1RT730 HT25 – Assignment 2

Evaluating and Improving LLM Trustworthiness

Due Date: 2025-10-08
Grades: Part A (Mandatory) – 1 point, Part B (Bonus) – 1 point

1. Background

Large Language Models (LLMs) such as GPT, Gemini, and Claude etc. have demonstrated remarkable capabilities in natural language understanding and generation. However, these models are prone to issues that challenge their **trustworthiness**, including generating harmful content, reinforcing societal biases, leaking sensitive information, or being susceptible to adversarial manipulation.

Understanding and improving the trustworthiness of LLMs is a key part of responsible AI development. This assignment aims to give you hands-on experience in **evaluating LLM trustworthy performance** and **developing strategies to improve it**.

2. Datasets

You can choose to work with one of the two recent benchmark suites discussed below for LLM trustworthiness evaluation:

2.1 DecodingTrust

- Scope: Comprehensive evaluation of trust dimensions: toxicity, stereotype bias, robustness, privacy risks, machine ethics, and fairness.
- Purpose: Reveals vulnerabilities in LLMs under realistic and adversarial conditions.
- Reference: DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models (NeurIPS 2023) (https://decodingtrust.github.io/)
- Usage: The dataset is available on HuggingFace under Dataset: AI-Secure/DecodingTrust, or in the GitHub repo AI-secure/DecodingTrust/data. Alternatively, you can also install `decoding-trust` following the instructions in the GitHub repo.

2.2 TrustLLM

- Scope: Comprehensive study of trustworthiness in LLMs across eight dimensions: truthfulness, safety, fairness, robustness, privacy, machine ethics, transparency, and accountability; across 30+ datasets.

- Purpose: Offers a broader evaluation and additional tools for automated assessment.

- Reference: TrustLLM: Trustworthiness Benchmark for Large Language Models (ICML 2024) (https://trustllmbenchmark.github.io/TrustLLM-Website/)

- Usage: TrustLLM dataset is available both on HuggingFace (Dataset: TrustLLM/TrustLLM-dataset) and GitHub (HowieHwong/TrustLLM/dataset). Note that data splits under each subset may have different column names, so you need to load each data file separately (i.e. explicitly using argument like `data_files="ethics/explicit_moralchoice.json"` while loading the dataset), or download the entire zip file and load each data split manually.

3. Assignment Tasks

3.1 Part A (Mandatory) – Evaluation of LLM trustworthiness

Objective: Choose an LLM to evaluate different aspects of trustworthiness of the LLM using benchmark datasets from DecodingTrust/TrustLLM.

Steps:

1. Model Selection:
    - Choose an LLM (either open-sources models such as Mistral / Qwen, or you can use your own API for Gemini / GPT) of manageable size to run on available hardware.
2. Deployment:
    - Set up the model locally or on a cloud platform.
    - Freely available computational resources (e.g., Google Colab, Kaggle) can be used.
3. Load data:
    - Select a benchmark dataset from DecodingTrust/TrustLLM, **choose at least 3 trust dimensions** among the scopes covered by the dataset.
    - Understand the data structure, then load the corresponding subsets (of the dimensions you choose) and the data splits within the subsets.
    - Analyze the content in each column of each data split/subset, extract the useful data for next steps (e.g. prompts to be used, measurement variables, and etc.)
4. Evaluation:
    - Test the LLM model on selected subsets.
    - Collect quantitative scores and qualitative examples of failures.
5. Report:
    - Summarize the evaluation process, key results, and ethical implications.
    - Provide specific example prompts and outputs illustrating ethical risks.
    - Visualize your quantitative results (e.g. using bar plot, histogram, boxplot, and so on).
    - Conclude and discuss your findings.

3.2 Part B (Bonus) – Improving Trustworthiness

Objective: Develop and test at least two practical mitigation strategies aimed at improving model performance in a specific trust dimension.

Possible strategies include (but are not limited to):

- Safety/Toxicity: Output filtering, improved system prompts.
- Bias/Fairness: Counterfactual data augmentation, debiasing templates.
- Privacy: Named-entity scrubbing, refusal policies.
- Robustness: Adversarial prompt detection, clarification turns.
- Transparency: Self-critique mode, citation enforcement.
- General: Fine-tunings, RLHF/RLAIF, Mixture-of-Experts, Re-ranking RAG.

Steps:

1. Choose the trust dimensions you want to improve (e.g., safety, bias, privacy).
2. Implement mitigation methods for the chosen dimensions.
3. Re-evaluate the model using the same DecodingTrust/TrustLLM subset.
4. Compare pre- and post-mitigation results and discuss performances and trade-offs.

4. Submission

You should submit a single notebook / PDF file in Studium, containing the following contents:

1. Technical Report
    - Introduction: Motivation and relevance of trustworthiness in LLMs. Mention about the benchmark selected.
    - Model & Setup: Model description, dataset subsets, and the three trust dimensions.
    - Results – Part A (mandatory): Benchmark dataset evaluation results, with metrics plots, tables, qualitative examples, etc.
    - Results – Part B (Bonus, if attempted): Motivation of selected mitigation methods, implementation details, post-mitigation evaluation and comparison.
    - Discussion: Ethical implications, observed limitations, and suggestions for future work, if any.
    - References for all tools, datasets, and key literature used.
2. Codes
    - All codes must be self-explainable with proper comments.

P.S. If you work with a .ipynb notebook then the report should be integrated as markdown cells (no need to write everything in one cell, can be splitted as several parts). The outputs of all executable cells must be included, **missing output leads to incomplete submission**. While if you use python scripts, then you should write a standard report in a PDF format and include all your codes in a link (GitHub/HuggingFace repo or similar platform), or attach all of them in the report as code blocks. Essential outputs from your scripts should also be included in the PDF in order to support your implementation.

5. Grading Criteria

- Model Deployment & Functionality: Model runs correctly; chatbot interface functional; clean code.
- Evaluation on benchmark dataset: Correct implementation; coverage of ≥3 trust dimensions; clear metrics, plots, examples, and analysis.
- Report Quality: Clarity, structure, analysis depth, connection to ethical implications.
- Bonus Part – Mitigation Implementation & Analysis: Soundness of chosen method; correct implementation; measurable impact; trade-off analysis and discussion.

Important: Part B has to be submitted before the specified deadline in order to be considered for a higher grade.

6. Submission Instructions

- **Deadline**: 2025-10-08, 23:59 CEST

- **Format**: Submit a notebook/PDF file containing your report together with codes.

- **Submission Portal**: Studium–Assignments–Hand-in Assignment 2 (A2).

7. Academic Integrity

Collaboration in understanding concepts is encouraged; however, all code, evaluation, and report writing must be on your own efforts. External code or datasets must be properly cited.