Data Preparation and Model Choice

This section outlines the detailed processes of data preparation and model selection employed in the CarbonAware-Infer project. The goal is to ensure transparency, reproducibility, and fairness while enabling carbon-efficient decision-making across multiple language model configurations.

1. Data Preparation

1.1 Dataset Scope & Provenance

The dataset chosen for this study will be sourced from open-access repositories such as Hugging Face or Kaggle. The dataset must be well-documented, licensed for academic use, and aligned with the short-form question-answering (QA) task. All dataset versions, licenses, and download hashes will be recorded to ensure complete traceability. The dataset will primarily consist of textual QA pairs with varying complexity levels to enable robust benchmarking across models of different sizes.

1.2 Data Composition and Documentation

A detailed datasheet will accompany the dataset, describing its motivation, collection process, labeling strategy, and known limitations. The datasheet will include fields such as dataset motivation, collection date, sources, data volume, and annotation methods. The dataset will also document inclusion and exclusion criteria to prevent biases and ensure domain diversity (e.g., general knowledge, science, history).

1.3 Ethical Screening and PII Removal

All data will undergo a strict ethical screening to remove any personally identifiable information (PII) such as names, email addresses, or contact numbers. A script will automate the process of identifying and redacting such data, and an audit log will record the number of affected records. The dataset will also be reviewed to exclude content containing offensive or discriminatory language.

1.4 Quality Control and Deduplication

To ensure data integrity, multiple preprocessing steps will be applied, including duplicate detection using hashing methods (MinHash or SimHash). Leakage prevention will be prioritized by ensuring that near-identical questions or passages do not appear in multiple dataset splits. Quality checks will include syntactic validation, missing value imputation, and consistency checks between input–output pairs.

1.5 Text Normalization and Tokenization

All text will be standardized using Unicode normalization, whitespace trimming, and basic punctuation normalization. Tokenization will use model-compatible tokenizers such as SentencePiece or Byte-Pair Encoding (BPE), and the tokenizer version will be logged. The preprocessing pipeline will be modular, ensuring the same normalization logic is used during both training and evaluation.

1.6 Context Construction and Input Truncation

For models requiring context (e.g., passage-based QA), a consistent context window will be defined with a maximum token limit. Long inputs will be truncated deterministically (e.g., front-truncation or middle-truncation) to maintain comparability between models. A histogram of input lengths will be recorded to understand the impact of truncation on dataset coverage.

1.7 Dataset Splitting and Stratification

The dataset will be divided into training (80%), validation (10%), and testing (10%) subsets. Stratified sampling will be applied based on question complexity or input length to ensure a balanced representation of easy and hard samples. Random seeds will be fixed and logged to guarantee reproducibility across runs.

1.8 Bias Detection and Subgroup Evaluation

Bias and fairness evaluation hooks will be embedded in the dataset to allow subgroup-based performance analysis (e.g., domain-based or linguistic subgrouping). This enables post-hoc assessment of whether model routing introduces systematic biases.

1.9 Carbon and Energy Metadata Logging

Each inference experiment will log hardware specifications, power consumption estimates, geographic region, and grid carbon intensity. Tools such as CodeCarbon will be used to measure real-time energy consumption and emissions per request. All metadata will be version-controlled to support energy-efficiency benchmarking across models and environments.

2. Model Choice

2.1 Design Objective

The project aims to compare models of varying scales to optimize the trade-off between accuracy, inference latency, energy consumption, and monetary cost. The routing mechanism will dynamically select the appropriate model based on question complexity and energy budget constraints.

2.2 Small Local Model

A smaller model, such as Phi-3 Mini or Gemma-2B, will serve as the baseline for local, low-latency inference. These models will be quantized to 4-bit or 8-bit formats for improved efficiency and reduced power draw. Their role is to handle simpler queries where large models would be excessive. Model configuration, quantization method, tokenizer version, and inference parameters (temperature, max tokens, etc.) will be clearly documented.

2.3 Large Cloud/API Model

A high-performance API-based model (e.g., GPT-4, Claude 3, or Llama 3.1 70B) will be used for complex or ambiguous queries. API usage costs, latency, and energy implications will be logged for each request. Safety settings and privacy safeguards will be documented, especially when handling user-generated text inputs.

2.4 Baselines and Ablation Studies

Three baseline strategies will be implemented: (a) small-model-only inference, (b) large-model-only inference, and (c) dynamic routing based on a learned or heuristic router. Additional ablations will test thresholds based on token length, question type, or confidence score to evaluate trade-offs between cost and performance.

2.5 Evaluation Metrics

Multiple metrics will be used to assess performance: (1) accuracy metrics such as F1-score and Exact Match (EM); (2) efficiency metrics including latency (median and 90th percentile); (3) energy consumption per request (Joules/request) and CO■-equivalent emissions; and (4) cost per request based on API token pricing or local electricity rates. A balanced score combining these metrics will help visualize trade-offs between accuracy and sustainability.

2.6 Reporting and Reproducibility

A comprehensive reproducibility checklist will be provided, covering dataset versions, code commits, model configurations, and random seeds. The experiment environment (hardware, OS, library versions) will be exported using environment YAML files. Additionally, a Model Card will summarize ethical considerations, limitations, and intended use cases of each model.

This framework ensures that both data and models are handled responsibly and scientifically. The data preparation steps emphasize ethical integrity, reproducibility, and fairness, while the model selection process aligns with the project’s sustainability goal of optimizing computational efficiency without compromising performance.