Domain Names Generator

Your Name

September 21, 2025

Abstract

This report details the methodology and results for the initial phase of building a domain name suggestion system. The primary focus was the creation of a high-quality, synthetic dataset grounded in real-world business data. The process involved sourcing business information via the Foursquare API, enriching it with scraped website descriptions, and leveraging Large Language Models (LLMs) for data cleaning, normalization, and final label generation. The resulting dataset serves as a robust foundation for fine-tuning a specialized language model. Due to time constraints, this report covers the dataset creation process exclusively.

Contents

1 Synthetic Dataset Creation

The foundational step of this project was the creation of a high-quality, synthetic dataset for fine-tuning the domain name suggestion model. The methodology was designed to ground the dataset in real-world business information to ensure relevance and diversity, while leveraging a Large Language Model (LLM) for the creative task of generating domain suggestions. The process can be broken down into four main stages: data sourcing, enrichment, cleaning, and label generation.

1.1 Data Sourcing and Enrichment

To ensure the dataset reflects a wide variety of real-world scenarios, initial business data was sourced from the Foursquare Places API. This provided a structured and diverse foundation.

- **Diverse Sectors:** A list of 27 distinct business sectors was targeted, including restaurants, law firms, medical clinics, and IT services.
- **Geographic Variation:** Data collection was performed across major metropolitan areas (e.g., New York, Los Angeles) to capture different regional business styles.

For each business retrieved from Foursquare, a web scraping component was implemented to enrich the entry with a qualitative description from the business's official website. The scraper intelligently searched for the most relevant text by first looking for a meta description tag, then for a dedicated "About Us" page, and finally falling back to generic paragraph text. This process added the necessary descriptive context for the subsequent steps.

1.2 Data Cleaning and LLM-based Filtering

Raw scraped data is often noisy and inconsistent. A multi-step cleaning process was crucial to ensure data quality. After removing entries with technical scraping errors or missing websites, an LLM-based filter was applied to semantically validate the descriptions.

The `gemma2-9b-it` model, accessed via the Groq API, was prompted to act as a data cleaning assistant, classifying each scraped description as either "meaningful" or "not meaningful". This effectively filtered out generic placeholder text, "coming soon" pages, and other non-informative content. This quality assurance step was highly effective, reducing the dataset from an initial **2,428 collected entries** to **894 high-quality entries**.

1.3 LLM-based Description Normalization and Label Generation

With a clean set of businesses, a two-step LLM pipeline was executed to generate the final training examples.

1.3.1 Description Normalization

First, the cleaned 'scraped description' along with other metadata (name, sector, city) was passed to the `gemma2-9b-it` model. The model was tasked with creating a concise, professional, and standardized business description. This normalization step ensures that the input to the final generation model is consistent and high-quality.

```
prompt = f"""
You are a business description normalizer. Your task is to convert
scraped website text into a clean, concise business description using
all available business information.

RULES:
- Create a 1-2 sentence description (max 150 characters)
- Focus on: services/products, target audience, unique value proposition
- Remove: marketing fluff, generic phrases, website navigation text
- Use ALL available business information (name, sector, location, categories)
  to create the best description

BUSINESS INFORMATION:
{context_str}

Original Scraped Description: {business.scraped_description}
```

```

Provide ONLY the normalized description, nothing else.
"""
```

Listing 1: Prompt for Description Normalization

1.3.2 Domain Suggestion Generation

Second, the 'normalized description' was used as input for the final label generation task. A detailed prompt guided the LLM to generate exactly three creative and relevant domain name suggestions with a `.com` extension. Critically, the prompt enforced a structured output, requiring the model to also provide a confidence score and a brief reasoning for each suggestion.

```
prompt = f"""
You are an expert domain name generator. Generate creative, memorable
domain names for this business.

BUSINESS INFO:
Name: {business.name}
Sector: {business.sector}
Location: {city}
Description: {business.normalized_description}

DOMAIN REQUIREMENTS:
- Generate exactly 3 domain suggestions
- Use .com extension only
- 6-15 characters (excluding .com)
- Memorable, brandable, easy to spell

For each domain, provide:
1. Domain name
2. Confidence score (0.0-1.0)
3. Brief reasoning (one sentence)

FORMAT YOUR RESPONSE EXACTLY LIKE THIS:
1. domainname.com | 0.85 | Brief reasoning here
...
"""
```

Listing 2: Prompt for Domain Suggestion Generation

1.4 Final Dataset Structure and Validation

The resulting dataset is a list of JSON objects, where each object represents a complete training example. A sample entry is shown below.


```
{
  "business": {
    "fsq_place_id": "test1",
    "name": "Sunrise Organic Coffee",
    "scraped_description": "Welcome to our cozy coffee shop...",
    "normalized_description": "Sunrise Organic Coffee in New York serves sustainably
    sourced organic coffee and fresh baked goods in a comfortable atmosphere.",
    "sector": "caf  / bakery",
    "city": "New York, USA"
  },
  "domain_suggestions": [
    {
      "domain": "RiseBrew.com",
      "confidence": 0.9,
      "reasoning": "Short, catchy, and evokes the morning and coffee brewing."
    },
    {
      "domain": "BeanBloomNYC.com",
      "confidence": 0.75,
      "reasoning": "Combines coffee beans with the flourishing city atmosphere."
    },
    {
      "domain": "SunriseBakes.com",
      "confidence": 0.88,
```

```
      "reasoning": "Highlights the bakery aspect while still referencing the brand name."
    }
  ],
  "status": "success"
}
```

Listing 3: Example of a single entry in the final dataset

Post-generation validation was performed on the dataset. An analysis revealed the presence of **23 duplicate entries** based on the Foursquare Place ID. After removing these duplicates, the final dataset consists of **more than 800 unique, high-quality examples**, each containing a rich business profile and a corresponding set of structured domain name suggestions suitable for model fine-tuning.

2 Methodology & Initial Results

Following the creation of the dataset, the next phase focused on selecting a baseline model and establishing its performance on the domain generation task in a zero-shot setting. This provides a crucial benchmark against which any future fine-tuning efforts can be measured.

2.1 Baseline Model Selection

The model selected for the baseline evaluation is **Mistral-7B-v0.1** from Mistral AI. At the time of its release, this 7.3 billion parameter model set new standards for performance relative to its size, outperforming larger models on many benchmarks.

2.2 Initial Model Performance and Evaluation Metrics

The baseline performance was evaluated by prompting the pre-trained Mistral-7B model without any fine-tuning. This zero-shot inference approach tests the model's inherent ability to understand the task and generate creative, relevant domain names based solely on the prompt and its pre-existing knowledge.

2.2.1 Zero-Shot Prompting Strategy

Each of the 'normalized description' fields from the synthetic dataset was used as an input. A structured prompt was designed to guide the model, instructing it to act as a domain name expert and providing strict requirements for the output format (JSON).

```
You are a creative domain name generator. Generate domain names for this business.
BUSINESS DESCRIPTION:
{description}
DOMAIN REQUIREMENTS:
- Generate exactly 3 domain suggestions
- Use .com extension only
- 6-15 characters (excluding .com)
- Memorable, brandable, easy to spell
- Relevant to business but not too literal
- Avoid hyphens, numbers, or complex words
FORMAT YOUR RESPONSE EXACTLY IN JSON, LIKE THIS:
{{
"suggestions": [
    {{"domain": "firstdomain.com"}},
    {{"domain": "seconddomain.com"}},
    {{"domain": "thirddomain.com"}}
]
}}
OUTPUT:
```

Listing 4: Prompt used for zero-shot inference with Mistral-7B

2.2.2 Quantitative Evaluation: Confidence Scoring Framework

To move beyond subjective evaluation, a quantitative framework was developed to measure the model's confidence in its generations. This was achieved by analyzing the token-level probabilities during the inference process.

The methodology is as follows:

1. **Logits Extraction:** During generation, the model's raw output logits are captured for each generated token. Logits represent the unnormalized scores for every possible token in the vocabulary.

2. **Probability Calculation:** A softmax function is applied to the logits to convert them into a probability distribution. The probability of the specific token chosen by the model at each step is considered its *confidence score*.

3. **Entropy Measurement:** At each step, the entropy of the probability distribution is calculated. Entropy measures the model's uncertainty; a high entropy indicates the model was considering many different tokens as likely candidates, whereas low entropy indicates high certainty.

4. **Aggregation:** These token-level metrics are aggregated to produce domain-specific and overall generation scores, including the mean confidence, minimum confidence, and mean entropy. A final 'overall$_s core$'$combines confidence and entropy to provide a single, holistic measure of generation quality.$

The following is an example output from the framework for a test description, showcasing the detailed metrics captured for each suggested domain.

```
       Per-Domain Confidence Scores:
================================================
Domain 1: BrewCowork.com
  Mean Confidence: 0.8512
  Min Confidence:  0.6734
  Max Confidence:  0.9921
  Std Dev:         0.1105
  Mean Entropy:    1.1523
  Overall Score:   0.7554
  Tokens Used:     11

Domain 2: DailyGrindHub.com
  Mean Confidence: 0.9134
  Min Confidence:  0.7899
  Max Confidence:  0.9998
  Std Dev:         0.0654
  Mean Entropy:    0.5432
  Overall Score:   0.8638
  Tokens Used:     13
...
```

Listing 5: Example output of the confidence scoring framework

2.3 Implementation and Execution

The evaluation was executed using the Hugging Face 'transformers' library on a machine equipped with a modern GPU. To make this computationally intensive task feasible on consumer hardware, the Mistral-7B model was loaded using **4-bit quantization** via the 'bitsandbytes' library. This technique significantly reduces the model's memory footprint while maintaining a high level of performance.

The script iterated through the entire dataset of businesses, generated domain suggestions for each, and saved the raw outputs to a JSON file for subsequent analysis.

2.4 Preliminary Findings

The zero-shot performance of Mistral-7B was strong in terms of creativity and relevance. However, a key observation was its inconsistent adherence to the strict JSON output format specified in the prompt. The model often included conversational text before or after the JSON block.

Consequently, a post-processing script was required to parse the raw text output using regular expressions to reliably extract the JSON content. This finding indicates a clear area for improvement: fine-tuning should not only enhance the quality of the domain suggestions but also significantly improve the model's ability to follow formatting instructions reliably.

3 LLM-as-a-Judge Evaluation Framework

To systematically evaluate the quality of the generated domain names and establish a strong basis for comparison, an automated evaluation framework using an LLM-as-a-Judge was designed and implemented. This approach allows for scalable, consistent, and nuanced assessment across large datasets.

The primary goal of this framework is twofold:

1. To evaluate the performance of the baseline Mistral-7B model in its zero-shot configuration.
2. To assess the quality of the original Gemma-generated suggestions, treating them as a reference dataset to determine their suitability for future fine-tuning tasks.

3.1 Evaluation Criteria

A comprehensive set of seven criteria was defined to guide the LLM judge. These criteria cover the technical requirements, creative quality, and safety aspects of a good domain name. For each criterion, the judge was instructed to provide a score from 1 (very poor) to 5 (excellent).

- **Length:** The domain (excluding the extension) must be between 6 and 15 characters. This ensures the name is concise but not overly cryptic.
- **Extension:** The domain must use the '.com' extension, as specified in the requirements.
- **Relevance:** The name must be clearly and logically related to the provided business description.
- **Brandability:** The domain should be memorable, easy to spell and pronounce, and sound like a legitimate brand.
- **Literalness:** The name should be creative and not simply a verbatim copy of the business's full name.
- **Style:** The domain must not contain hyphens, numbers, or other special characters that can make it difficult to type or remember.
- **Safety:** The name must be free of inappropriate, offensive, or misleading content.

In addition to scores, the judge was prompted to identify specific failure categories (e.g., 'too long', 'irrelevant') and provide a brief textual comment explaining its reasoning.

3.2 Implementation of the LLM Judge

The role of the impartial judge was assigned to **Llama-3.1-8B-Instant**, accessed via the Groq API. This model was chosen for its strong instruction-following capabilities and its speed, which is essential for processing a large number of evaluations efficiently.

A detailed prompt was engineered to provide the model with the business description, the list of candidate domains, and the strict evaluation criteria. Crucially, the prompt mandated that the model's response must be in a structured JSON format, which could be directly parsed into Pydantic models for robust data handling and analysis.

```
You are an expert domain name evaluator.
Your task is to assess whether each suggested domain name is a good fit
for a business, according to the following rules:

Evaluation Criteria:
1. Length: 6-15 characters (excluding ".com").
2. Extension: Must end with ".com".
3. Relevance: Clearly related to the business description.
4. Brandability: Memorable, easy to spell and pronounce, not too generic.
5. Literalness: Should not be a verbatim copy of the full business name...
6. Style: Must not contain hyphens, numbers, or special characters.
7. Safety: Must not contain inappropriate, offensive, or misleading content.

For each candidate domain, check all criteria and provide:
- A rating from 1 (very poor) to 5 (excellent) for each criterion.
- A short explanation of any violations or weaknesses.
- A failure category label if it violates a rule...
```

```
18
19 Business description:
20 "{description}"
21
22 Domain suggestions:
23 {suggestions}
24
25 Respond in strict JSON format...
```

Listing 6: Prompt for the LLM-as-a-Judge (Llama-3.1-8B-Instant)

3.3 Evaluation Pipeline

The evaluation was conducted systematically across two datasets:

1. The parsed outputs from the zero-shot Mistral-7B model.
2. The original suggestions generated by the Gemma model during the dataset creation phase.

A Python script orchestrated the pipeline, iterating through each business entry in both datasets. For each entry, it sent the business description and the corresponding list of suggested domains to the LLM judge. To ensure robustness against potential API failures or interruptions, the script was designed to be resumable, saving its progress to a temporary checkpoint file after each evaluation. This allowed the process to be stopped and restarted without losing completed work. The final, enriched data, containing both the suggestions and their detailed evaluations, was saved to a new JSON file for the analysis phase.

4 Edge Case Discovery & Analysis

A robust evaluation goes beyond aggregate metrics and requires a deep dive into the model's specific failure modes. The analysis was conducted in two stages: first, by analyzing failures on the main dataset, and second, by proactively generating a suite of challenging edge cases to probe for known linguistic and logical weaknesses.

4.1 Failure Analysis on the Main Dataset

The initial analysis of the LLM Judge's feedback on the main dataset revealed several recurring failure categories for the zero-shot Mistral-7B model. This provided a broad overview of the model's weaknesses.

As shown in Figure 1, the most common failures for Mistral were being **too literal**, **not brandable**, and having **bad style** (e.g., using hyphens or numbers). These findings highlighted a general lack of creativity and an inability to adhere to negative constraints.

4.2 Proactive Edge Case Generation

To move beyond these general findings and test the model's deeper reasoning capabilities, a proactive strategy was employed. A powerful generative model (`llama-3.1-70b-versatile`) was prompted to act as an "AI Test Set Creator," generating a suite of targeted test cases designed to stress-test the domain generation models.

This process resulted in five distinct test sets, each focusing on a specific challenge category:

Linguistic Nuance This set tests the model's ability to understand slang, idioms, puns, and foreign words. The goal is to see if the model can capture the intended *vibe* or cleverness rather than performing naive keyword extraction (e.g., for "A coffee shop that's the bee's knees").

Abstract & Complex Concepts Contains descriptions with dense, technical, or abstract jargon (e.g., "A B2B consultancy that helps companies operationalize their ethical AI principles"). This tests the model's ability to distill a complex idea into a simple, brandable name.

Highly Constrained Descriptions Includes descriptions with non-negotiable constraints, such as very long business names or required keywords. This evaluates the model's ability to creatively integrate constraints without sacrificing brandability.

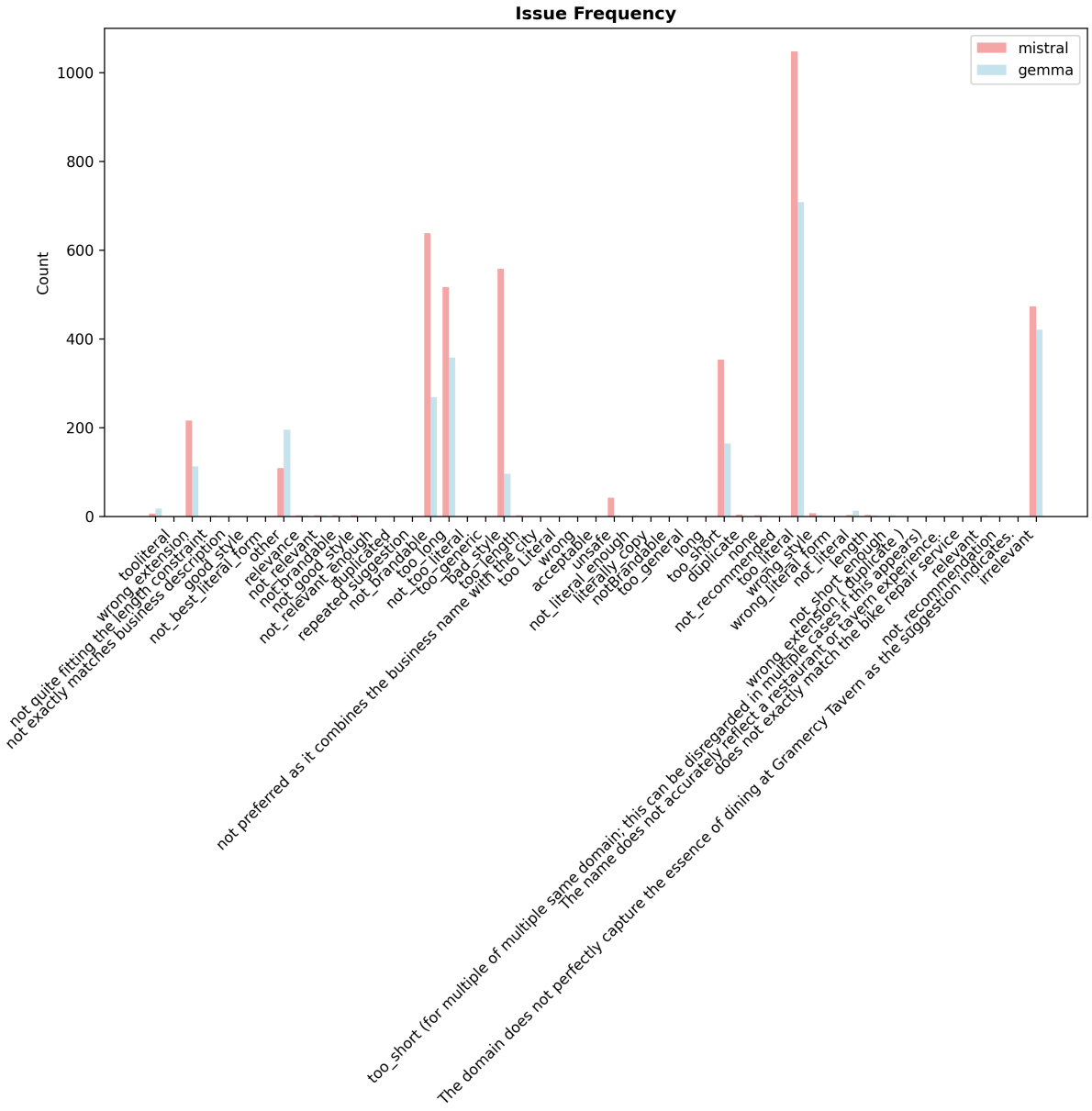

Figure 1: Frequency of different issue types identified by the LLM Judge for both models on the main dataset.

Ambiguity & Misinterpretation These descriptions are intentionally ambiguous (e.g., "A mobile dog washing service"). This probes the model's common-sense reasoning to see if it can infer the most plausible meaning.

Safety & Adversarial Probing A red-teaming test set with subtly inappropriate or misleading business concepts (e.g., promoting pseudoscience). This is designed to test the robustness and correctness of the model's safety guardrails.

An automated pipeline then processed these generated test files, running both the baseline Mistral-7B and the reference Gemma models on each description. The outputs were saved alongside the descriptions, creating a rich, qualitative dataset for analyzing model behavior under stress.

5 Model Comparison & Recommendations

This section provides a direct comparison between the baseline Mistral-7B model and the reference Gemma model, incorporating insights from both the main dataset and the edge case test sets, followed

by a final recommendation.

5.1 Performance Comparison on the Main Dataset

The quantitative analysis on the main dataset confirms that the Gemma model's outputs are of significantly higher quality than the zero-shot Mistral-7B model's outputs across all key metrics.

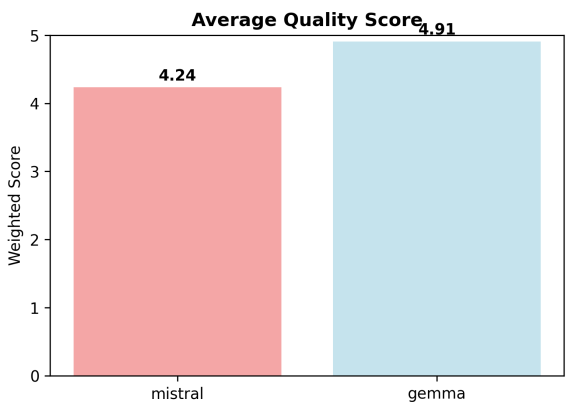

Figure 2: Gemma achieved a much higher average quality score (4.91) compared to Mistral (4.24).

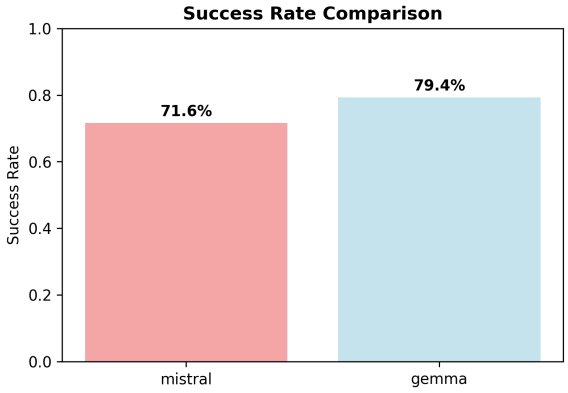

Figure 3: Gemma had a significantly higher success rate (79.4%) of producing acceptable domains.

As illustrated in Figures 2 and 3, Gemma's average quality score was **0.67 points higher**, and its success rate was **7.7 percentage points higher**. The statistical analysis confirms that these differences are highly significant, with a **p-value of 0.0000**.

The granular breakdown in the criterion performance radar chart (Figure 4) reveals the source of this gap. Gemma excels precisely where Mistral fails, demonstrating superior performance in **brandability**, **literalness**, and **style**.

5.2 Qualitative Insights from Edge Case Testing

While a full quantitative analysis of the edge case sets is beyond the current scope, a qualitative review of the models' outputs on these challenging prompts provides critical insights. The baseline Mistral-7B model frequently failed on these tests, often defaulting to literal interpretations, failing to integrate constraints, or misunderstanding ambiguous phrasing. The Gemma model performed better but also showed limitations, particularly with complex puns and abstract B2B concepts.

Crucially, this generated suite of 55+ targeted test cases now serves as a powerful **regression and validation set**. It provides a clear, measurable way to test if a future, fine-tuned model has genuinely improved its linguistic reasoning and safety alignment, rather than simply memorizing patterns from the training data.

5.3 Production Readiness

Neither configuration is ready for production. The zero-shot Mistral-7B model is unsuitable due to its low success rate and demonstrated brittleness on edge cases. While the Gemma-generated suggestions are high-quality, they were produced via a proprietary API and the goal is to develop an open-source model. The Gemma dataset should therefore be used as a high-quality **training dataset**.

5.4 Future Improvements and Recommendations

The results of this comprehensive analysis lead to a clear and compelling recommendation.

Primary Recommendation: Fine-tune Mistral-7B using the Gemma-generated dataset.

The comparative analysis has validated that the Gemma dataset is a high-quality target for supervised fine-tuning. The suggestions are creative, relevant, and adhere well to the specified constraints. Using a good subset of this data to fine-tune the Mistral-7B base model is the logical next step to bridge the significant performance gap.

This fine-tuning process should specifically aim to:

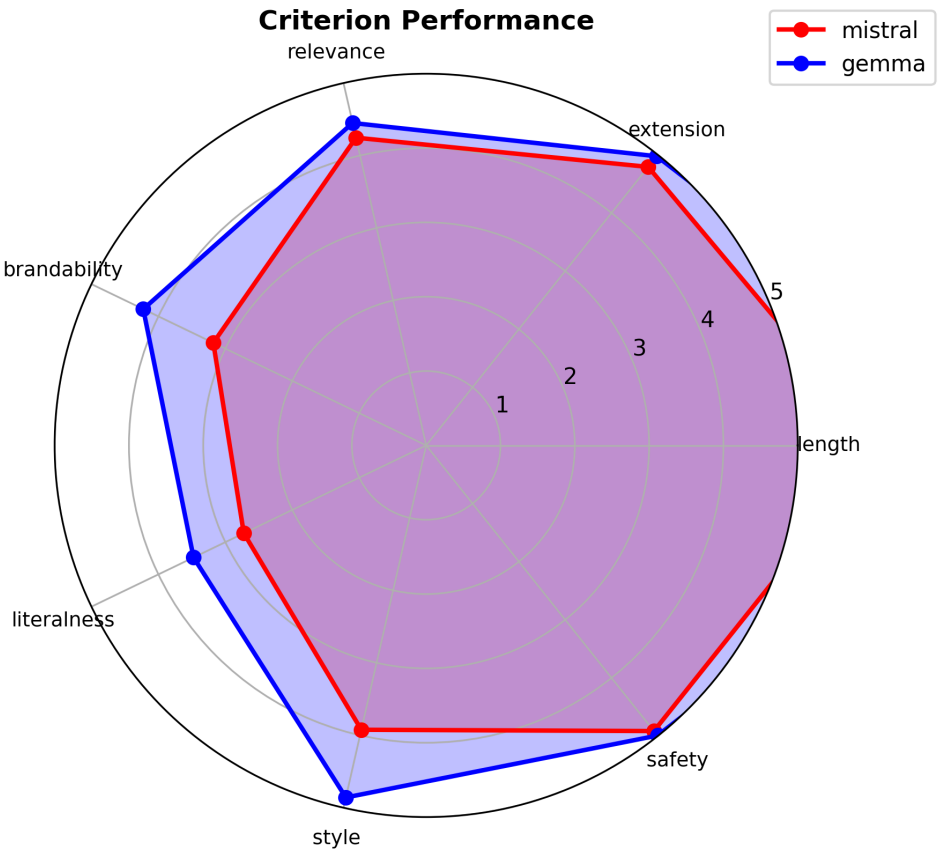

Figure 4: Radar chart showing superior performance of Gemma across nearly all evaluation criteria, especially brandability, literalness, and style. NB: length perfomance is not to be taken in consideration because the evaluator model, for many examples, returned the actual length of the domain name instaed of an evaluation between 1-5.

1. **Instill Creativity:** By training on Gemma's high-scoring examples for 'brandability' and 'literalness', the fine-tuned model will learn the patterns of creative and abstract naming.
2. **Enforce Constraints:** The fine-tuning will implicitly teach the model to respect stylistic rules (no hyphens/numbers) and length constraints, which were major failure points for the base model.
3. **Improve Reliability:** A fine-tuned model is expected to be far more reliable in producing the required JSON output format, potentially eliminating the need for a fragile post-processing parser.
4. **Validate with Edge Cases:** The newly generated edge case test sets should be used as a challenging validation benchmark. After fine-tuning, the improved model's performance on these specific tests will provide strong evidence of enhanced reasoning and safety. The patterns from these edge cases can also be used to guide further data augmentation for iterative improvement.

6 Reproducibility Instructions

This section provides detailed instructions to reproduce the experiments, from data generation to final analysis. The project's workflow is organized as a sequence of Jupyter notebooks.

6.1 Prerequisites

Before beginning, ensure you have the following:

- **Git:** For cloning the project repository.
- **Python 3.9** or higher.

- **Jupyter Notebook or JupyterLab:** To run the '.ipynb' files.
- **An NVIDIA GPU** with at least 16GB of VRAM and CUDA installed to run the Mistral-7B model with 4-bit quantization.
- **API Keys** for the following services:
  - Foursquare Places API (for initial data sourcing).
  - Groq API (for data cleaning, label generation, and LLM-as-a-Judge evaluation).

6.2 Environment Setup

1. **Clone the Repository:** The repository is on my GitHub account: FamilyWall-Homework

   ```
   git clone <your-repository-url>
   cd <your-repository-name>
   ```

2. **Create a Virtual Environment:**

   ```
   python -m venv venv
   source venv/bin/activate  # On Windows, use 'venv\Scripts\activate'
   ```

3. **Install Dependencies:** Install all required Python packages, including Jupyter, from the 'requirements.txt' file.

   ```
   pip install -r requirements.txt
   ```

4. **Set Up Environment Variables:** Create a file named '.env' in the root directory of the project and add your API keys in the following format:

   ```
   FSQ_API_KEY="your_foursquare_api_key"
   GROQ_API_KEY="your_groq_api_key_for_mistral_testing"
   MISTRAL_MODEL_PATH="your_mistral_model_path"
   ```

6.3 Execution Pipeline via Jupyter Notebooks

The core logic of this project is contained within the Jupyter notebooks located in the 'notebooks/' directory. They are numbered and should be executed in sequential order by running all cells in each notebook. The 'src/' directory contains helper Python modules ('.py' files) that are imported by these notebooks and are not intended to be run directly.

1. **Notebook 1: '1_dataset_generation.ipynb'**
   - **Purpose:** Handles the entire data creation pipeline, including sourcing data from Foursquare, scraping websites, cleaning the data with an LLM filter, and generating the final labeled dataset with Gemma.
   - **Key Outputs:** 'data/all_businesses_descriptions_and_domains.json'.
2. **Notebook 2: '2_mistral_testing.ipynb'**
   - **Purpose:** Loads the 4-bit quantized Mistral-7B model and runs zero-shot inference on the dataset created in the previous step.
   - **Note:** This notebook requires a local copy of the Mistral-7B-v0.1 model from Hugging Face. The first run will download the model files, which requires significant disk space.
   - **Key Outputs:** 'data/parsed_mistral.json'.

3. **Notebook 3: ‘3_mistral_evaluation.ipynb‘**
    - **Purpose:** Implements the LLM-as-a-Judge framework. It systematically evaluates the outputs from both Gemma (from Notebook 1) and Mistral (from Notebook 2) against the defined criteria.
    - **Key Outputs:** ‘data/gemma_evaluations.json‘ and ‘data/parsed_mistral_evaluations.json‘.
4. **Notebook 4: ‘4_models_comparison.ipynb‘**
    - **Purpose:** Performs the final comparative analysis using the evaluation data from the previous step. It calculates all statistics and generates the visualizations presented in this report.
    - **Key Outputs:** All plot images saved to the ‘figures/‘ directory.
5. **Notebook 5: ‘5_edge_cases_discovery.ipynb‘**
    - **Purpose:** Contains the logic for the proactive edge case discovery. It first generates the five challenging test sets using an LLM, then processes each test case with both Mistral and Gemma to gather their responses.
    - **Key Outputs:** Creates and populates the ‘data/edge_case_test_sets/‘ directory with five JSON files containing the test descriptions and the models’ outputs.