**Benchmarking Energy Consumption Across Large Language Models: A Comprehensive Analysis of Prompt Processing Efficiency**

**Abstract**

This research presents a comprehensive framework for analyzing energy consumption patterns in Large Language Models (LLMs) based on linguistic and semantic characteristics of input prompts. We develop a systematic approach to measure, analyze, and predict energy consumption across multiple LLM architectures using real-world conversation data from the LMSYS-Chat-1M dataset.  
Our methodology incorporates over 30 linguistic features spanning syntactic complexity, lexical diversity, semantic content, sentiment analysis, and topic classification to identify key factors driving energy consumption. The study provides actionable insights for optimizing prompt design and model selection for energy-efficient AI applications.

**1. Introduction**

**1.1 Research Question**

**Primary Research Question:**  
How do measurable prompt characteristics—such as word count, syntactic complexity, sentiment, and frequency of specific semantic categories—influence the energy consumption of large language models (LLMs)?

**Secondary Questions:**

* Which linguistic features are most predictive of energy consumption?
* How do different LLM architectures respond to varying prompt complexities?
* Can we develop predictive models for energy consumption based on prompt characteristics?
* What are the cost-energy tradeoffs across different model architectures?

**1.2 Motivation**

As LLMs become increasingly integrated into production systems, understanding their energy consumption patterns becomes critical for:

* **Environmental Impact:** Reducing carbon footprint of AI systems
* **Cost Optimization:** Minimizing computational costs for large-scale deployments
* **Resource Planning:** Predicting infrastructure requirements
* **Model Selection:** Choosing appropriate models for specific use cases

**1.3 Contributions**

This research contributes:

1. **Comprehensive Feature Engineering Framework:** 30+ linguistic and semantic features
2. **Multi-Model Energy Benchmarking:** Comparative analysis across OpenAI, Groq, and Mistral models
3. **Real-World Dataset:** Analysis of 75,000+ prompts from actual user conversations
4. **Reproducible Methodology:** Open-source pipeline for energy consumption analysis
5. **Predictive Models:** Statistical models for energy consumption prediction

**2. Related Work**

**2.1 Energy Consumption in AI Systems**

Prior studies highlight the substantial energy requirements of large language models:

* GPT-3 training estimated at 1,300 MWh
* Inference energy varies significantly by model architecture and input complexity
* Growing need for standardized benchmarking of energy efficiency

**2.2 Linguistic Analysis in NLP**

Text characteristics influence computational effort:

* Syntactic complexity affects parsing time and memory usage
* Lexical diversity impacts vocabulary lookup and processing
* Semantic richness influences attention mechanisms and resource allocation

**3. Methodology**

**3.1 Data Collection**

**Dataset Source:** LMSYS-Chat-1M (Hugging Face)  
**Sample Size:** 75,000 first-turn user prompts (random sample)  
**Final Dataset:** ~67,500 clean prompts after filtering  
**Rationale:** Real-world diversity (210K users), pre-cleaned, average length 69.5 tokens

**Pipeline Steps:**

1. 01\_data\_collection.ipynb – Data download
2. 02\_data\_cleaning.ipynb – Extraction and filtering
3. 03\_energy\_measurement.ipynb – Energy measurement
4. 04\_feature\_engineering.ipynb – Feature extraction
5. 05\_exploratory\_data\_analysis.ipynb, 06\_statistical\_analysis.ipynb – Analysis

**3.2 Feature Engineering Framework**

**Text Complexity (5 features)**  
syntactic\_tree\_depth, clause\_count, flesch\_kincaid\_grade, gunning\_fog\_index, smog\_index

**Lexical Features (4 features)**  
avg\_word\_frequency, lexical\_diversity, type\_token\_ratio, vocabulary\_richness

**Semantic Features (2 features)**  
named\_entity\_density, semantic\_category\_diversity

**Sentiment Features (2 features)**  
sentiment\_polarity, sentiment\_intensity

**Information Content (2 features)**  
information\_density, avg\_sentence\_length\_prompt

**Topic Keyword Density (8 categories)**  
tech\_ai\_density, business\_finance\_density, health\_medical\_density, education\_learning\_density,  
science\_research\_density, social\_relationships\_density, entertainment\_culture\_density, travel\_lifestyle\_density

**Concept Density (6 categories)**  
abstract\_thinking\_density, problem\_solving\_density, communication\_density, emotional\_psychological\_density,  
decision\_making\_density, time\_change\_density

**3.3 Energy Measurement Protocol**

**Models Evaluated:**

* GPT-4o-mini (OpenAI) — 50% of prompts
* Llama 3.1 8B (Groq) — 40% of prompts
* Mistral Large (Mistral.ai) — 10% of prompts

**Parameters:** Temperature 0.3, Max Tokens 50  
**Metrics:** Energy (Wh), latency, throughput, cost

**Estimated Costs:**

| **Model** | **Share** | **Est. Cost** | **Notes** |
| --- | --- | --- | --- |
| GPT-4o-mini | 50% | $35–40 | Baseline energy data |
| Llama 3.1-8B | 40% | $12–15 | Groq-optimized |
| Mistral Large | 10% | $60–70 | Open-weight |
| **Total** | **100%** | **$110–150** | **Within $400 cap** |

**3.4 Technical Implementation**

**Architecture:**  
Raw → Cleaning → Energy → Features → Analysis  
(01\_collection → 02\_cleaning → 03\_energy → 04\_features → 05\_06\_analysis)

**Core Features:**

* Incremental batch processing
* Retry logic and error recovery
* Real-time logging
* Data validation and reproducibility

**Libraries:** spaCy, NLTK, textstat, wordfreq, pandas, numpy, matplotlib, seaborn, scikit-learn, scipy

**4. Experimental Design**

**Phase 1:** Pilot (100 prompts) → verify energy logging  
**Phase 2:** Full scale (1M prompts) → monitored runs  
**Phase 3:** Validation → energy calibration, reliability checks

**5. Expected Outcomes**

**Primary Findings:**

1. Feature importance rankings
2. Cross-model energy efficiency comparison
3. Predictive energy models
4. Cost-energy tradeoffs

**Secondary Insights:**

* Prompt optimization guidelines
* Model selection criteria
* Environmental impact assessment

**Deliverables:**

* Dataset with 30+ features and energy values
* Statistical and visualization tools
* Full reproducible pipeline and research paper

**6. Data Storage and Format**

**Raw JSON Structure**

{"prompt": "...", "model": "...", "energy\_consumed\_wh": 0.015, "duration": 2.03}

**Feature-Enhanced JSON Structure**  
(includes all engineered features for analysis)

**7. Analysis Framework**

**Exploratory Analysis:** Descriptive statistics, correlations, distributions  
**Statistical Modeling:** Regression, feature selection, cross-validation  
**Visualization:** Energy plots, correlation heatmaps, model comparison charts

**8. Reproducibility and Open Science**

* Code repository (open-source)
* Anonymized dataset and metadata
* Full parameter documentation
* Validation procedures and limitations

**9. Ethical Considerations**

* Data privacy and anonymization
* Environmental transparency
* Ethical use of compute resources

**10. Future Work**

* Include additional model families
* Explore deep semantic feature extraction
* Real-time inference energy tracking
* Develop optimization recommendations

**11. Conclusion**

This study establishes a robust framework for benchmarking energy consumption in LLMs, identifying key linguistic and structural drivers of computational cost.  
By integrating feature-based analysis with reproducible methodology, it contributes actionable insights for sustainable and cost-efficient AI system design.

**References**

1. LMSYS-Chat-1M Dataset – <https://huggingface.co/datasets/lmsys/lmsys-chat-1m>
2. Energy Consumption in AI Systems – <https://arxiv.org/pdf/2407.16893>
3. Linguistic Analysis in NLP – <https://chat.deepseek.com/a/chat/s/5c44573e-18ac-4e7d-a6ab-a275731811d3>

**Appendix**

**A. Complete Feature List**  
(See categorized breakdown: token, syntactic, lexical, semantic, sentiment)

**B. Technical Specifications**  
Python 3.8+, spaCy, NLTK, textstat, pandas, numpy, scikit-learn

**C. Reproducibility Checklist**  
☑ Complete codebase  
☑ Requirements.txt pinned  
☑ Documentation and validation steps  
☑ Sample data and limitations noted