J-WATR-BUFFALO

Boundless User-focused Framework for Advanced LLM Optimization

Jayanth, Will, Advit, Tarun, and Raunak
Cisco Research

08/11/23

BUFFALO

The Buffalo Spirit brings strength, stability, gratitude and abundance. Buffaloes are nomadic and in their wanderings they trust that the Earth will sustain them and they will find bounty on their path.

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

Limitations of the LLM Landscape

Why can't we deploy GPT-4, LLAMA-2 for everything?

Data Privacy
- Sharing PII, divulging customer/enterprise data

Expenses
- Cost/Power, usage limitation, redundant queries

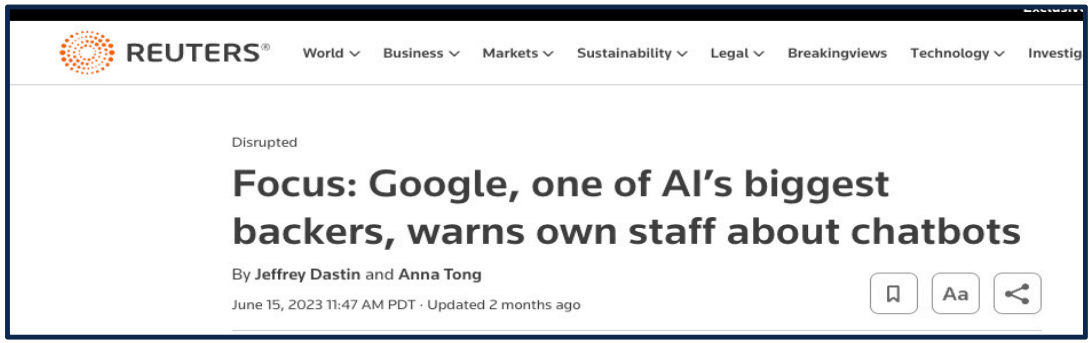
Data Security
- Leaking confidential data, unauthorized access

Hallucinations
- Misleading information, improper tool usage

Explainability
- Robustness, removal of bias, truthfulness

Competence
- Lack of reasoning, attribution, false confidence

Issues Across Enterprises



REUTERS® World Business Markets Sustainability Legal Breakingviews Technology Investig

Disrupted

Focus: Google, one of AI's biggest backers, warns own staff about chatbots

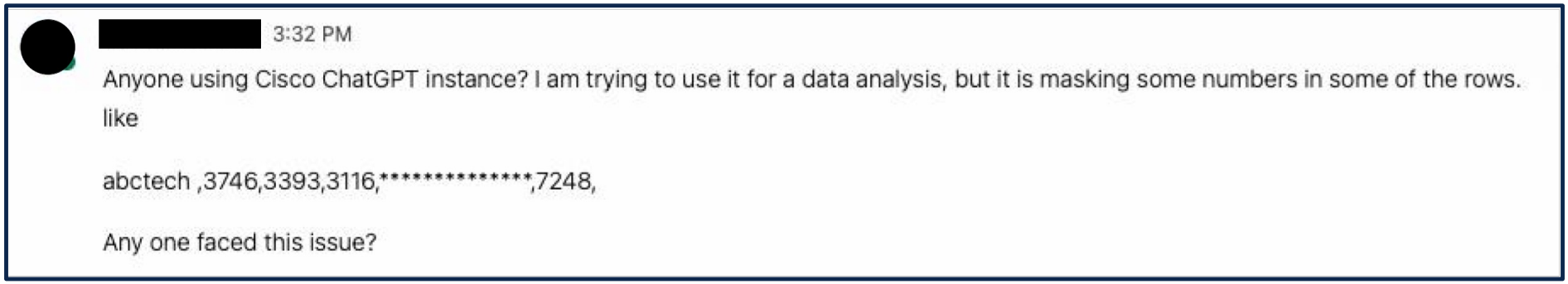
By **Jeffrey Dastin** and **Anna Tong**

June 15, 2023 11:47 AM PDT · Updated 2 months ago

Tech Monitor

https://techmonitor.ai › technology › ai-and-automation

Samsung bans staff from using ChatGPT after data leak



[redacted] 3:32 PM

Anyone using Cisco ChatGPT instance? I am trying to use it for a data analysis, but it is masking some numbers in some of the rows. like

abctech ,3746,3393,3116,**********,7248,

Any one faced this issue?

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

Solution Architecture

An Enterprise LLM Gateway, J-WATR-BUFFALO

Gateway Features

Prompt Layer

- PII Redaction, Intent Classification
- Prompt Translation, Cost Optimization
- Defense against Prompt Injection Attacks

LLM Cache

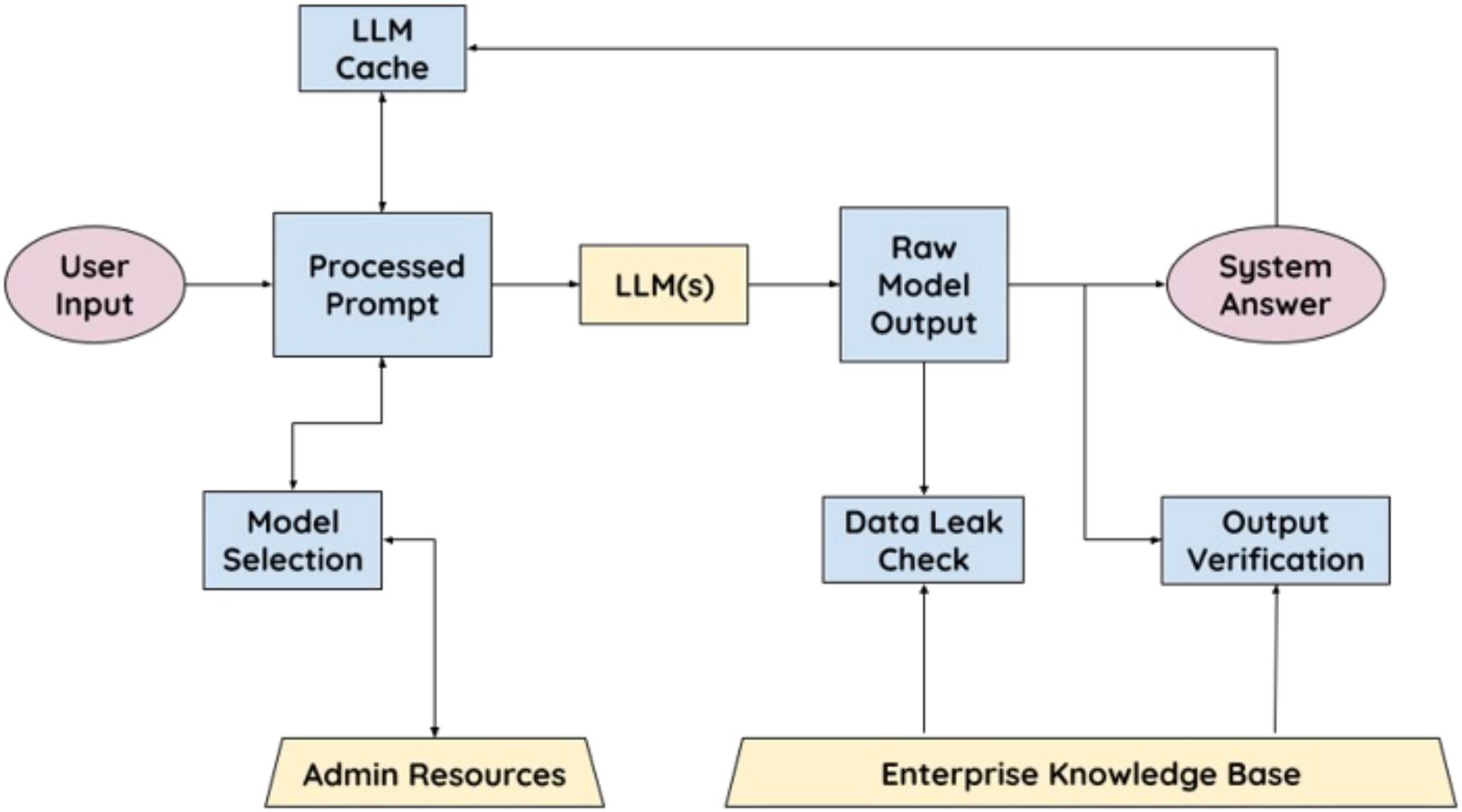
- Improving Efficiency
- Across enterprise, reducing cost
- Hierarchical, self-improving

Data Exfiltration

- ACL for different information
- Detection of leakage of sensitive info
- Permission-variance per user

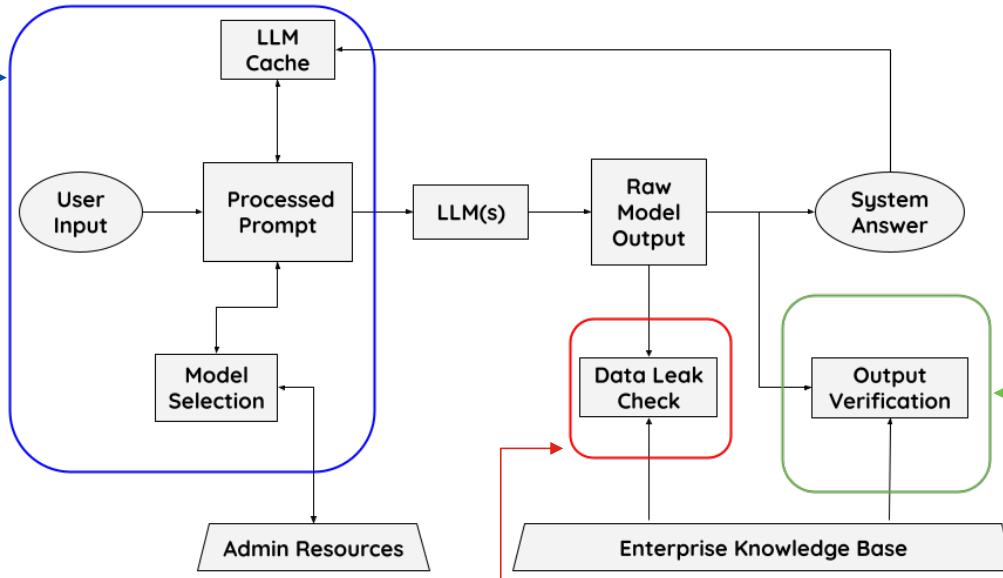
Output Verification

- Retrieval Augmented Correction
- Knowledge Graph comparison
- Truthfulness score



Sponsored Research

- Black Box Optimization - Aditya Grover
- Cascaded QA Model for Efficient Inference - Eunsol Choi
- Knowledge-Grounded and Time-Evolving Conversational Recommendation Systems - Julian McAuley
- Knowledge Extraction and Discovery from Massive Texts via Extremely Weak Supervision - Jingbo Shang



- Event Linking and Event Based Population - Violet Peng
- Customizing Robust and Controllable Text Processing Models - Kai-Wei Chang
- Grounding Language Models to Real-World Environments - Yu Su
- Fact Verification via Combining Knowledge Graph Reasoning and LLM - Yizhou Sun

In-house research – Prompt, Cache, Exfiltration, Verifcation

Gateway Highlights

Composability

- Modular, add/subtract features
- Admin configurable, monitoring

Online learning

- Improve functionality over time

Optimizable

- In speed, cost, and accuracy

Transparency

- Open source
- Users can see I/O of each layer

Security

- Prompt injection prevention
- Data exfiltration safeguard

Reliability

- Hallucination detection

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

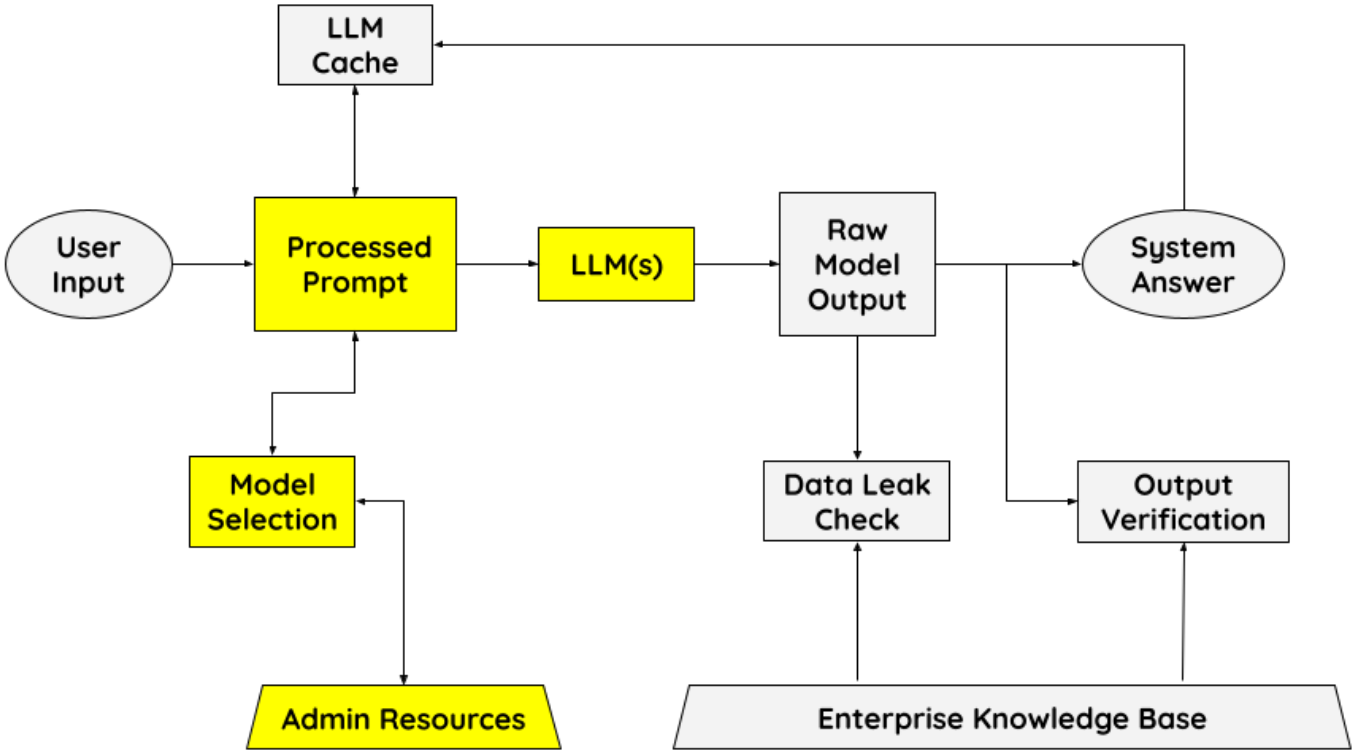
Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

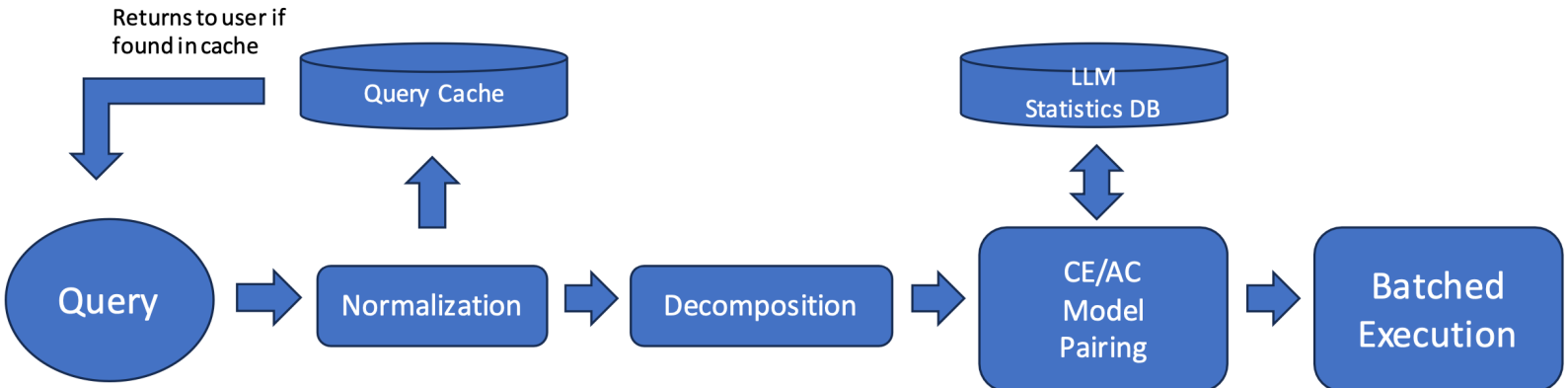
Prompt Layer



Prompt Layer Problem Statement

- Controlling and managing large language models (LLMs) usage is tough in enterprises.
- Understanding and handling how users interact with LLMs is also important due to the rising demand for LLM usage.
- Navigating cloud comes with its own challenges like complex pricing models and risk of handling sensitive data.
- To deal with user questions and personal data, we need a simple, safe, and cost-effective way!

Flow for a Composable Query Optimization Engine for cost effective querying from a list of LLMs



Normalizes the query

1. Cleans and preprocesses
2. Embeds the meaning
3. Compresses to minimal tokens

Decomposes the query into logical/functional parts

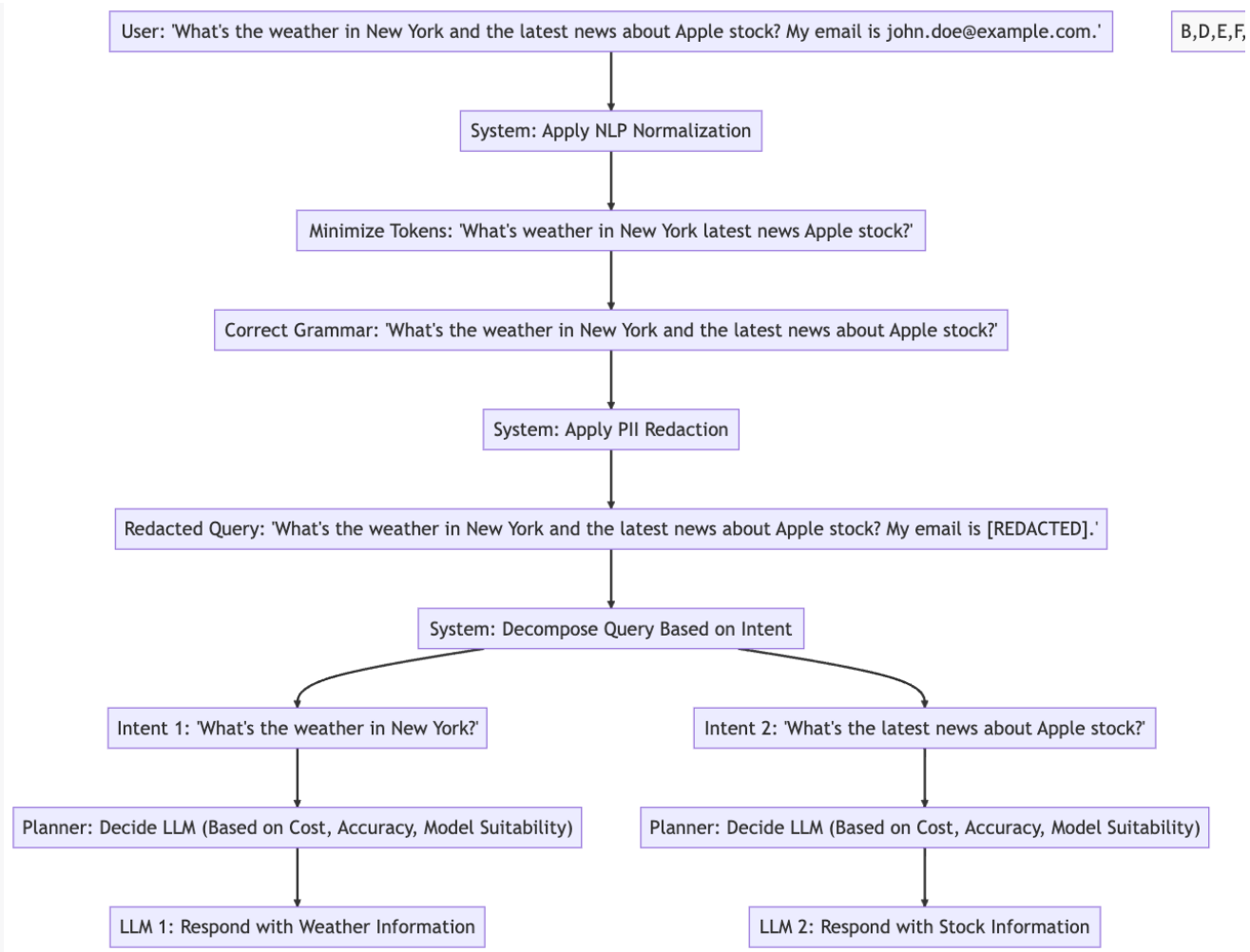
Pairs the specific part of the query with the LLM best suited to respond.

Operates in two modes – cost effectiveness, or, accuracy centric.

Returns to the user costs for various execution plans.

After the user selects one plan, executes it in a batch in parallel *

*to the extent possible. Else creates a topological graph.



Prompt Layer Technical Breakdown

- First part is Query Normalization and PII Redaction.
  - Query Normalization with NLP - Tokenization, Stop-word removal, Lowercasing, Synonym expansion, Regularization, Part-of-speech tagging, N-gram extraction, Temporal Cognitive Resetting, Phonetic normalization, Spelling correction, Phrase segmentation.
  - PII Redaction with NLP - Named Entity Recognition, Pattern Matching, Dictionary Based Replacements, Ontology Based Redaction, Rule based Methods.
  - Experimenting with a subset of above for future release of J-WATR-BUFFALO.
  - Current release utilizes basic NLTK methods to minimize tokens without sacrificing meaning

- Second part is Query Decomposition
  - Query decomposition with NLP – Semantic Parsing, Intent Recognition, Dependency Parsing, Sub-Query Generation.
  - Currently do this using standard NLTK tools
  - Experimenting with these and better tools
  - Very interested in this as future work

- Third (and final) part is Model Pairing.
  - Decomposed queries matched to models per user specifications
  - Mostly heuristic (requires approximate algorithms): cost/accuracy objectives defined by user choice
  - Optimization based on internal DB of LLM specs
  - Simplified use of cost and accuracy metrics for model-query pairings
  - Future potential: defer to ML model with more data for decisions

Conclusions/Future Work

- **Prompt Layer**: Essential for user experience, expenditure, security with cloud-based LLMs
- **NLP Techniques Exploration**: Including new SoTAs; focus on intent recognition
- **Bright Long-Term Scenario**: More LLMs and enterprise users incoming!
  - Expand to Multimodal Interactions
  - Develop Personalized User Profiles
  - Enable Real-Time Collaboration & Sharing

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

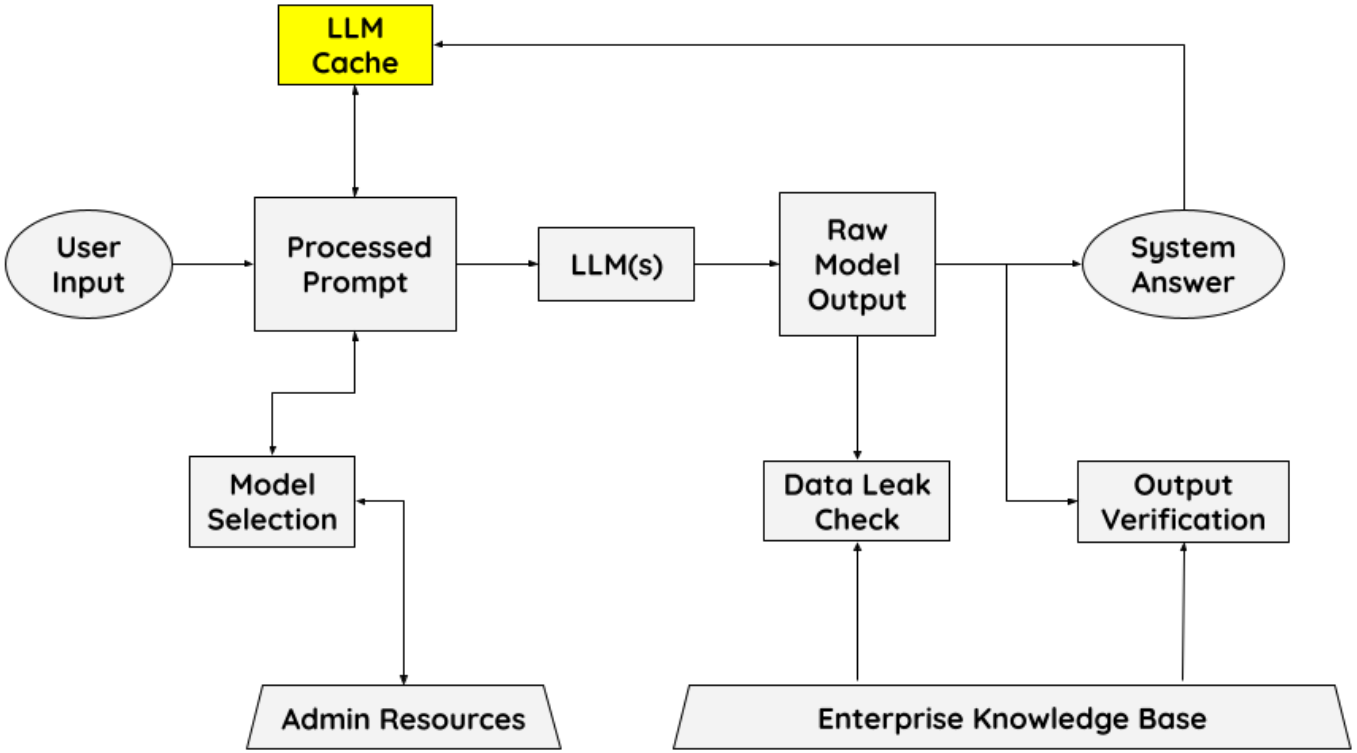
Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

Cache Layer



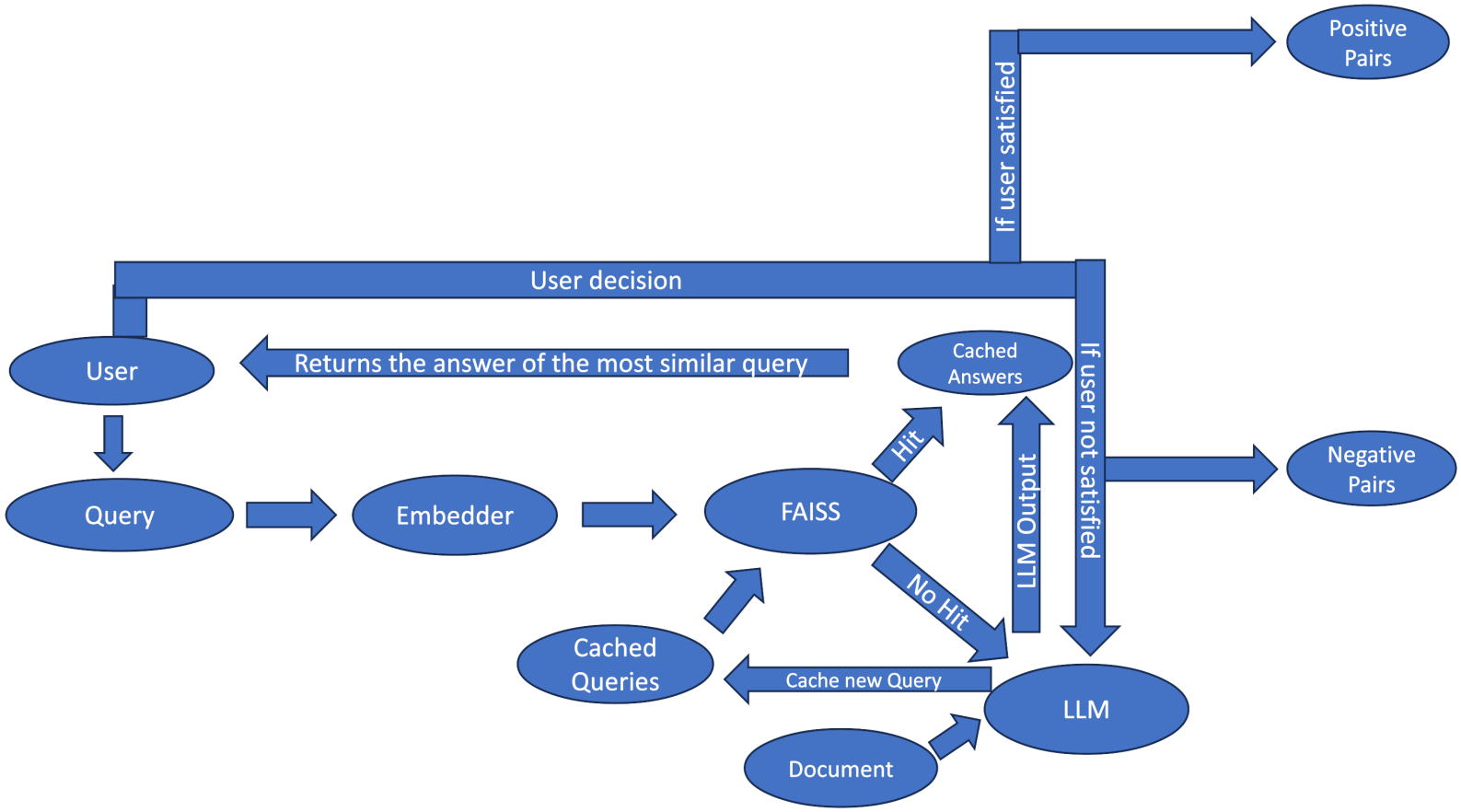
LLM Cache Problem Statement

- Motivation:
  - Rapidly growing demand for real-time question-answering solutions
  - LLMs are non-deterministic systems and pay for this stochasticity in both time and computation
  - Several use cases don't require stochasticity in responses and thus API costs and time can be saved by using a cache of QA pairs
- Our Approach:
  - Implement an LLM Cache that intelligently stores and retrieves QA pairs to optimize a balance of response time, costs, and accuracy

LLM Cache Experiments

- Embedding Algorithms Evaluated:
  - **Sentence Transformers:** Pre-trained models for transforming sentences into embeddings.
  - **BERT Fine-tuned on SQuAD (bert squad model):** BERT model fine-tuned specifically for question-answering.
  - **SimCSE (SimCSE):** Focuses on creating semantically meaningful sentence embeddings.
- Results:
  - Comparative analysis revealed that SimCSE consistently outperformed other embedding methods for LLM Cache.
  - SimCSE embeddings led to more accurate and contextually relevant query similarity assessment.
- FAISS Integration:
  - After embedding, FAISS was employed for efficient semantic search in the cached QA pairs.
  - FAISS's superior speed and memory efficiency made it the optimal choice for handling high-dimensional embeddings.
- Final Implementation:
  - LLM Cache employs SimCSE embeddings and FAISS indexing for optimal query-response efficiency.

Architecture:



LLM Cache Current Work

- **Contrastive Learning Integration:**
  - Use online learning to acquire a dataset of hard negative pairs and positive pairs
  - Identify common structures among hard negatives and use it to generate a dataset of likely hard negatives
  - Train LLM embeddings to maximize distinctiveness for different query clusters
- **Dynamic Similarity Thresholds:**
  - Implement a query classifier to dynamically adjust similarity thresholds based on query types

Limitations and Conclusions

- Challenges Encountered:
  - **Semantic Variability:** Ambiguity in user queries can lead to suboptimal cache utilization
  - **Threshold Generalization:** Difficulty in determining universally applicable similarity thresholds
- Limitations:
  - Cache Hit Rate vs. Resource Trade-off: Striking the right balance remains a challenge
  - Handling Evolving Data: Adapting to changing user behaviors and query patterns
- Conclusion:
  - The LLM Cache represents a substantial step forward in optimizing response time and resource usage
  - Future work involves addressing challenges and continually refining the cache strategy

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

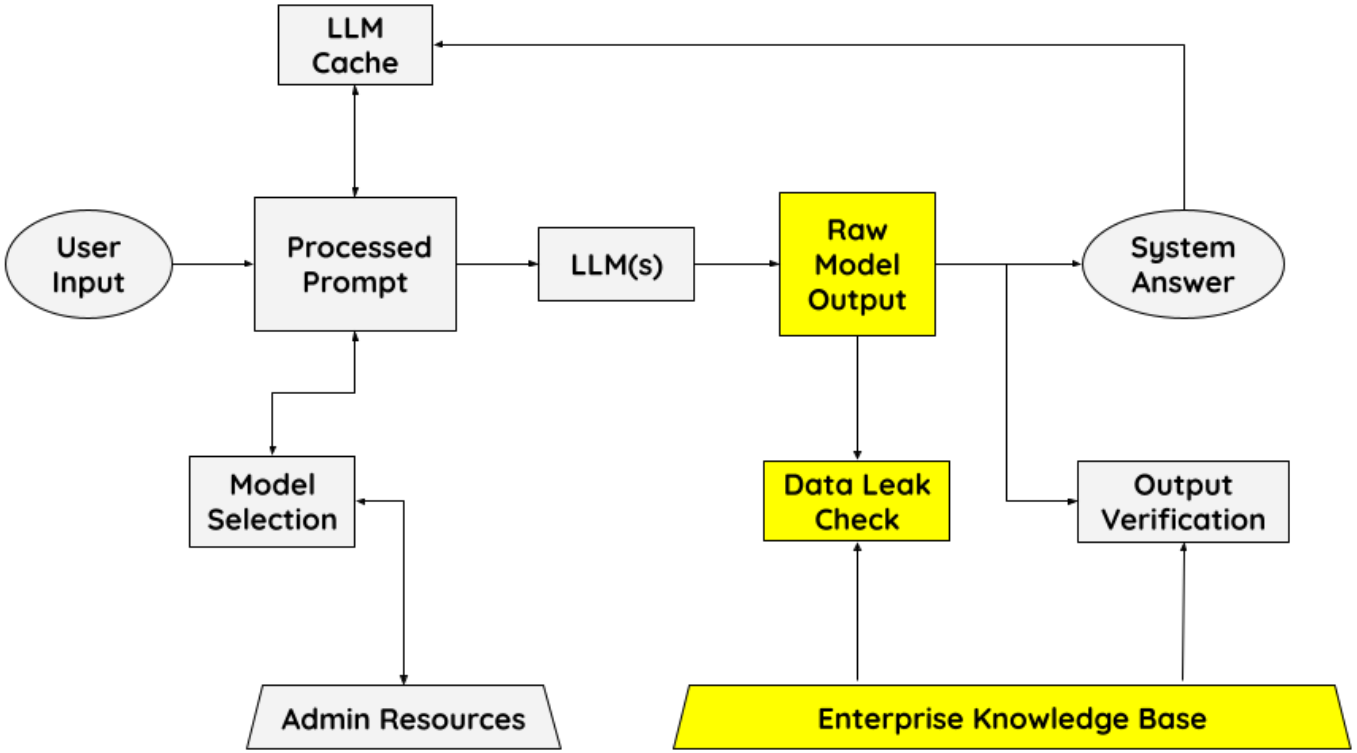
Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

Data Exfiltration Layer



Data Exfiltration Problem Statement

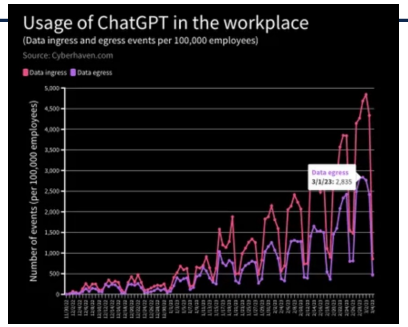
- LLMs understand world knowledge.
- Trained on expansive knowledge bases (undisclosed).
- Memorization of sensitive information is a big concern
    - Private personal details (PII)
    - Intellectual Property

Models fine-tuned on specific enterprise data pose a threat.

Great skew in documents that are to be preserved vs other information, for example, <0.5% of data may need to be protected.

Employees Are Feeding Sensitive Biz Data to ChatGPT, Raising Security Fears

More than 4% of employees have put sensitive corporate data into the large language model, raising concerns that its popularity may result in massive leaks of proprietary information.



Are Large Pre-Trained Language Models Leaking Your Personal Information?

Are Large Pre-Trained Language Models Leaking Your Personal Information?

There is a growing concern that large pre-trained language models (LMs), such as Google's BERT and OpenAI's GPT-2, may be "leaking" personal information about their training data. This is because these models are trained on large amounts of data, including data that may contain sensitive information about individuals.

There is no definitive answer to this question at present. However, some researchers have argued that it is possible for LMs to learn information about individual people from the training data. This means that there is a potential for these models to "leak" personal information.

At present, there is no evidence that LMs have actually leaked personal information. However, the potential for this to happen is a cause for concern. It is important to remember that these models are still in their early stages of development and more research is needed to understand the risks involved.

Data Exfiltration Experiments Overview

1. **Embedding Based**
    1. Finding out natural boundaries in text for embedding level.
    2. How to cohesively embed long-form document which focusing on key ideas?
2. **Topic Modelling Based**
    1. Each document has a set of topics. How does these topics relate to each other in the vector space?
3. **Combined Modelling**
    1. Each document is linked to k-topic embeddings (can be thought as sub-categories in analogy to classification).
    2. Need to ensure all topics of a particular document are close-by but also separable from similar topics of other documents [contrastive learning].

Experiment 1: Embedding Based

1. What is the level of embeddings? Document vs sentence vs paragraph?
2. Comparing SimCSE [1], RAN [2], Sentence-BERT [3]
3. Which embedding technique holds the most information?
4. Information can be held at both the idea level (top-view) or low-information level, which is better for observing leaks?

1. SimCSE: Simple Contrastive Learning of Sentence Embeddings https://arxiv.org/abs/2104.08821
2. Recurrent Attention Networks for Long-text Modeling https://arxiv.org/abs/2306.06843
3. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks https://arxiv.org/abs/1908.10084

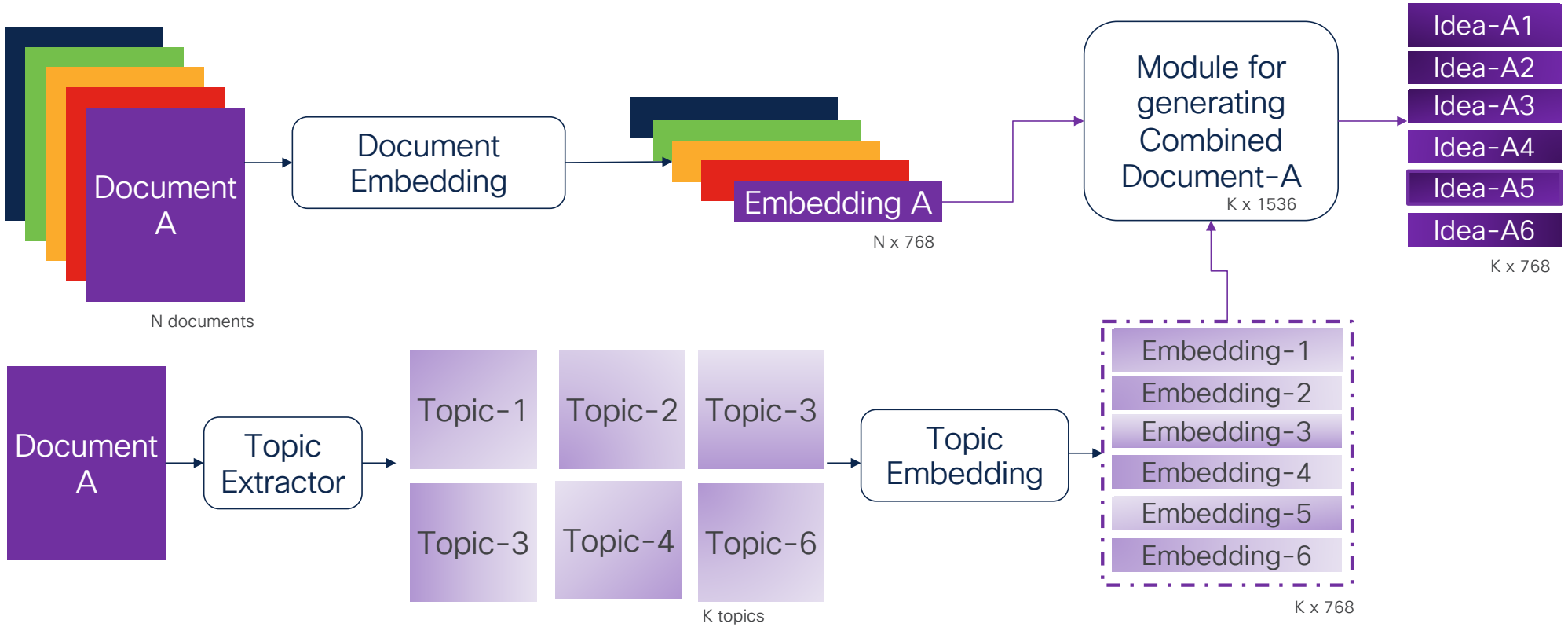
Experiment 2: Topic Modelling Based

1. At what level of knowledge should the topics be created?
2. How can each topic be centered around main idea (subjects) in the document?
3. Each document has a set of topics. How does these topics relate to each other in the vector space?
4. Comparing Top2Vec [1] vs BERT-Topic [2].

For example, an article about Cisco and another about Roads will both have topic (ideas) on networks, but these are semantically very different

1. Top2Vec: Distributed Representations of Topics https://arxiv.org/abs/2008.09470
2. BERTopic https://maartengr.github.io/BERTopic/index.html

Experiment 3: Combined (Idea) based embedding



Projecting the document + topic embedding into a single space

Experimental Protocol

SQUAD 2.0 Dataset

- Training dataset
  - 440 documents
  - Consider x% as sensitive [experiment how model perform with x ranging from 0.5-10]

Positive input passage
- For each document in sensitive set:
  - Protocol-1: Randomly sample 1-3 paragraphs from each document
  - Protocol-2: Cross pollinate information from different documents
  - Protocol-3: Paraphrase information

Negative input passage
- For each document in overall set:
  - Replicate protocol-1, 2, 3 from positive cases

Data Exfiltration Current Work

- Evaluated similarity-based threshold model on following baselines (compared on mentioned protocol)
  - Document embeddings
  - Sentence level embeddings
  - Paragraph level embeddings

  [Note: All experiments are done for different embedding base models, for example simCSE vs diffCSE]

- Implementing and evaluating different Topic Modelling techniques for unsupervised topic extraction
  - BERTopic gives diverse embedding as compared to Topic2Vec

- Combing best similarity-based method (document embedding) with BERTopic embeddigns

Data Exfiltration Conclusion and Future Work

A single vector to represent a document is overloaded with information, drawback include:

- Intricate ideas get left out
- Similarity matching can happen at a high-level

- Contrastive Learning objective:
  - Bring all related combined-idea embeddings closer using dropout for +ve pairs
  - Quantitative evaluation
    - All all the topic augmented of each document closer to each other?
    - How separable are embedding across documents?

- Retrieval based evaluation?
  - How do these idea based embedding compare to input embedding? (note: input embeddings don't have any notion of topic augmentation)
  - Do embeddings still make semantic sense?

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

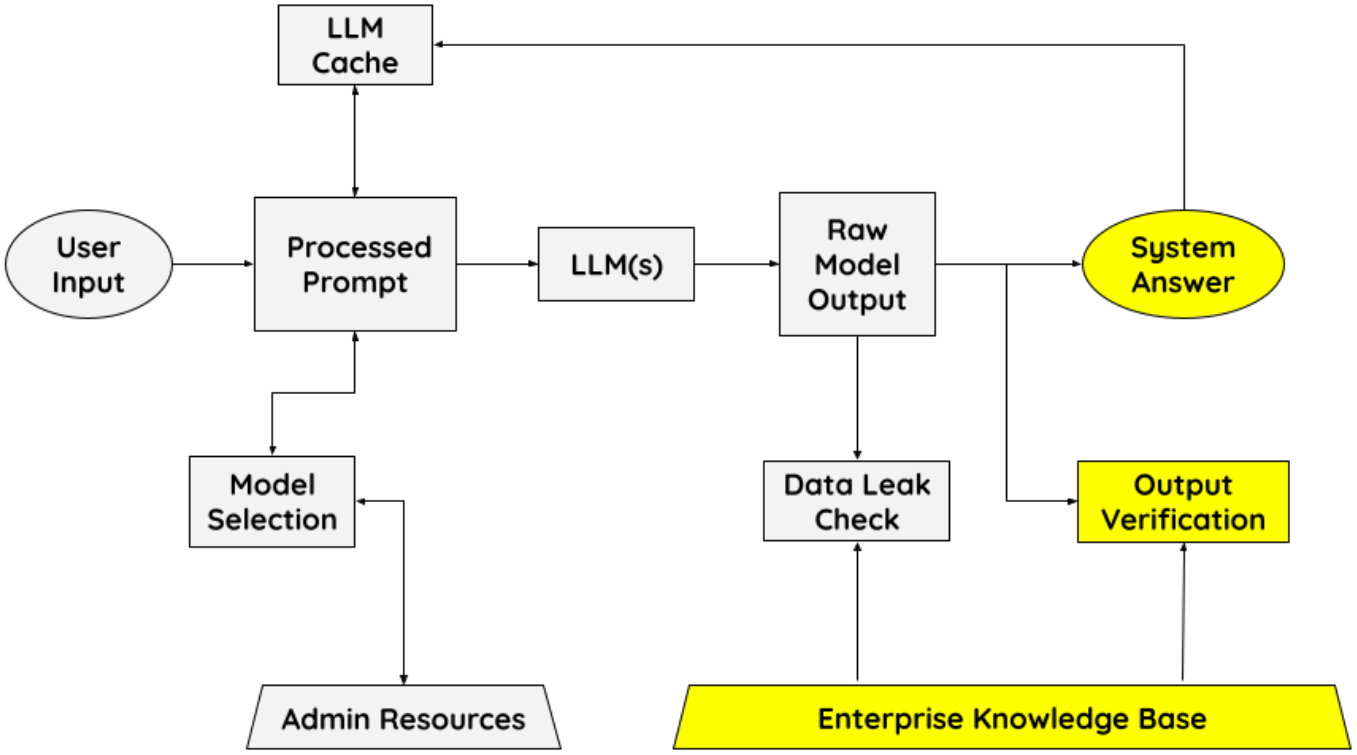
Output Components

Layer: Data Exfiltration

Layer: Output Verification

BUFFALOs and Beyond

Verification Layer



Verification – Problem Statement

Given the following:

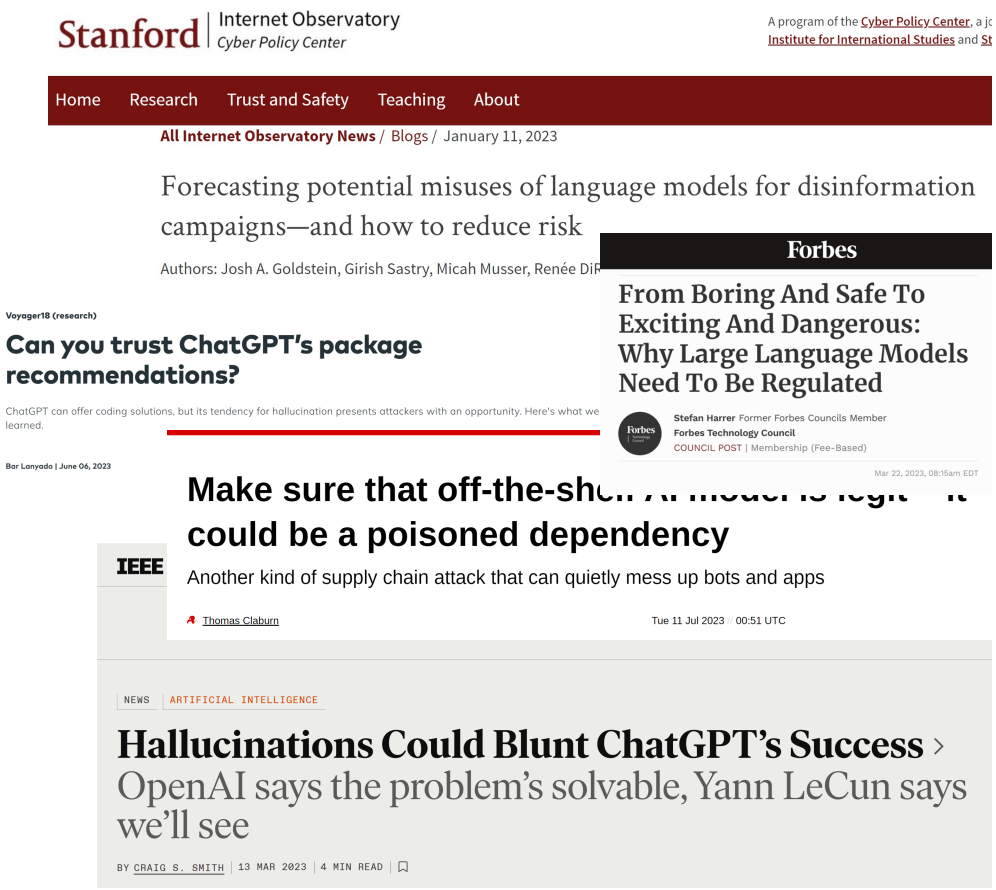
- Input Prompt
- Output response
- Knowledge Base (optional)

How do we verify the output response?

- Interpretable, low-cost confidence metric

Constraints

- Cannot query another LLM
- Easily explainable metric
- Scalable to Knowledge Base



Verification – Experiments Overview

Case 1.
No Knowledge Base

A) Topic Modelling

B) Ensuing Sentence

C) Who/What/When/Where/Why

D) Facts vs. Opinion Identification

Case 2.
Some Knowledge Base

E) Knowledge-Graph Subset
- Semantic-Search
- (Retrieval-Augmented Correction)

F) Document-Level Embeddings
- Reverse Exfiltration
- (Retrieval-Augmented Generation)

A) Topic Modelling

Extract topic(s) of prompt and response to ensure relevancy

Technologies: NER, BERTopic, LDA, LSA, HuggingFace Topic Models

Text	Topics
<prompt>	1973 Oil Crisis, Nixon
<outputA>	1973 Oil Crisis
<outputB>	1973 Oil Crisis, Nixon, OPEC

Prompt topics should be subset of Output topics

"If I wanted to build a farmhouse for my ducks, I would…"

- *(Output A) "start by purchasing ducks to ensure that you have ducks."*
  - Perplexity score of Prompt + A: `60.9341`
- *(Output B) "start by determining how much space your ducks need."*
  - Perplexity score of Prompt + B: `43.7368`

Technologies: BERT, Next Sentence Prediction, Perplexity Score

Next Sentence model
to quantify likelihood of output given input

B) Ensuing Sentence

C) W* Tagging

Tag with <who/what/why/when/where>, match to ensure each portion of prompt has an answer

Technologies: sentence distillation, keyword extraction, classification

(Input Prompt) "<when> When was JFK elected to the presidency <when>, and <what> which party did he represent? <what>. <where> Where was his inauguration speech <where> and <when> how long after his election did this occur?<when>"

(GPT-3.5 Response) "<when> JFK was elected to the presidency in 1960 <when>, representing the Democratic Party. <where> He gave his inauguration speech in Washington D.C. <where> and <when> this occurred approximately two months after his election. <when>"

(Generated Response) "Eeyore is one of the most miserable characters in Winnie the Pooh. In season 1, episode 2, he exclaims "looks like fun, wish I could have some" upon seeing his friends having a party. This clearly shows that he feels left-out and insecure to join the rest of the animals."

(Response, Labelled) "Eeyore is one of the most miserable characters in Winnie the Pooh (opinion, claim). In season 1, episode 2, he exclaims "looks like fun, wish I could have some" upon seeing his friends having a party (fact, evidence). This clearly shows that he feels left-out and insecure to join the rest of the animals (argument). "

Technologies: fact-or-opinion-xmlr, score-claim-identification models

Identify facts/opinions in response to determine how much can be verified.

D) Fact vs. Opinion

E) Knowledge-Graph

Extract relationship tuples from output/prompt, independently verify each tuple from Knowledge Base KG

Technologies: OpenIE, AMR, ES

(LLM Output) "Disney released Beauty and the Beast in 1991, continuing Disney's Golden Age"

Compare triplets with WikiGraph for Disney, Beauty and the Beast

LLM Output	WikiGraph
(Disney, released, Beauty and the Beast)	(Disney, produced, Beauty and the Beast)
(Beauty and the Beast, released in, 1991)	(Beauty and the Beast, made, November 1991)
(Disney, has, Golden Age)	(Disney, third era, Golden Age)

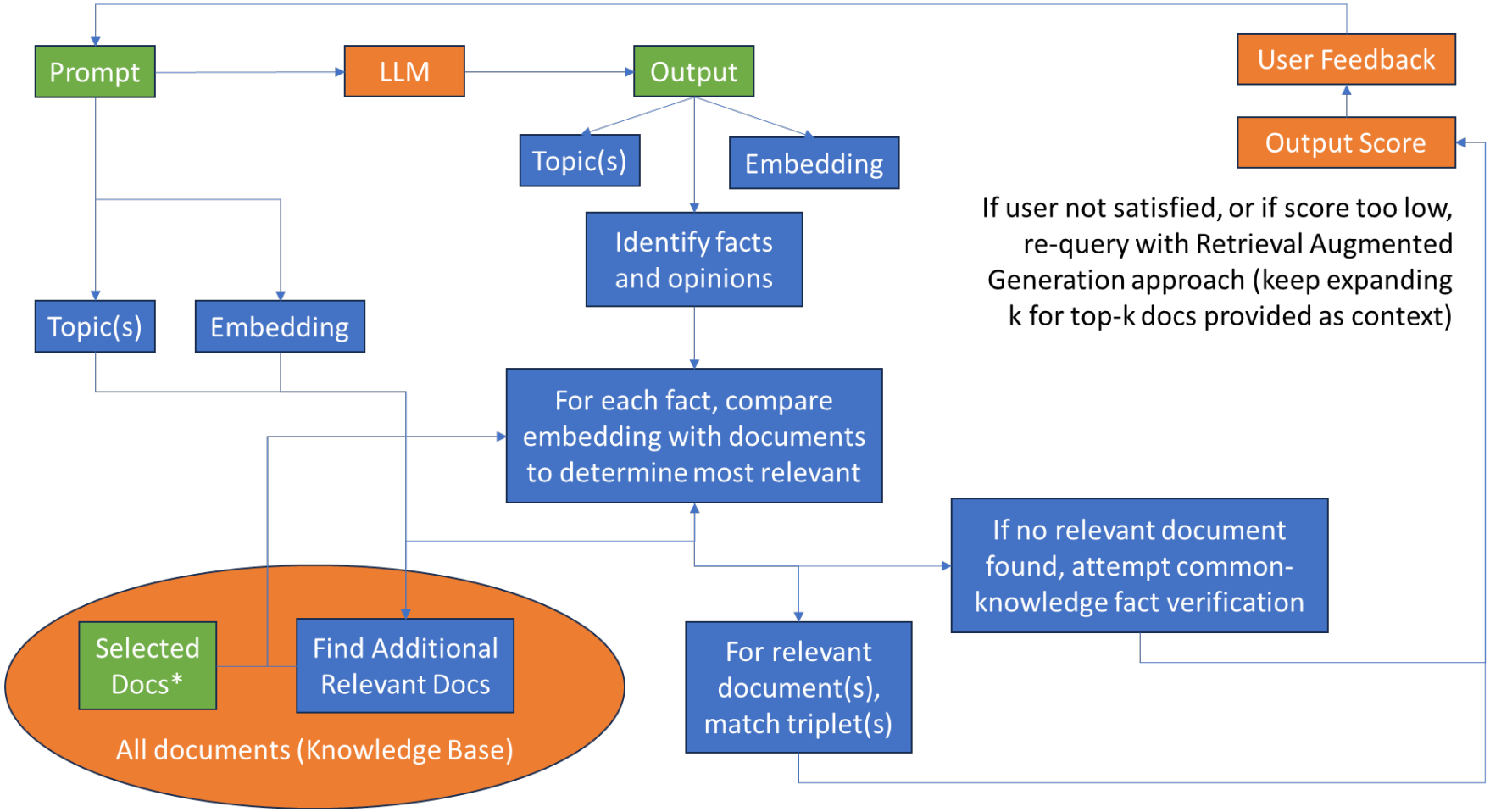
- KG Comparison
  - (only look at subset of top-k docs)
- Retrieval-Augmented-Generation
  - (extract, re-query if < score)

Raunak's Data Exfiltration (reverse)

Compare embeddings to determine similarity scores for each doc

- Which docs has this come from?

F) Doc-Level Embedding



From Experiments to Enterprise

Verification - Conclusion & Future Work

- Improving each stage of the Output Verification pipeline
  - New topic modelling methods
  - Faster document/KG search
  - Implementing AMR-IE
  - Knowledge Graph Alignment
- Conversion of Enterprise Documents to KG
  - Moving away from Elasticsearch + BERT
- Easy Output-Layer Additions
  - URL Checking, PII Redaction, Sentiment/Tone Guidance

Charting a Course

Limitations of the LLM Landscape

Introducing, J-WATR-BUFFALO

10,000 Feet

From BLAZE to Buffalo

Input Components

Layer: Prompt

Layer: LLM cache

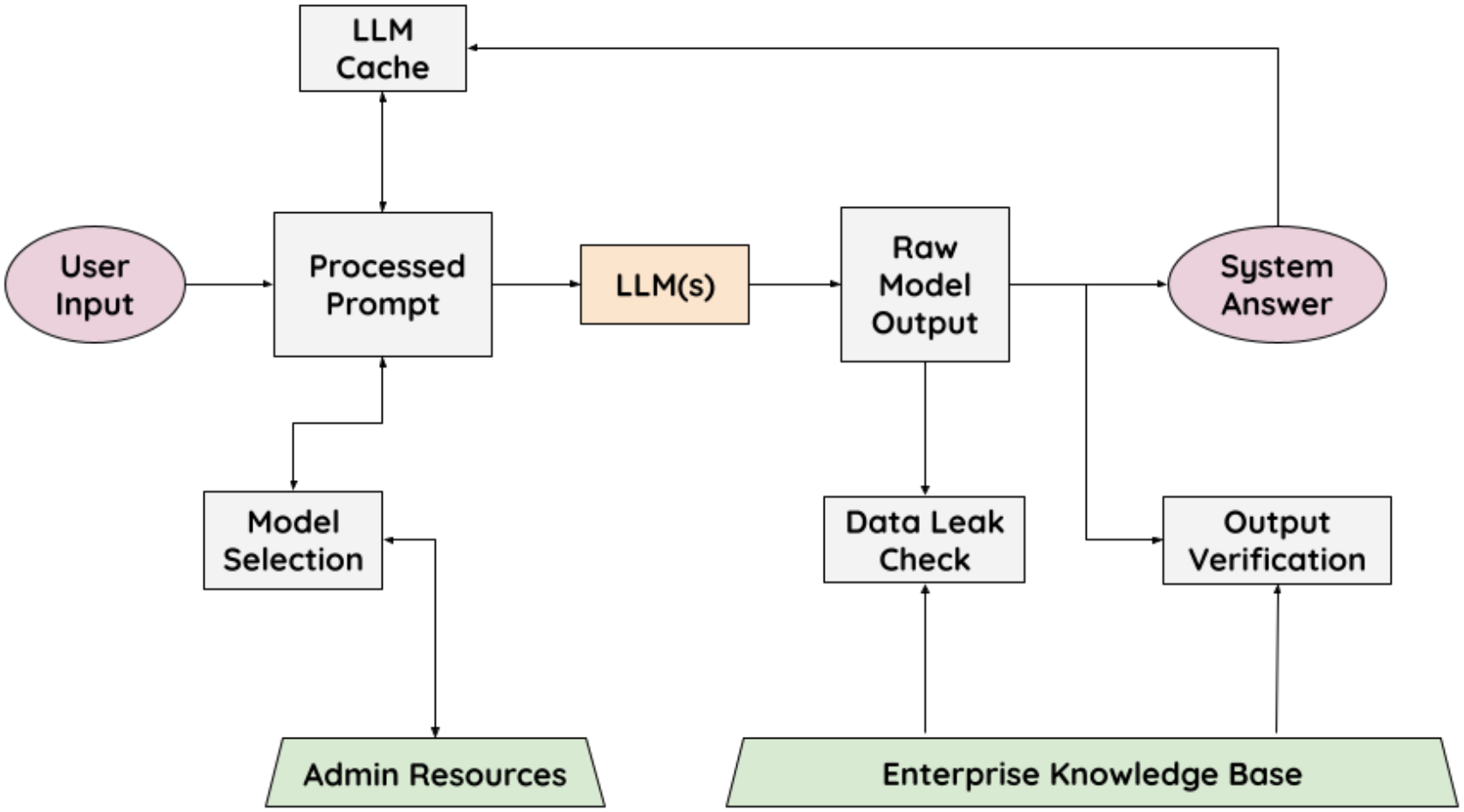
Output Components

Layer: Data Exfiltration

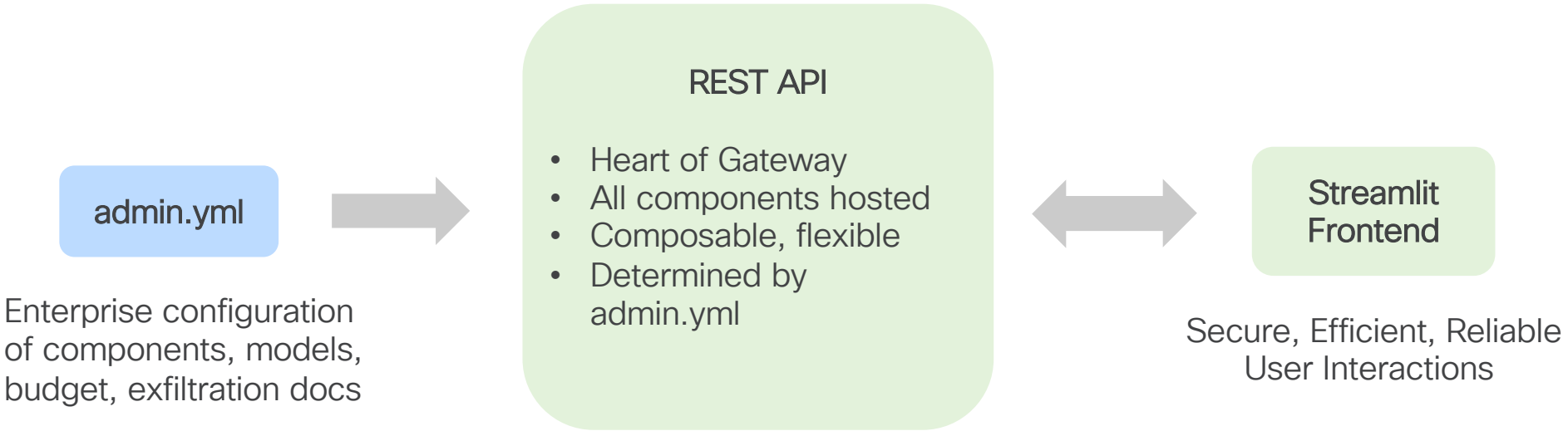
Layer: Output Verification

BUFFALOs and Beyond

J-WATR-BUFFALO Demo



J-WATR-BUFFALO - Architecture



gateway > ! admin.yml

```yaml
# Enterprise LLM Gateway - Admin Configurable Policy

# Here, admins can specify the settings for their Gateway.
# Currently-implemented features are detailed below.


# Section 01) Prompt Layer

use_prompt : True
user_model_choice: True
display_knowledge: True

supported_models:
- "gpt3.5"
- "gpt4"
- "gptJ"
- "llama2"
- "dolly-3b"

# Section 02) Cache Layer

use_cache: True
cache_thresh: "auto"
cache_default: 0.8

# Section 03) Exfiltration

use_exfiltration: True
sensitive_info:
  - "employee_info.txt"
```

. . .

(GET) /docs

(POST) /llm

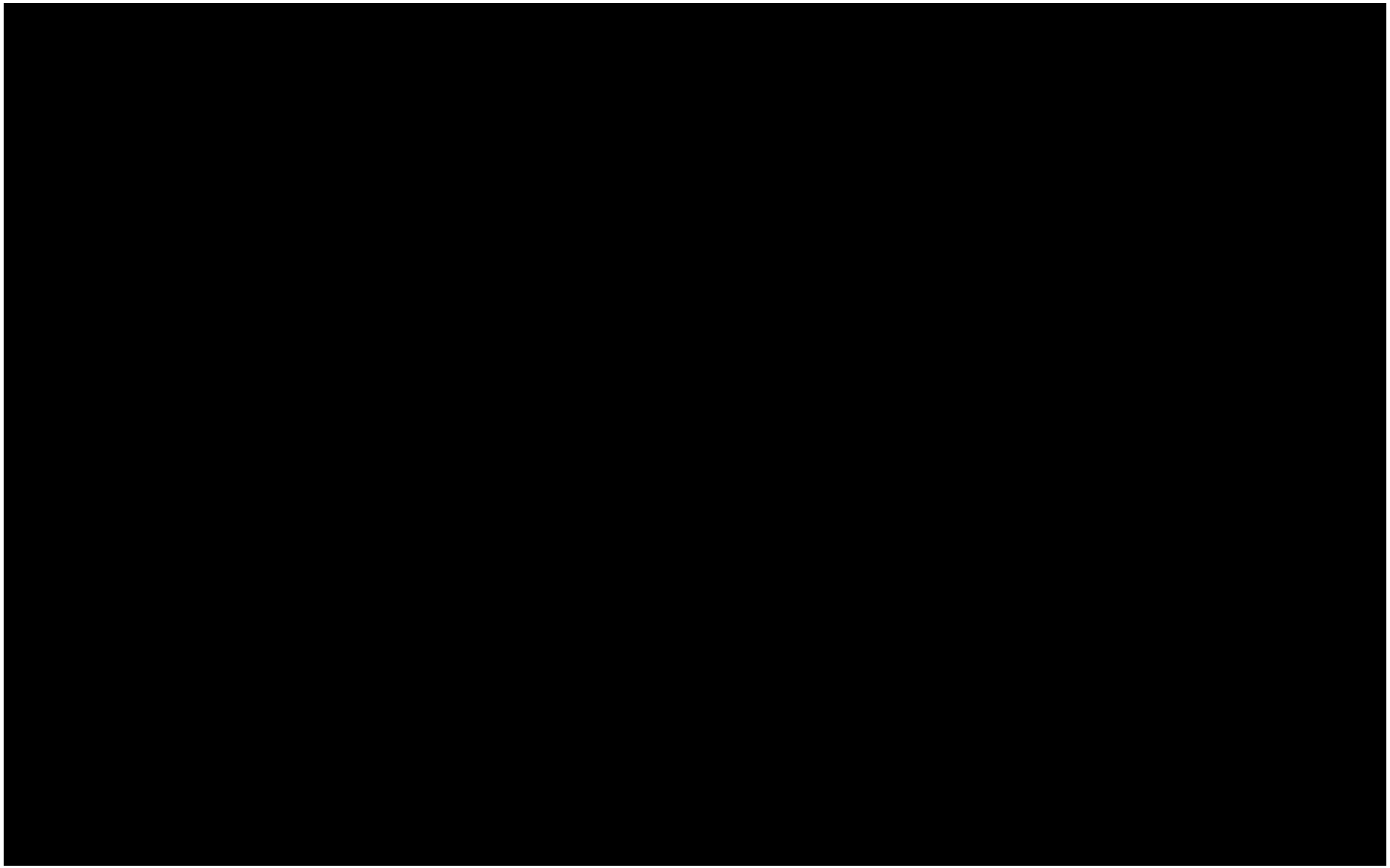
(POST) /prompt

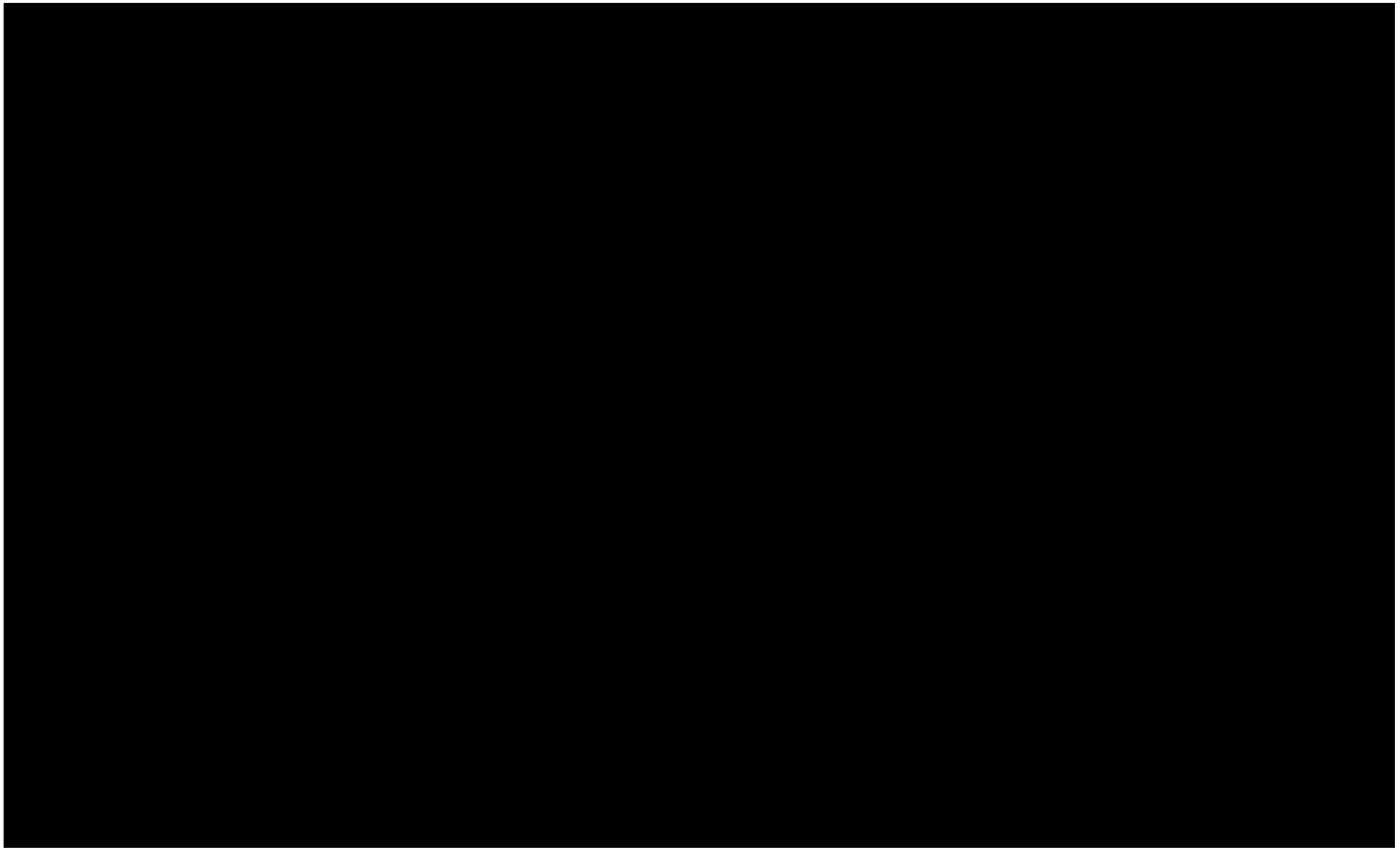
(POST) /cache

(POST) /exfiltrator

(POST) /verify

. . .





Limitations of the LLM Landscape?

Bridging the Gap with BUFFALO

Data Privacy
- Context-aware PII Redaction, Info Safeguards

Expenses
- Cost Monitoring, Caching, Query Batching

Data Security
- Data Exfiltration, User-specific Access Control

Hallucinations
- Verification, Retrieval-Augmented Correction

Explainability
- Step-by-step Decomposability, Open-Source

Competence
- ToolGPT, In-House Research Paper

LLM Gateways – Existing Endeavors

Name	Supported Feature(s)	Limitations (The BUFFALO Benefit)
LIVEPERSON	• Called "Hallucination Detection", but only validates URLs, Phone #s, Email Addrs • URL Post-processing (wraps HTML tags)	• Nothing beyond checking URLs, #s, @s • BUFFALO includes all stated features • BUFFALO's broader hallucination detection
Wealthsimple	• Runs PII scrubbing heuristics • Developed in-house PII removal model • Have rate limits to prevent errors	• Not open-source, only provides PII redaction • BUFFALO's context-aware PII redaction • BUFFALO's Document-level configurability
Espressive Barista	• Demo: Use w/ Zoom - transcript summary • Internal LLM to better understand nuances of organization's lexicon • Translation – an LLM per language	• BUFFALO is composable: has demo of WebEx transcript analysis (live and past meetings) • BUFFALO has optimized LLM selection based on intent/classification of query (admin-config)
Prediction Guard	• API endpoints for consistency, factuality • Integration with LangChain with Wrapper • "Factuality" score that uses reference, text	• BUFFALO also provides configurable endpoints • Working on integration with BLAZE, LangChain • Is closed-source (unsure what scores mean), while BUFFALO is decomposable, explainable • Working on open-sourcing BUFFALO

Next Steps

- Integrating with functionality including function calling, tool usage (PanopticaGPT)
- Incorporate research...
- Integrate with BLAZE/opensource repo
- Enhance gateway with external research (sponsored research)
- Incorporate active/online learning and analytics into solution

Meeting the Team

Tarun Raheja

Masters @ UPenn

- Published blog post on SOTA prompting techniques
- Leading project on novel SoTA method of reasoning via classification with LLMs.
- Released open source Python package, ToolGPT (62 stars).

Will Healy

Bachelors @ Stanford

- Designed LLM cache - adding features, conducting experiments to improve the performance
- Worked on designing, implementing pipeline
- Collaborating on research for using classification tasks for logical reasoning

Raunak Sinha

Masters @ UCLA

- Working on publication on data exfiltration
- Collaborating for the AAAI 2024 draft for solving first order logical reasoning
- Researching on various combinations for adding to LLM gateway pipeline and implemented for the PoC

Advit Deepak

Bachelors @ UCLA

- Experimented with and implemented novel verification using KGs
- Worked on designing, implementing pipeline
- Collaborating on research for using classification tasks for logical reasoning

The bridge to possible