

Hierarchical Classification of Customer Feedback using Groq's LLaMA3-70B-8192

Objective

This project was developed to accurately classify customer reviews related to body wash products into hierarchical categories. The classification task was aimed at generating structured, actionable insights that can inform product development and customer experience strategies.

The system utilizes Groq's `llama3-70b-8192` large language model for its efficiency and high-quality output, striking a balance between performance and real-time usability.

Model Configuration and API Integration

The implementation uses the `Groq` Python client, with the model `llama3-70b-8192`. The environment variable `GROQ_API_KEY` is loaded using `dotenv` to securely authenticate requests to the model.

Key model parameters include:

- **Temperature:** 0.2 (for deterministic outputs)
- **Max Tokens:** 1024 (to control output length)

The model is invoked through a function `get_labels(...)`, which accepts the review text and two lists of predefined labels (Level 1 and Level 2). These are used to guide the model's output through an engineered prompt.

Prompt Design and Structure

Prompt engineering was a core part of this implementation. The model prompt includes:

- A system instruction defining the task (classifying reviews into Level 1 and Level 2 tags).
- Explicit listing of all available Level 1 and Level 2 labels.
- A request to return the output strictly in JSON format with only relevant categories.

- A sample input/output example to enforce consistency in formatting.

This structured prompt reduces ambiguity and ensures that the LLM only selects from predefined labels, improving classification reliability.

Data Pipeline and Preprocessing

The input data (customer reviews) is assumed to be stored in a structured format (e.g., CSV). The process involves:

1. Reading and preprocessing the text data.
2. Feeding reviews into the classification pipeline.
3. Parsing the model's JSON output for Level 1 and Level 2 tags.
4. Storing results in CSV format for analysis.

The prompt and API interaction are embedded in a loop or batch process, ensuring scalability.

Evaluation Metrics

Two evaluation metrics are used to validate the classification performance:

1. Jaccard Similarity

- Uses `jaccard_score` from `sklearn`.
- Measures the intersection over union between predicted and ground truth labels.
- Effective for strict categorical matching in multi-label classification.

2. Semantic Similarity

- Uses embeddings from `SentenceTransformer`.
- Computes cosine similarity between embeddings of predicted and actual labels.

- Captures semantic alignment even if the exact label differs slightly.

TF-IDF and other vectorization techniques are also referenced in the code to support alternative evaluation strategies.

Reporting and Output

The classification results include:

- The original review text.
- Predicted Level 1 and Level 2 tags.

These are compiled into a report format and exported to CSV for visualization, audit, and further downstream analysis.

Conclusion

This project demonstrates a data pipeline for multi-level text classification using a t LLM. By combining prompt engineering, performance benchmarking, and robust evaluation techniques, the solution enables structured insights from unstructured customer feedback. The use of Groq's high-performance LLaMA3 model ensures that the classification is both accurate and scalable for real-time applications.