LEXART

Forged in Code. Driven by Challenge

Predictive Middleware for Asynchronous Token Estimation in Ultra-High-Speed LLMs.

*Author: **Alex Casadevall** (CEO), **Whendel Morais** (Software Architect) @ Architects Team*

The Budget & Latency Challenge

LEXART

- The stochastic and autoregressive nature of ultra-high-speed models, like gemini-3-flash-preview, makes output length difficult to estimate without executing full inference.
- Managing budget exhaustion requires accurately predicting this consumption beforehand.
- The challenge is doing this without introducing noticeable latency overhead into the system.

The Core Paradigm Shift

LEXART

- Instead of evaluating the massive full context of a prompt, our system shifts the focus to extracting instructional features from the input.
- We utilize lightweight proxy models rather than heavy neural layers for initial estimations.
- This approach is what allows us to maintain the strict ultra-low latency budget intrinsic to high-speed proxy estimators.

System Architecture Overview

LEXART

Feature Extraction & The MoPE Router

LEXART

- The Extractor utilizes a TF-IDF vectorizer to map the prompt's features efficiently.
- A lightweight router assigns the verbosity prediction task to specialized regressors.
- These specialized regressors act as "Experts" for distinct domains, such as Code Generation, Summarization, or General Chat.

The Risk of Standard Estimation

LEXART

- Using standard mean regression models tends to underestimate token counts.
- Underestimating token counts is dangerous because it directly causes budget exhaustion.
- To safeguard user request limits with high precision, the system requires a stricter mathematical guardrail.

Mathematical Foundation I

LEXART

The Baseline Problem: Mean Regression (MSE)

- Standard models predict the expected value: $\hat{y} = E[Y|X]$
- Risk: The probability of underestimating is roughly 50% $P(Y > \hat{y}) \approx 0.5$
- *Result:* Unacceptable rate of budget exhaustion.

The Solution: Quantile Regression

- Instead of the mean, we predict the conditional quantile: $\hat{y}_q = Q_q(Y|X)$
- We set a high safety threshold, e.g., . $\mathrm{q} = 0.85 or 0.95$
- Guarantee: $P(Y \leq \hat{y}_q) = q$. We ensure the actual output length is strictly less than our prediction $q\%$ of the time.

The Objective Function: Pinball Loss (Quantile Loss)

- To train our MoPE router to predict this upper bound, we minimize:
- $L_q(y, \hat{y}) = \max(q(y - \hat{y}), (1 - q)(\hat{y} - y))$
- Where y is actual tokens, $\hat{y}$ is predicted tokens, and q is the target quantile.

Mathematical Foundation II

LEXART

The MoPE Prediction Function

The total predicted output ($\hat{T}_{out}$) is calculated by routing the input features (x) through our Mixture of Experts:

$$\hat{T}_{out} = \sum_{i=1}^{k} g_i(x) \cdot E_i(x)$$

- *Where $g_i(x)$ is the gating function (probability of routing to expert i) and $E_i(x)$ is the specific expert's quantile prediction.*

The Total Estimated Cost Function

We decouple the deterministic from the stochastic: $C_{total}(x) = T_{in} + \hat{T}_{out} + \delta$

- T_{in} *: Exact input token count (Deterministic, computed locally).*
- δ *: An optional system-level safety padding (Static).*

The Guardrail Decision Matrix

Given a remaining user quota (Q_{rem}), the middleware acts as a step function:

$$\text{Action} = \begin{cases} \text{Execute API}, & \text{if } C_{total}(x) \leq Q_{rem} \\ \text{Block (HTTP 429)}, & \text{if } C_{total}(x) > Q_{rem} \end{cases}$$

The Asynchronous Feedback Loop

LEXART

The Reality: Static proxy models degrade over time as user prompt behavior changes ("Prediction Drift").

Continuous Asynchronous Optimization:

- We maintain our 85th percentile guardrail by continuing to optimize against the **Pinball Loss** in production.

The Mechanism:

1. System logs the actual token count (y) from the Gemini API response.
2. Calculates the Asymmetric Error: $L_q(y, \hat{y}) = \max(q(y - \hat{y}), (1 - q)(\hat{y} - y))$
3. Applies gradient updates (e.g., via SGD) to recalibrate the MoPE experts during off-peak hours.

Future Work

LEXART

The Next Challenge: Continuous input streaming and massive context windows (1 Million+ tokens).

Sparse Attention Architectures:

- Transitioning the proxy layer toward sparse mechanisms.
- Isolating informative input windows to bypass excessive positional overhead.

Multimodal Estimation: Integrating the predictor to estimate cost limits for image and video ingestion.

References & Academic Foundations

LEXART

Proxy Models & Encoder-Only Prediction:

- *NeurIPS (2023).* Foundations of proxy modeling for LLM verbosity and routing.
- *arXiv:2404.08509 (2024).* Demonstrating semantic pattern recognition for token length inference.

Feature Extraction:

- *IEEE 11290434 (2024).* Prompt2Length methodologies utilizing symbolic instructions over dense embeddings.

Mixture of Prediction Experts (MoPE):

- *ICLR (2025).* Equinox and advanced routing mechanisms directing prompt domains to specialized regressors.