Large language model

PROGRESS REPORT
23/01/2025

Overview

1. Hallucination detection
    1. LLM-as-a-judge method
    2. Classifier overlay

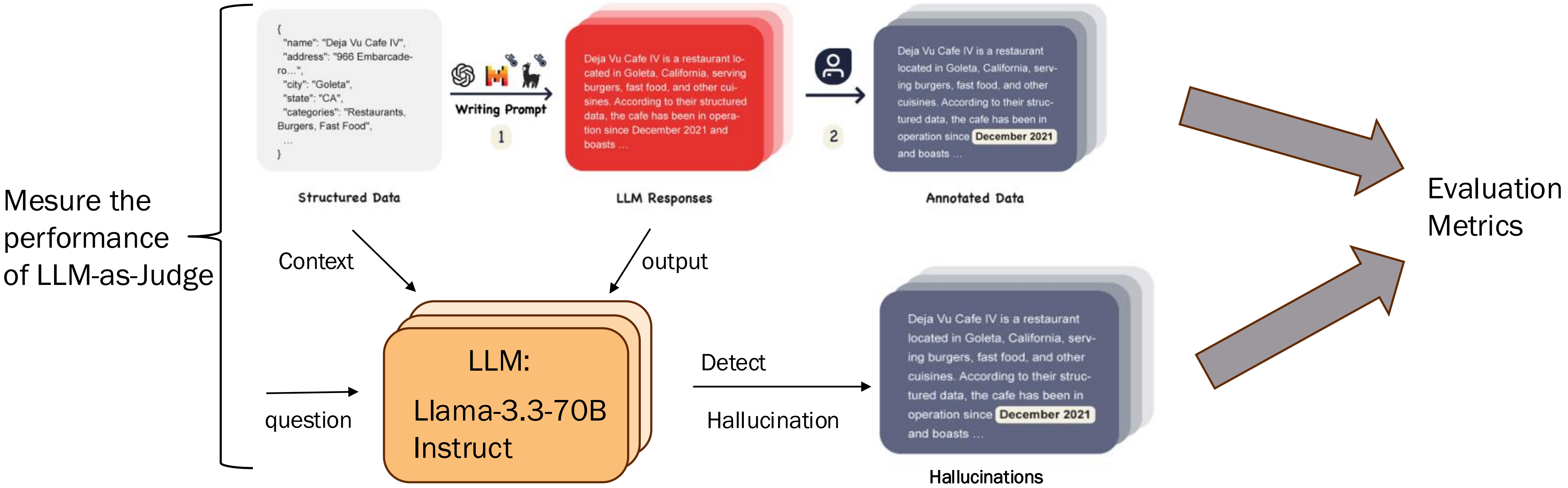
Hallucination detection

- Hallucination VS tasks:
  - QuestionAnswering
  - Data-to-textWriting
  - Summarization
- Hallucination VS models:

6 models implemented in the paper

- GPT-3.5-turbo-0613 and GPT-4-0613 from OpenAI
- GPT-3.5-turbo-0613 and GPT-4-0613 from OpenAI
- Llama-2 7B-chat, Llama-2-13B-chat and Llama-2-70B-chat from Meta

Hallucination detection



Hallucination detection

- Evaluation Metrics:
  - <u>Faithfullness</u> = $\frac{\#\,true\ statements}{\#statements}$ --> concentrate on true statements != paper approach (detect halluciantion)
  - <u>Answer relevancy</u>
    - Generate questions q_i based on the provided RAG answer.
    - $AR = \frac{1}{n}\sum_{i=1}^{n} \text{cosine\_}sim(q_i, q)$

  --> not at all a good metric (no consideration of context)

  - <u>Response-level Detection</u>

    Accuracy, **precision**, **recall**, **F1 score** for each detection algorithm and its variants across different tasks

    --> Sample based approach (detect if the overall sentence contains halucination)

  - <u>Span-level Detection</u>

    overlap between detected span and human-labeled span and report the precision,recall,and f1score

  --> Word level evaluation

Results

Task	Nb samples	Accuracy	Precision	Recall	F1 score
Overall performance	895	0.817	0.773	0.872	0.819
QA	295	0.792	0.727	0.800	0.762
Summary	300	0.815	0.786	0.846	0.815
Data2Text	300	0.839	0.789	0.938	0.857

- We can achive better performance by **finetuning** the LLM as a judge

--> Halucination will be detected easily for general data

Next steps

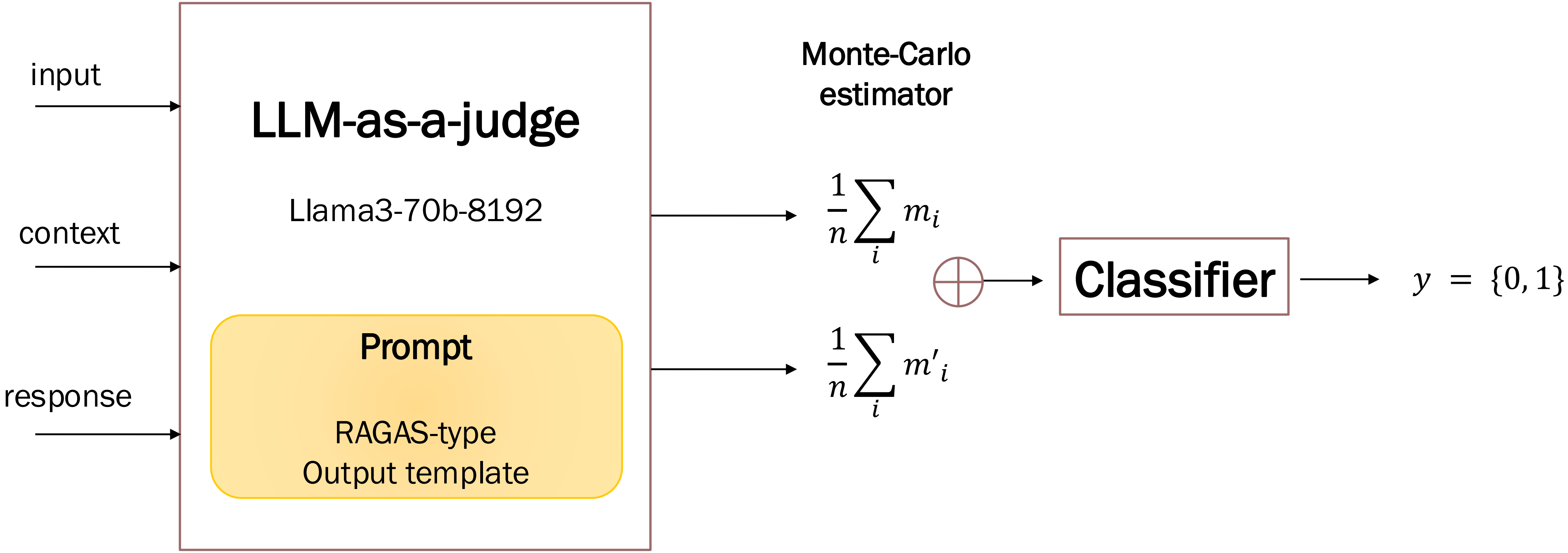
- LLM as a judge
  - Test the performance of the model to detect hallucination for **Fianancial data**

  --> Find labeled financial dataset
  - Test the performance of the model based on **other Metrics : Completeness , Toxicity**

- **Final Evaluation**
  - Weighted sum of different evaluation metrics

  --> find weights

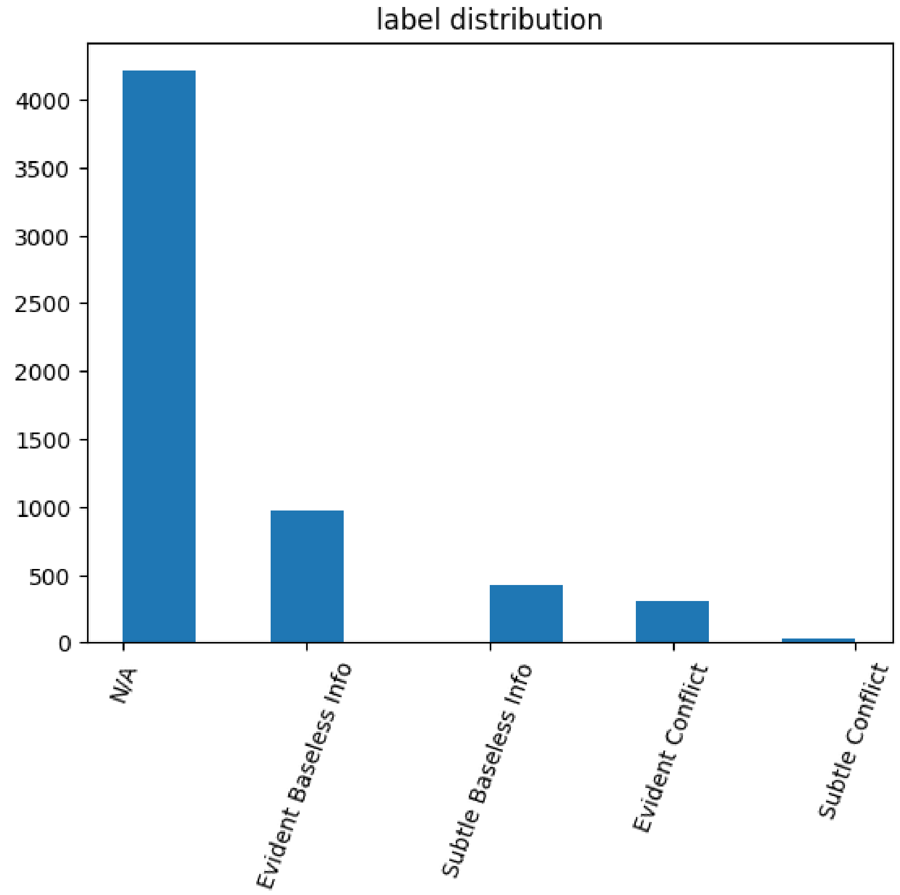
Classifier model



Metric bundles

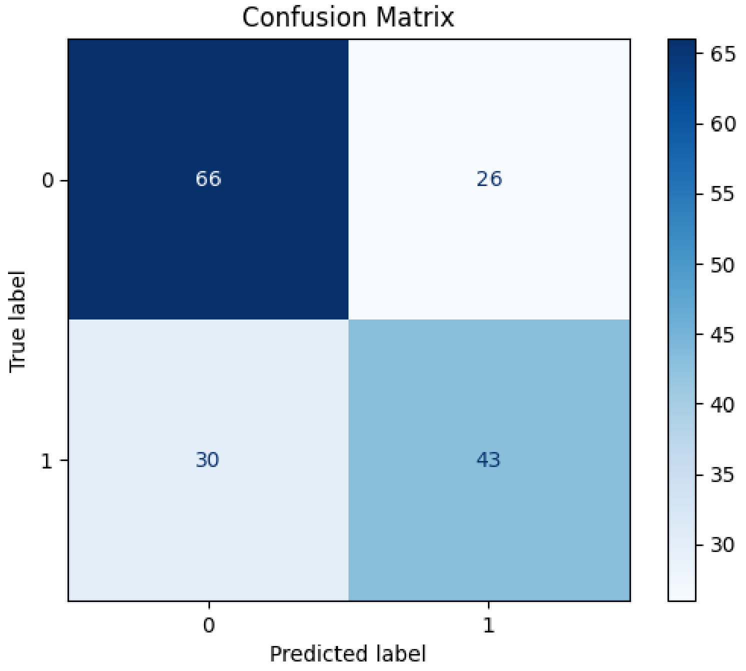
Binary criterions	LLM Metrics
Hallucination	Faithfulness (RAGAS def.) Answer relevancy (RAGAS def.) Temperature
Completeness	Contextual recall/precision Negative acceptance (GroUSE def.) Positive refusal (GroUSE def.)
Toxicity	IBM metrics [https://arxiv.org/html/2403.06009v1]

RAGTruth dataset



- QA samples
  - Same inputs but with different models can have different results
- Keep only binary classification.
- Use some rebalancing of data

Results



FFNN - $input\_size \rightarrow 6 \rightarrow 4 \rightarrow output\_size$

Classifier	ρ	τ	Accuracy
FFNN	0.36	0.31	0.66
GradBoost	0.04	0.02	0.53

- Underwhelming results overall
- The "balanced" dataset has made the model has a bias towards detecting hallucination.
- Hints towards the fact that the features are not the best for the problem.

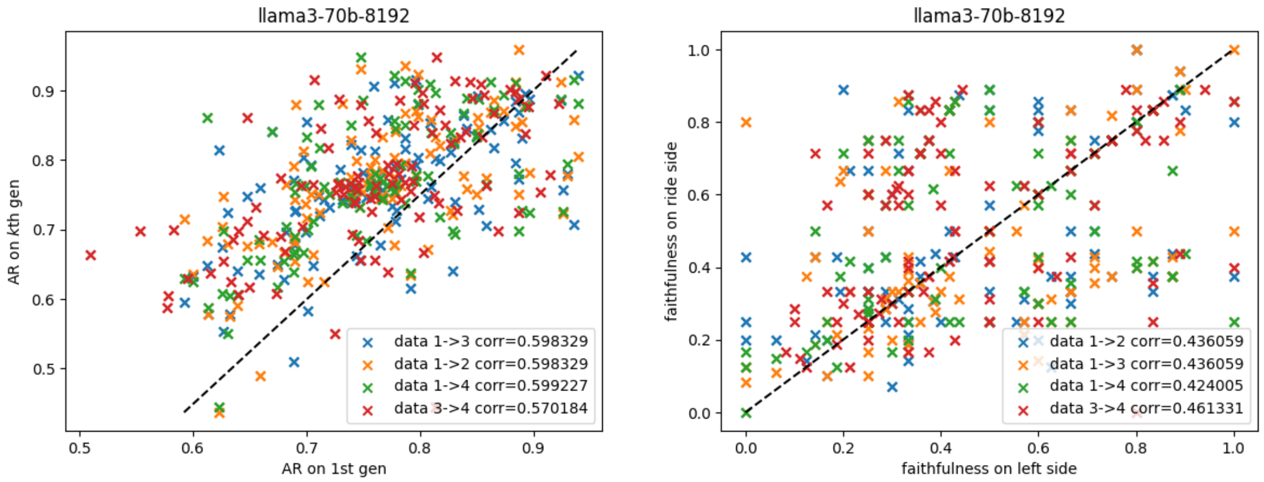
Robustness & improvements

- Averaged metric is necessary as robustness is not obvious for the method
- In theory, $\lim_{n\to\infty}\frac{1}{n}(\sum_i m_i) - m = 0.$

$\Longrightarrow$ Greater n should be better in practice.

$\Longrightarrow$ Other improvements

- Prompt engineering
  - GroUSE templates
  - CoT [ChainPoll arxiv.org:2310.18344]
  - Self-consistency
- More metrics may be necessary by classifier



Plot of different metric generation of faithfulness for the same LLM input
(Each gen is plotted against each other)