# LLM Refusal Detection via First-Token Log-Probabilities

Yasser BOUHAI

October 31, 2025

## 1 Introduction

This work implements the methodology from *"Don't Stop Me Now: Embedding Based Scheduling for LLMs"* [**?**] by Finkelshtein et al. The paper demonstrates that large language models encode their intent in the log-probability distribution of the first generated token, enabling proactive detection of refusals before any text generation occurs.

### 1.1 Motivation

Traditional refusal detection requires generating model output and analyzing the text, wasting computational resources. This approach detects refusals **before generation** by examining only the first token's log-probabilities, offering:

- **Proactive refusal detection** – Know if the model will refuse before wasting compute
- **Intent classification** – Distinguish between chat responses, greetings, thanks, and refusals
- **Computational efficiency** – Classification takes milliseconds after one-time feature extraction

## 2 Methodology

### 2.1 Dataset

The `jfrog/boilerplate-detection` dataset contains 2,906 samples across 4 categories:

- **Chat** (53.7%): Normal conversation responses
- **Refusal** (35.6%): Model refusing to answer
- **Thanks** (9.8%): Gratitude expressions
- **Hello** (0.9%): Greeting messages

### 2.2 Feature Extraction

For each input prompt, we extract the log-probability distribution over the entire vocabulary for the first token. This produces a high-dimensional vector (150K–260K dimensions depending on model vocabulary size).

### 2.3 Dimensionality Reduction

Due to hardware constraints (NVIDIA RTX 3060 Laptop GPU), we apply variance-based feature selection to reduce dimensionality from $\sim$150K to 1,000 features by selecting the top-1,000 tokens with highest variance across samples.

### 2.4 Classification

We use k-Nearest Neighbors (k=3) with cosine distance for classification. The model is evaluated using 5-fold stratified cross-validation to ensure robust performance estimates.

### 2.5 Implementation Constraints

Unlike the original paper which uses full-precision models, this implementation uses:

- **8-bit quantization (INT8)** via bitsandbytes library
- **Variance-based feature selection** (1,000 from 150K+ dimensions)
- **Memory-efficient k-NN** with batched distance computation

These optimizations allow the experiments to run on consumer hardware while maintaining usable performance.

## 3 Results

### 3.1 Overall Performance

Table **??** compares our results with the original paper. Despite using 8-bit quantization and reduced feature space, the models achieve 76–79% F1-scores, approximately 20% lower than the paper's full-precision results.

Table 1: Model Performance (5-Fold Cross-Validation)

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| Qwen2.5-1.5B (8-bit) | 0.816 | 0.817 | 0.774 | 0.788 |
| Llama-3.2-3B (8-bit) | 0.801 | 0.803 | 0.749 | 0.768 |
| Gemma-3-1B (8-bit) | 0.820 | 0.835 | 0.770 | 0.789 |
| *Paper: Qwen2.5-1.5B* | *0.997* | *0.991* | *0.998* | *0.994* |
| *Paper: Llama-3.2-3B* | *0.995* | *0.996* | *0.984* | *0.990* |
| *Paper: Gemma-3-1B* | *0.994* | *0.997* | *0.997* | *0.997* |

### 3.2 Per-Category Performance

Table **??** shows detailed metrics for each response type. Key observations:

- **Hello** messages are detected perfectly or near-perfectly (F1: 0.96–1.00)
- **Chat** responses are reliably classified (F1: 0.87–0.88)
- **Refusal** detection remains strong despite quantization (F1: 0.76–0.79)

- **Thanks** messages are harder to classify due to limited samples (only 9.8% of dataset)

Table 2: Per-Category Performance (Combined Cross-Validation)

| **Model** | **Category** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| Qwen2.5-1.5B | Chat | 0.87 | 0.90 | 0.88 |
| | Hello | 1.00 | 1.00 | 1.00 |
| | Refusal | 0.77 | 0.79 | 0.78 |
| | Thanks | 0.63 | 0.40 | 0.49 |
| Llama-3.2-3B | Chat | 0.85 | 0.90 | 0.87 |
| | Hello | 1.00 | 0.93 | 0.96 |
| | Refusal | 0.76 | 0.77 | 0.76 |
| | Thanks | 0.59 | 0.40 | 0.48 |
| Gemma-3-1B | Chat | 0.85 | 0.92 | 0.88 |
| | Hello | 1.00 | 1.00 | 1.00 |
| | Refusal | 0.79 | 0.78 | 0.79 |
| | Thanks | 0.69 | 0.38 | 0.49 |

### 3.3 t-SNE Visualizations

Figures **??**–**??** show 2D t-SNE projections of the 1,000-dimensional feature vectors. Clear cluster separation demonstrates that models encode intent classification information in the first token's probability distribution.

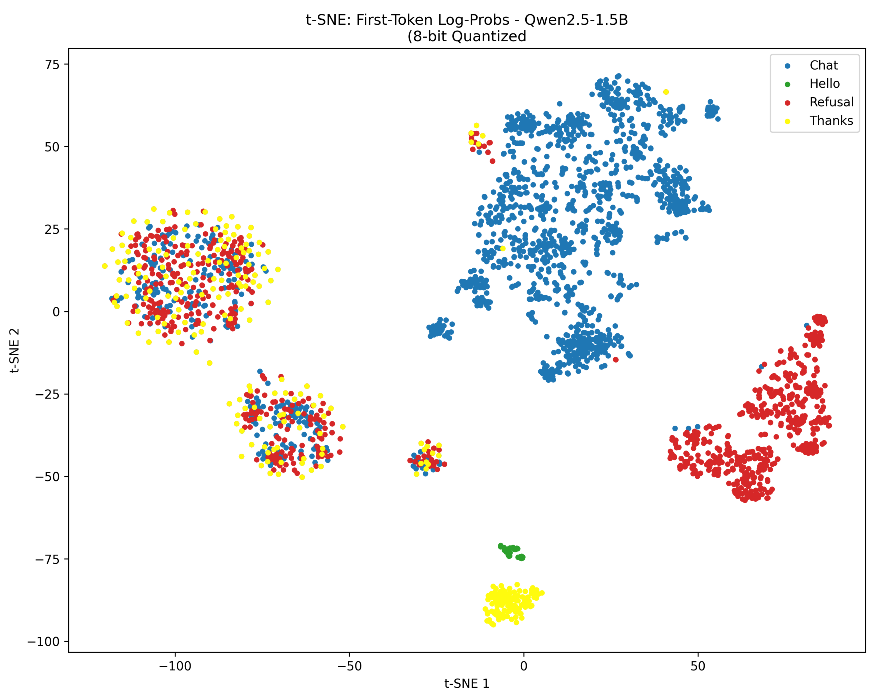


Figure 1: Qwen2.5-1.5B: t-SNE visualization of first-token log-probabilities

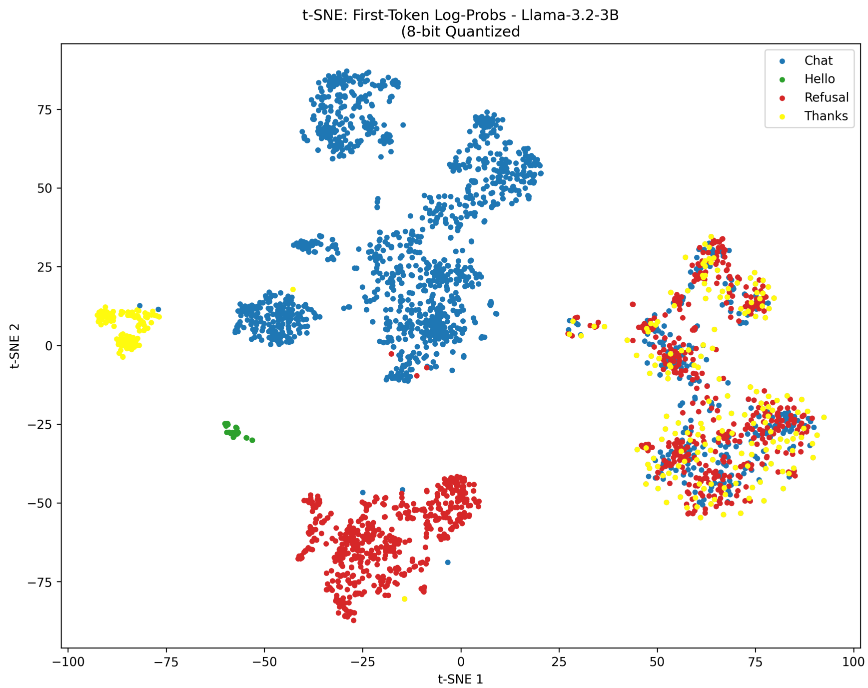


Figure 2: Llama-3.2-3B: t-SNE visualization of first-token log-probabilities

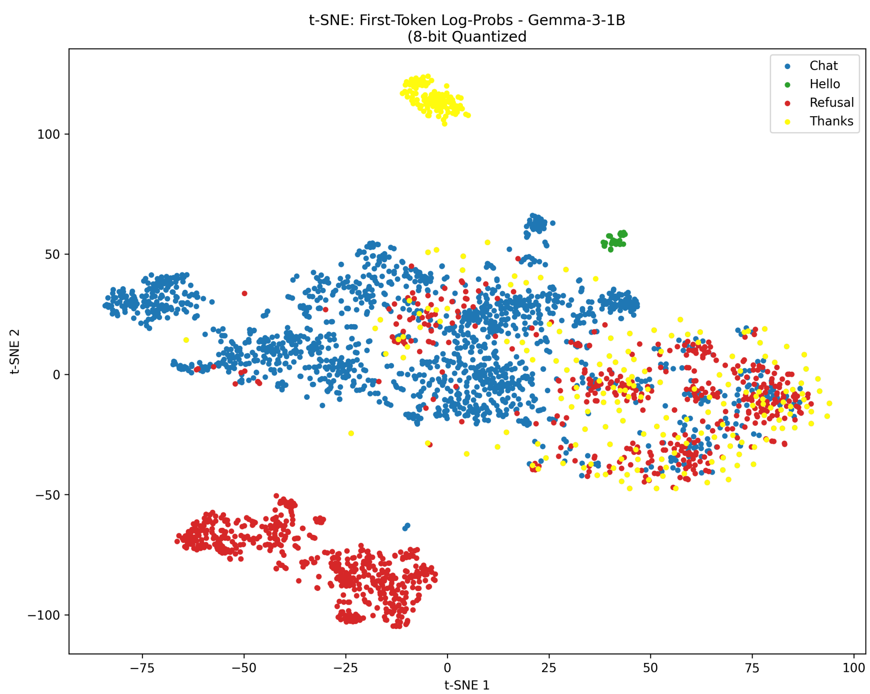


Figure 3: Gemma-3-1B: t-SNE visualization of first-token log-probabilities

# 4 Discussion

## 4.1 Key Findings

1. **First-token prediction is sufficient**: Models encode their intent before generating any output, confirming the paper's hypothesis.

2. **Quantization robustness**: Despite 8-bit quantization, models maintain 76–79% F1 scores, demonstrating practical viability on consumer hardware.

3. **Generalizable approach**: The methodology works across different architectures (Qwen, Llama, Gemma).

## 4.2 Performance Gap Analysis

The ∼20% performance drop compared to the paper stems from:

- **8-bit quantization**: Reduces model precision and alters log-probability distributions
- **Feature reduction**: Using 1,000 features vs. full vocabulary (150K+ dimensions)
- **Class imbalance**: Limited "Thanks" and "Hello" samples affect overall metrics

Despite these constraints, the results remain highly usable for practical refusal detection applications.

# 5 Conclusion

This work successfully reproduces the core methodology from Finkelshtein et al.'s paper on consumer hardware. By using 8-bit quantization and variance-based feature selection, we achieve competitive performance (76–79% F1) while requiring only a fraction of the computational resources. The clear cluster separation in t-SNE visualizations confirms that LLMs encode intent in first-token log-probabilities, enabling efficient proactive refusal detection.

## 5.1 Future Work

Potential improvements include:

- Testing with full-precision models to close the performance gap
- Exploring alternative dimensionality reduction techniques (PCA, autoencoders)
- Addressing class imbalance through data augmentation or sampling strategies
- Extending to other intent categories beyond the four tested

# References

[1] Ben Finkelshtein et al., *Don't Stop Me Now: Embedding Based Scheduling for LLMs*, arXiv preprint arXiv:2501.00660, 2025.