Evaluating Large Language Models (LLM) and Ensuring Quality Applications

Speaker : Galina Naydenova

Speaker Introduction: Galina Naydenova

Freelance Machine Learning Engineer

Impact Start-ups, NGOs, Educational Institutions

Bulgaria  -> U  -> 

Japan

NEC

FORRESTER®

Work in Tech and Research

The Open University

Data Science Manager, OU, UK

Higher Education Academy

Learning Analytics HEA Fellow, UK

2020 Freelance **Machine Learning Engineer**

Taught Data Science at Le Wagon Tokyo

AI for Social Good

- Lead Machine Learning Engineer
- Product Owner, Mentor
- Leader of **Omdena Japan Chapter**

Building AI For Good, For the People, By the People

Omdena Japan

Galina Naydenova | LinkedIn

Evaluating Large Language Models and Ensuring Quality Applications

Content:

1. Why Testing and Evaluating LLMs is Essential
2. Typical issues for LLMs and aspects of quality LLM output
3. Ways to evaluate LLMs
4. LLM Benchmarks
5. Ways to automate LLM evaluation
6. Business specific cases – example from practice

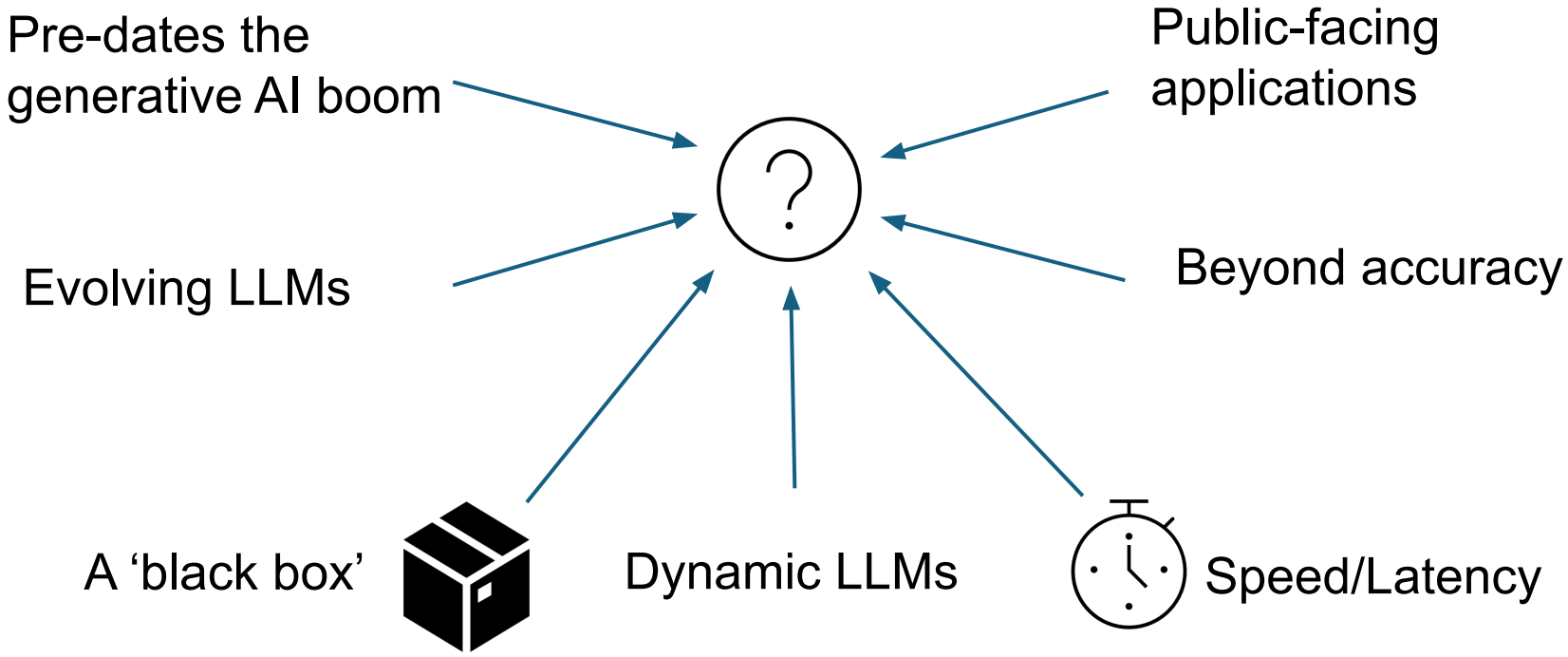
Large Language Model (LLM) ⟷ Generative AI

Testing ⟷ Evaluation

Why is Essential to Evaluate Large Language Models

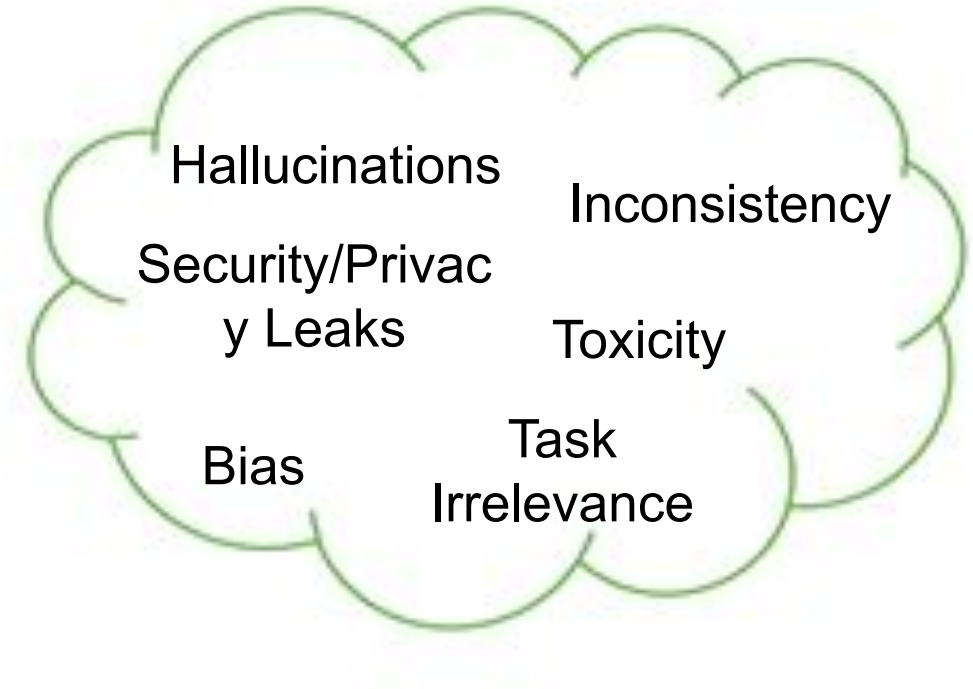
Testing and Evaluation for LLM

Why needed?

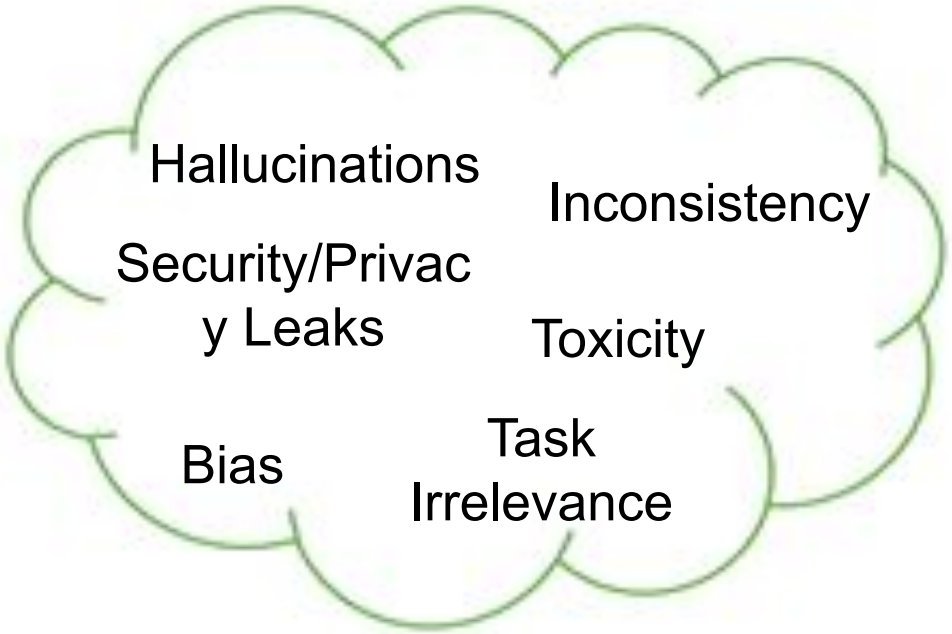


All this makes continuous and scalable evaluation of LLM an imperative!

Typical Issues with Large Language Models



Metrics to the Rescue!



- ❑ Coherence
- ❑ Fluency
- ❑ Relevance
- ❑ Fairness
- ❑ Relevance
- ❑ Interpretability

LLM Output – Too Many Moving Parts

Hallucinations

Inconsistency

Security/Privacy Leaks

Toxicity

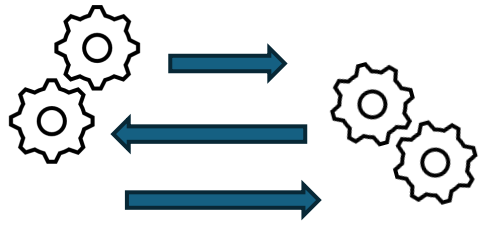
Bias

Task Irrelevance

- ❑ Coherence
- ❑ Fluency
- ❑ Relevance
- ❑ Fairness
- ❑ Relevance
- ❑ Interpretability

LLM

LLM / API parameters



Prompts

Fine-tuning

RAG Architecture

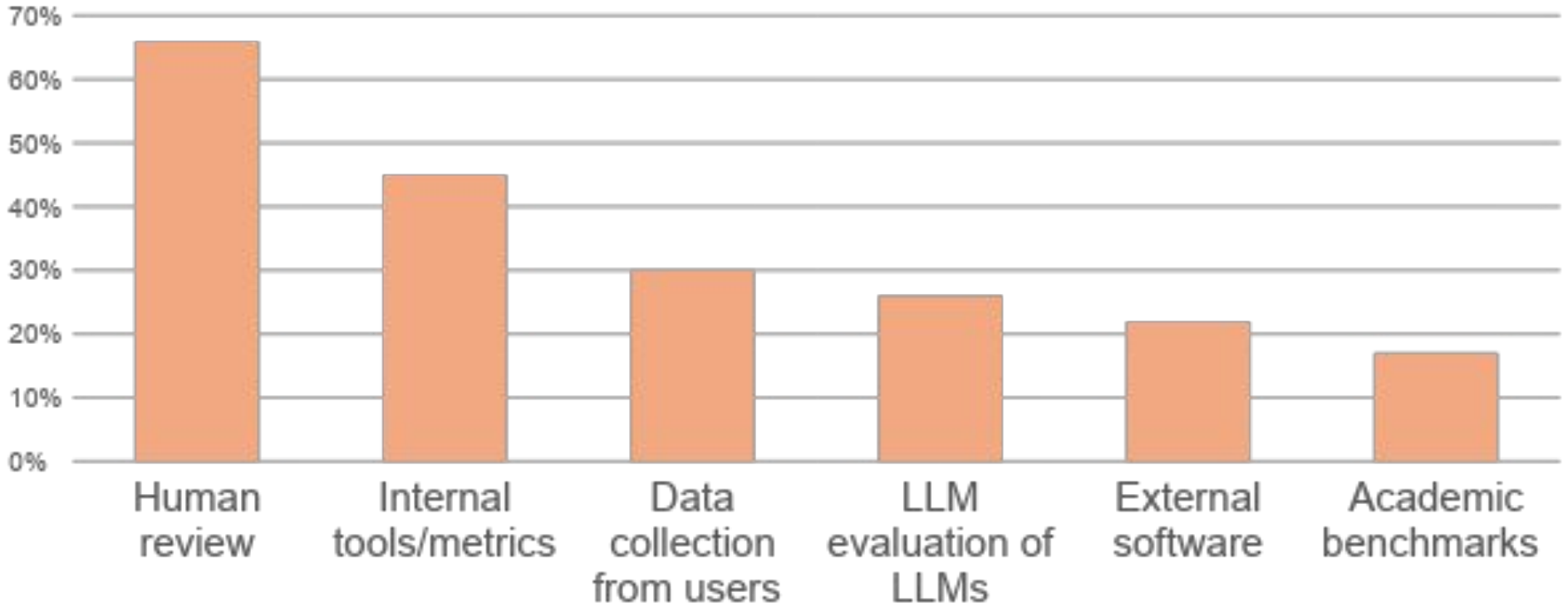
+

Speed / Latency

Speech to Text Interface

How are Businesses Evaluating LLMs

- Predominantly manually - complex LLM output, multiple aspects of quality.
- Can be adapted to the task, but time-consuming and subjective.
- Benchmarks and benchmark-based external tools only give part of the picture.

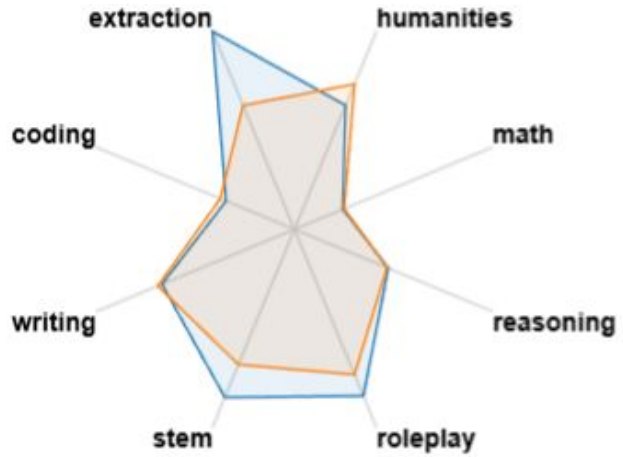


2023: The State of Generative AI in the Enterprise - Menlo Ventures (menlovc.com)

LLM Benchmarks

- For selecting the base LLM
- Early LLM benchmarks - accuracy focus (Word Error Rate, perplexity).
- Task-specific – e.g. **BLEU** for machine translation, **ROUGE** for summary generation

- **Multi-task** evaluations - publicly available, large-scale, standard
- **GLUE** – diverse tasks reflecting general Language Understandi
- **JGLUE** (Japanese General Language Understanding Evaluatio
- **LLM Leaderboards** – e.g. Nejumi for Japanese (**W&B**)



- **External software** – structured/scalable metrics use. (e.g. **W&B**)
- **Open Source datasets** –E.g. **HuggingFace** - built-in task-specific (BLEU, ROUGE), custom metrics.
- **Benchmarks Issues**: Disconnected from business purpose and production performance.

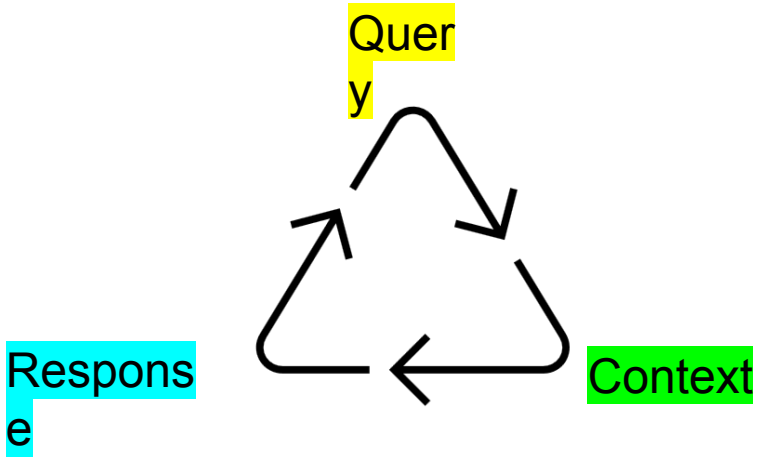
Automating Evaluation – LLMs Evaluating LLMs

- **Holistic** evaluation: beyond the language elements, can look into relevance and clarity.
- **Issues**: Increases complexity. Who evaluates the evaluating LLM?
- **Examples**: LangSmith, RAGAS, Trulens

- **Trulens** - GitHub - truera/trulens: Evaluation and Tracking for LLM Experiments

- experiment and results logging, **Feedback** module, built-in functions (e.g. Criminality, Hate)

- **RAG applications** –interaction between Query, Context, and Response



- Query and Context – **Context Relevance** - Is the retrieved context (documents) relevant to the query?
- Context and Response - **Groundedness** – Is the response supported by evidence in the context?
- Query and Response - **Answer Relevance** – ultimately, is the Response relevant to the question?

Evaluating LLM Output: Scenario

Task: need to test and evaluate whether a **teaching chatbot app** produces **quality output**

1. Define **user personas** and use cases
2. Define what is **quality output** for all these cases
3. What **features** will ensure quality output – LLM responses and beyond - voice, captions, transcripts, content filtering, latency, interface
4. Collect **test cases** - (processes, inputs and outputs). Example questions: “What are the possible outcomes?”, "How else can a user use the feature?“. Test unexpected or ‘null’ inputs
5. Consider **integrations** –parts of the system (app, cloud provider, API, file system, database).
6. **Performance** – response time, load testing, cloud resources geo-location

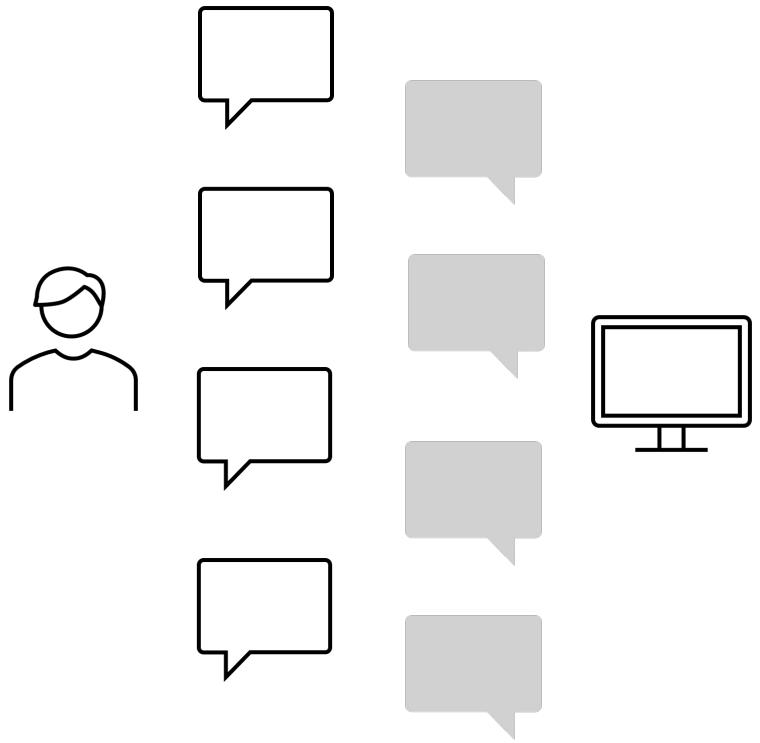
Defining Quality Criteria

Task: Defining what is **quality output** for the particular **domain**

My app: A **generative dialogue** model

Explore: **Quality Assurance of Generative Dialog Models**

(e.g. arXiv:2203.15414v1 [cs.SE] 29 Mar 2022)



Example criteria:

- to have consistent persona,
- maintain one topic at the time,
- remember user-provided information,
- be robust against typing mistakes and word order, punctuation

+ Add **own** criteria: e.g. to adapt to language level, to set response token limit

Preparing your Toolbox

Question: what systems/procedures do I need to have to be able to evaluate my LLM app

Testing and logging results:

- Prompt, model and prompt parameters, responses.
- Template dialogues (consecutive questions and answers).
- Negative testing - deliberate mistakes and ‘provocations’

Prepare your toolbox:

- **References**: Desirable/undesirable outputs
- Make the most of the **LLM-provided** features: API responses carry additional information
- **Cloud provider** features – e.g. MSFT Azure OpenAI content controls, query to context similarity metric and document references for RAG.

! Make the most of your **Humans:** domain experts, users. They know what works best!

Conclusions

- The need to measure LLM performance is not new
- Quality LLM output has many facets, making evaluation complex
- LLM Benchmarks help select your base model
- There are ways to automate, including using LLM to evaluate LLM output
- Define your criteria and ‘non-negotiable’ requirements
- Make the most your model and cloud provider features
- You will still need human input, including from users
- Be ready to repeat, repeat, repeat

THANK YOU !