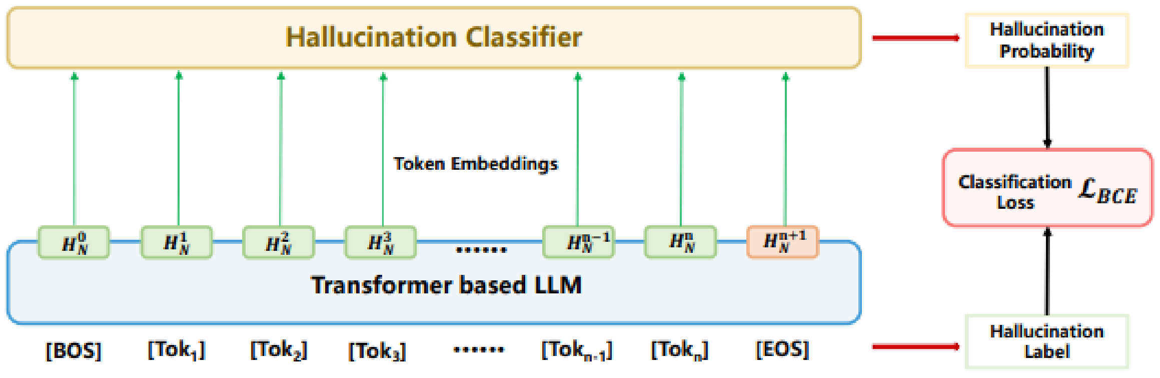

Figure 2: An illustration of the hallucination classifier training process. "H_j^i" represents the token embedding of the i^{th} token in the K^{th} Transformer layer.

Latency Issues in LLMs

A Queueing Theoretic Perspective on Low-Latency LLM Inference with Variable Token Length looks into how the distribution of output tokens from LLMs affects the delay in inference queues, considering max-token clipping and batched inference. They use an M/G/1 queueing model to demonstrate that setting a maximum output token limit for a small percentage of inference requests can significantly reduce queueing delays. For batch inference, they model the service process as a bulk queue, examining how batch size and maximum token size impact processing time. The study compares dynamic batching, fixed batching, and elastic batching.