Achieving Near-Zero Hallucation In Large Language Models

Zoltán Blahovics Bence Nagy Krisztián Nemes-Kovács Balázs Fekete

Table of Contents

Introduction

Methodology

- Experimental investigation in two phases:
  1. Baseline Establishment and Data Quality Analysis
  2. Architectural Intervention and Evaluation

Methodology - Phase 1

- Establishing model architecture and configuration for comparative experiments
  - Canonical transformer model
  - 6 encoder layers, 6 decoder layers, 8 attention heads
  - Standard parameter initialization
  - Adam optimizer with inverse square root scheduler

Methodology - Phase 1

- Creating two large training sets from publicly available datasets in the following steps:
  1. Classifying the data based on empirical reliability into categories:
     - High Reliability - exhibiting high factual consistency and validation
     - Low Reliability - exhibiting high factual heterogeneity and weak source attribution
  2. Based on the classification, formalizing the two distinct corpora:
     - Low-Reliability Corpus ($\mathcal{D}_{\text{LR}}$)
     - High-Reliability Corpus ($\mathcal{D}_{\text{HR}}$)
  3. Strict deduplication and factuality validation for $\mathcal{D}_{\text{HR}}$

Methodology - Phase 1

- Creating and training two baseline models
- Goal: isolate the effect of training data reliability on factual consistency
- Models: Baseline B_1 and Baseline B_2 were setup following to the previously demonstrated configuration
- Training B_1 and B_2 followed the identical training regimen, but used different data:
    - B_1 was trained exclusively on the $\mathcal{D}_{\text{LR}}$
    - B_2 was trained exclusively on the $\mathcal{D}_{\text{HR}}$

Methodology - Phase 1

- For measuring the effect of training data on factual adherence, a robust testing regimen was established.
- Metric: Hallucination Rate defined as the proportion of responses with ≥ 1 factually incorrect claim.
- $\mathcal{T}_{\text{Fact}}$ was created from evaluation prompts sampled equally from the *FEVER* dataset and *TruthfulQA* benchmark.
- $\mathcal{T}_{\text{Fact}}$ is used exclusively for internal benchmarking to measure improvement across development stages.
- Will be reused for measuring the improvement gained by data quality improvement (and for testing the novel architecture in a later step)

Methodology - Phase 2

- Novel architecture proposal
- The Layer-Specific Factual Gate (LSFG)
- Replaced the Standard Feed-Forward Networks (FFN) in the final decoder layers with a novel Gated Factual Network (GFN)
- The gating mechanism enforces selective suppression of activations contributing to factual inconsistency in the terminal layers.
- Formalization:
  $\text{Output} = g \odot \text{FFN}(z), \quad \text{where} \quad g = \sigma(W_g z + b_g)$
- The experimental model M_{LSFG} trained exclusively on $\mathcal{D}_{\text{HR}}$
- The internal benchmarking on $\mathcal{T}_{\text{Fact}}$ showed that both the data quality and the model architectural alterations contributed greatly to the reduction of hallucination rate.

Results

Limitations and Further Research

Bibliography

Kyungjin Yoo and Rajeev Barua. “Recovery of Object Oriented Features from C++ Binaries”. *21st Asia-Pacific Software Engineering Conference*. **Vol. 1. 231–238**, (2014)

Andre Pawlowski et al. “MARX: Uncovering Class Hierarchies in C++ Programs"., *Network and Distributed System Security Symposium* (2017)

Edward J. Schwartz et al. “Using Logic Programming to Recover C++ Classes and Methods from Compiled Executables"., *ACM SIGSAC Conference on Computer and Communications Security*, **426–441**, (2018)

Rukayat Ayomide Erinfolami and Aravind Prakash., “DeClassifier: Class-Inheritance Inference Engine for Optimized C++ Binaries", *arXiv eprints* **arXiv:1901.10073** (2019)