PIndicQA: Physical Indic QA dataset

Sayambhu Sen
Amazon Alexa
sensayam@amazon.com

Sona Elza Simon
IIT Bombay
sona.simon@iitb.ac.in

Snegha A.
IIT Bombay
23m2160@iitb.ac.in

Abstract

The existing physical reasoning dataset is predominantly Western-centric, relying on customs and scenarios that fail to capture global cultural nuances. This bias limits the fair and accurate evaluation of reasoning capabilities in multilingual models. To address this gap, we introduce PIndicQA, a novel, culturally-localized physical reasoning benchmark for five Indic languages: Hindi, Bengali, Tamil, Kannada, and Malayalam. We created this dataset using a new pipeline that combines LLM-based taxonomy generation, embedding-based clustering to discover semantic categories, and localized sample generation. Our core contribution is a rigorous, multi-stage human-in-the-loop (HITL) process where native-speaking authors corrected and validated every machine-generated and translated sample to ensure high linguistic and cultural accuracy. The result is a challenging new benchmark designed to test a model's understanding of localized, non-Western physical commonsense, providing a vital resource for advancing multilingual reasoning.

1 Introduction

The original Physical Reasoning dataset (Bisk et al., 2020) contains samples which are almost always English specific datasets with samples very specific to Western customs, cultures and lifestyles. We know that because of the predominance of English specific datasets in pretraining most LLMs also have stronger Western specific knowledge (Simon et al., 2025). To that we propose that there are certain invariant meta-labels specific to each sample which is true regardless of the cultural aspects of the dataset. This method of using LLM generated meta-labels is a technique which we have found to increase diversity in samples. For translations, first translations were almost always bad, so getting them corrected by a multilingual LLM was a strong way of improving the model results. There have been multiple corrections on each of the changes while also making sure that for repeated samples, we reject them and get new samples generated and corrected. The important part of our technique is our localization step that creates a much more diverse dataset with unique ideas. Since, most LLMs have low knowledge about local cultural techniques (Simon et al., 2025), there is a lot of nuance in the way physical reasoning happens around local items, which is bound to create confusion in LLMs especially in non-English settings. Fine level annotations were done by us researchers while we let two or three more speakers of the traditional languages review samples of the multilingual generated text so that we can get general feedback about what works and what doesn't.

In the next few sections we will cover each part of the data creation process :

1. Open meta-label and taxonomy generation.
2. Clustering and closed classification.
3. Label sampling and in-context English sample generation.
4. Localization and translation.
5. Translation correction.

2 Data Generation and Correction

We show the full flowchart of the meta-data based synthetic data generation workflow in Figure 1. A detailed example of a single row of the dataset is given in Table 1.

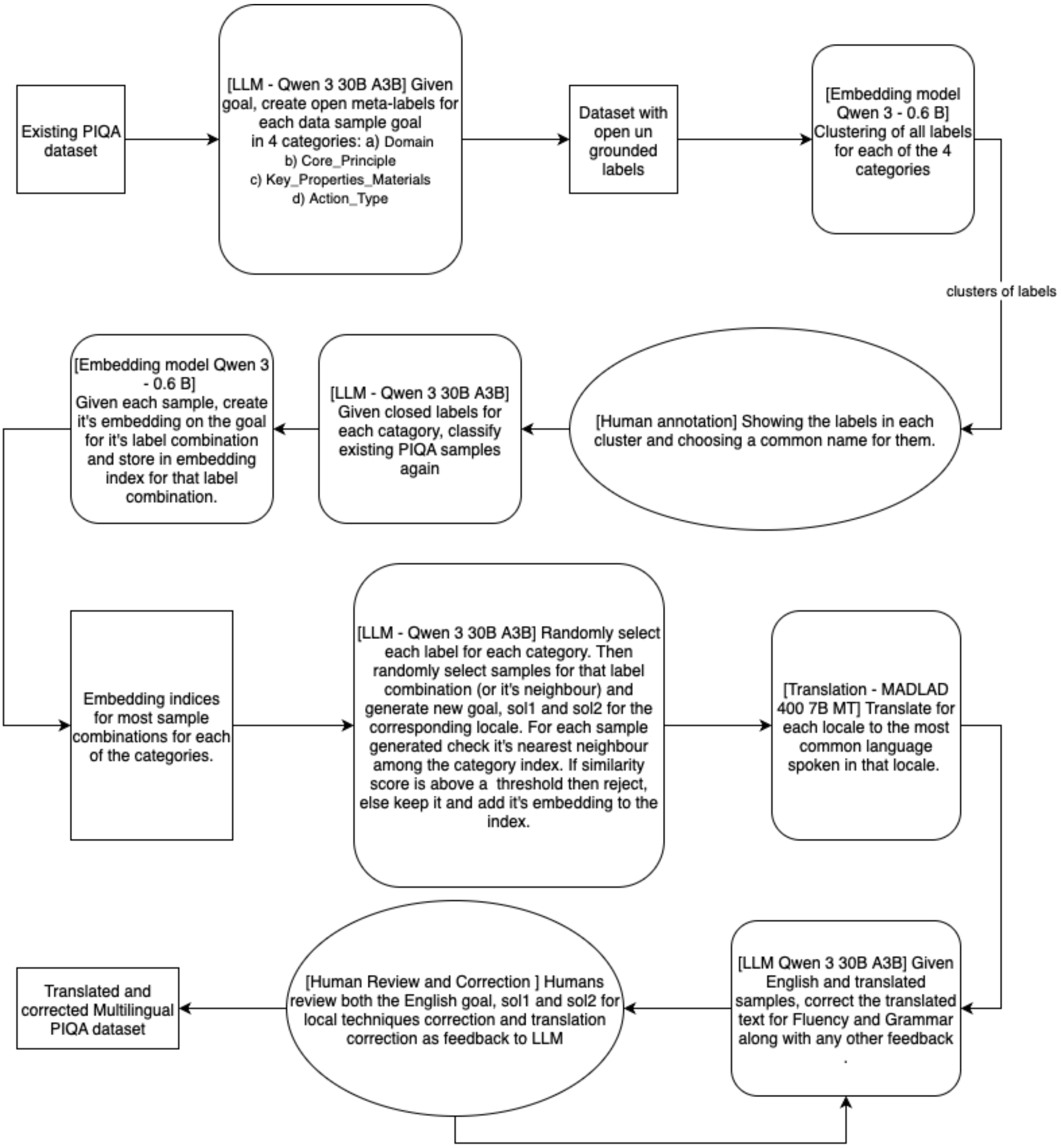
Existing PIQA dataset
[LLM - Qwen 3 30B A3B] Given goal, create open meta-labels for each data sample goal in 4 categories: a) Domain b) Core_Principle c) Key_Properties_Materials d) Action_Type
Dataset with open un grounded labels
[Embedding model Qwen 3 - 0.6 B] Clustering of all labels for each of the 4 categories
clusters of labels
[Human annotation] Showing the labels in each cluster and choosing a common name for them.
[LLM - Qwen 3 30B A3B] Given closed labels for each catagory, classify existing PIQA samples again
[Embedding model Qwen 3 - 0.6 B] Given each sample, create it's embedding on the goal for it's label combination and store in embedding index for that label combination.
Embedding indices for most sample combinations for each of the categories.
[LLM - Qwen 3 30B A3B] Randomly select each label for each category. Then randomly select samples for that label combination (or it's neighbour) and generate new goal, sol1 and sol2 for the corresponding locale. For each sample generated check it's nearest neighbour among the category index. If similarity score is above a threshold then reject, else keep it and add it's embedding to the index.
[Translation - MADLAD 400 7B MT] Translate for each locale to the most common language spoken in that locale.
[LLM Qwen 3 30B A3B] Given English and translated samples, correct the translated text for Fluency and Grammar along with any other feedback .
[Human Review and Correction] Humans review both the English goal, sol1 and sol2 for local techniques correction and translation correction as feedback to LLM
Translated and corrected Multilingual PIQA dataset

Figure 1: The data generation and translation pipeline for PIndicQA.

Table 1: A detailed example of a single data sample from the PIndicQA pipeline.

Field (Column Name)	Example Value
Nearest Neighbor Goal	How to make rice crispy treats.
Goal	How to properly cook and remove husk from puffed rice (muri) for a traditional Bengali snack?
Sol1	Spread fresh puffed rice (muri) on a clean cotton cloth over a wide banana leaf. Place it under direct sunlight for 20 minutes to dry slightly, then gently rub with hands to loosen and remove any remaining husk particles.
Sol2	Spread fresh puffed rice (muri) on a steel plate and place it under direct sunlight for 20 minutes to dry slightly, then gently rub with hands to loosen and remove any remaining husk particles.
Label	0
Domain	['Culinary Activities']
Core Principle	['Culinary Heat Application']
Key Properties Materials	['Cereal-Based Products & Textural Characteristics']
Action Type	['Direct Manipulation & Extraction']
Locale	West Bengal, India
Source	synthetic searched few shot (modified)
Translated Language	Bengali
Translated Goal	কিভাবে সঠিক ভাবে রান্নাকরতে হয় এবং কিভাবে ঝোলথেকে ঝোল বের করতে হয়?
Translated Sol1	একটি পরিষ্কার কাগজের কাপড়েঝুলিয়ে একটি বড় কাঁচা কাঁচা চাল(মুরি) ছড়িয়ে দিন। এটি-কে সূর্যেরআলোতে ২০ মিনিট রেখে কিছুটাশুকিয়ে নিন। তারপর হাত দিয়েহালকা ভাবে ঘষুন যাতে ময়দাটিছিঁড়ে যায় এবং বাকি ময়দাটি সরিয়েফেলুন।
Translated Sol2	স্টিলের প্লেটে তাজা ভাজা চাল (মুরি) ছড়িয়ে দিন এবং সূর্যের আলোতে ২০ মিনিট রেখে দিন যাতেকিছুটা শুকিয়ে যায়, তারপর হাত দিয়ে হালকা ভাবে ঘষুন যাতে ঝুলে থাকা ভুসির অংশগুলোবের হয়ে যায়।
Corrected Goal	ঐতিহ্যবাহী বাঙালি নাস্তার জন্য মুড়ি সঠিকভাবে ভেজে খোসা ছাড়ানোর উপায় কী?
Corrected Sol1	একটি চওড়া কলাপাতার উপর একটি পরিষ্কার সুতির কাপড় বিছিয়ে, তার উপর টাটকা মুড়ি ছড়িয়ে দিন। এগুলিকে ২০ মিনিটের জন্য সরাসরি সূর্যের আলোতে রেখে সামান্য শুকিয়ে নিন। এরপর, হাত দিয়ে আলতো করে ঘষে লেগে থাকা অবশিষ্ট তুষের কণা ঝেড়ে ফেলুন।
Corrected Sol2	একটি স্টিলের থালায় তাজা মুরি ছড়িয়ে সরাসরি সূর্যের আলোতে ২০ মিনিট রাখুন, যাতে সামান্য শুকিয়ে যায়। এরপর হাত দিয়ে আলতো করে ঘষে অবশিষ্ট তুষের কণাগুলো সরিয়ে ফেলুন।

2.1 Open Meta-label and Taxonomy Generation

In this section, we use meta-labels as contextual guidance for the generation of new data. We did this using a mix of human design and LLM generated labelling. We initially used an LLM (Gemini 2.5 Pro) (Comanici et al., 2025) for figuring out the classification taxonomy to be generated. We started by creating a prompt such that the LLM is given English samples of PIQA and that there are two classification categories 'Domain' and 'Core Principle'. The LLM should create further hierarchical classification categories. Of the many classification categories suggested, we selected the 'Key Properties Materials' and 'Action Type' as the two final categories.

Now that the classification categories are created, we use an LLM for open meta-label generation. We use Qwen3-30B-A3B (Yang et al., 2025) non-thinking for the meta-label generation. We use the A.1.1 prompt to classify the existing PIQA datasets to generate open meta-labels. As is the nature of non-deterministic LLMs, it generates meta-labels without any grounding. So it generates many hundreds of labels, many of which are similar.

2.2 Clustering and Closed Classification

Once all the labels are generated, for each category we use Qwen-3-Embedding-0.6B (Zhang et al., 2025) to generate embeddings for these labels, and then apply k-means clustering with the ELBO method to obtain clusters.

We cluster the labels for each of the four categories and present the resulting clusters to human annotators. By creating at most 2030 clusters, we obtain meaningful semantic groupings of labels. For each cluster, an LLM generates suggestions for a common name, from which the annotator either selects one or provides a new label (see Appendix A.1.2). Once all clusters are labeled, we annotate the existing samples accordingly and construct embedding indices for each unique label combination. This process defines our classification taxonomy, consisting of 19 classes for 'Domain', 32 for 'Core_Principle', 39 for 'Key_Properties_Materials', and 32 for 'Action_Type' (see Appendix A.1.3).

Using this taxonomy, we perform closed classification of the English PIQA dataset (Bisk et al., 2020) with the prompt shown in Appendix A.1.4 and the Qwen-3-30B-A3B non-thinking model. After classification, we enumerate all category combinations and observe that most combinations are empty, due to the four-category structure, resulting in a sparse set of embedding indices. We build indices for each non-empty label combination, which are then used both for in-context example retrieval and for similarity search in generated samples.

2.3 Label Sampling and In-Context English Sample Generation

Now that we have our label taxonomy of four categories along with the embedding indices for label combinations of the existing datasets, we perform our English specific data generation which is to be translated into the local language later.

The main idea behind creating those samples is that we sample labels from each of the categories and then check if there exists an index for that label combination. Then we generate new samples for that label combination of the 4 categories using the prompt for generation (see Appendix A.1.5). This prompt helps in creating new samples for each locale. Thus the samples created are geared for each locale with 5 locales for Indic category Hindi, Bengali, Kannada, Tamil and Malayalam for each of respective states where it is spoken by the majority. (P.S. These languages were chosen since these languages are the ones known to the authors of this paper.) For Hindi, which is spoken across multiple Indian states, we chose Delhi as the localization example for it.

If an index exists for a given label combination, we sample directly from it to provide in-context examples. Otherwise, we use the closest available label combination with existing indices. After generating new samples, we create embeddings for the “goal" field only and add them back to the category index. This process is repeated until 100 samples are created for each locale.

During this process, we observed that some generated goals were repetitive. To address this, each set of 100 samples per locale underwent human review to filter out repetitions, and additional samples were generated as needed. Another common error was that the generated solutions were either identical or both incorrect. In such cases, human reviewers curated one correct and one incorrect solution for each sample. Each sample was reviewed by two annotators to check for mistakes.

2.4 Localization and Translation

Localization is done in the data generation step itself so that the translations are more culturally and geographically relevant to the region where it is spoken. It also helps in making the translation more natural and more unique since most samples in the original PIQA dataset is very Western in focus. By putting the focus on Indic locales, we generate unique samples with very specific cultural knowledge known only to Indic local people.

Once these localized generated texts are validated, we now perform translation in two steps: translation by a translation model and translation correction by a relatively strong multilingual model. We first use Googles MADLAD-400-7B model (Kudugunta et al., 2023) as the translation model, as it was found to be more accurate for Indic translations than Metas NLLB model (Team et al., 2022). This observation was confirmed by our human reviewers, particularly for Bengali and Kannada, and the results were found to be comparable for the other languages as well.

2.5 Translation Correction

The initial machine translations were often flawed, exhibiting artifacts such as garbage tokens, repetitive phrases, and grammatical errors. To mitigate this, we employed a second multilingual LLMeither Qwen-3-30B-A3B or Gemma-3-27B-it (Team et al., 2025) providing it with both the original English source and the flawed translation as context to generate a corrected version.

While this automated pass corrected many obvious flaws, a significant number of errors remained. This necessitated a human-in-the-loop approach where our fluent human reviewers prompted the correction LLM with specific lexical choices and grammatical fixes. To refine this process further, we sampled approximately 30 common errors per language and used these human-corrected examples as few-shot exemplars. These were added back into the correction prompt, along with explicit instructions detailing common error types to avoid and specific stylistic guidelines. This exemplar-guided approach significantly reduced the error rate in the remaining samples. Finally, the remaining 20 samples per language that still contained minor mistakes were either corrected manually by our annotators or finalized by prompting the LLM for one last correction.

3 Limitations and Strengths

The data generation process using meta-data as context turned out to be very useful in generating diverse samples along with localization to create unique scenarios which only local people will have unique knowledge of. One way to create more diverse samples would be to create some more meta-categories like, say, scenarios like Onam in Kerala or Poila Boishakh (Bengali New Year, 1st of Baisakh month) to create unique scenarios. We did not have time to explore them.

One reason this approach worked well is that humans extensively checked the meta-categories to ensure that the LLM-generated labels, as well as the common names for clusters of labels, were properly identified. With appropriate meta-data labeling, the need for human supervision of individual samples particularly for detecting repetition was significantly reduced. Another challenge was that the LLM sometimes produced incorrect answers due to its lack of knowledge about local customs or materials. In such cases, the human annotator's familiarity with local practices was crucial. However, there were instances where even the annotators themselves were unsure, which required them to search online to find the correct answer.

Acknowledgments

I would like to thank IIT-B and Amazon for the collaboration that made this paper possible. I would like to especially thank my co-authors for their help in verifying and correcting each and every data point. I would especially like to thank Mr. Sudhir R Narayana who helped in verifying and correcting the Kannada data very extensively.

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and Luke Marris and. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: a multilingual and document-level large audited dataset. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Sona Elza Simon, Soumen Kumar Mondal, Abhishek Singhania, Sayambhu Sen, and Preethi Jyothi. 2025. LoFTI: Localization and factuality transfer to Indian locales. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16635–16662, Vienna, Austria. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 6 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler

Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

A Appendix

A.1 Prompts

A.1.1 Open Meta-Labelling Prompt

```
Your task is to analyze a given text and
generate a structured JSON object
containing meta-labels that describe it.

The JSON object must have the following
four keys: "Domain", "Core_Principle",
"Key_Properties_Materials", "Action_Type"


For each key, provide an array of relevant
string labels.

Do not provide more than three labels for
any given key.

The output must be a single, valid JSON
object.

Label Definitions:

Domain: The general field or area the
text belongs to (e.g., "Cooking",
"Social Media").

Core_Principle: The fundamental concepts,
theories, or ideas involved (e.g.,"Heat
Transfer", "Digital Navigation").

Key_Properties_Materials: The primary
objects, substances, or components and
their key characteristics (e.g.,
"Water (Liquid)", "Software Application").

Action_Type: The main actions or verbs
described in the text (e.g., "Boiling",
"Scrolling").

Example 1:

text: "How to boil eggs"

Meta-labels:

{{
  "Domain": ["Kitchen", "Cooking"],
  "Core_Principle": ["Heat Transfer",
  "Thermodynamics"],
  "Key_Properties_Materials": ["Water
  (Liquid)", "Egg (Fragile)"],
  "Action_Type": ["Heating", "Boiling"]
}}

Example 2:

text: "How to check your Facebook feed"

Meta-labels:

{{
  "Domain": ["Digital", "Social Media"],
 "Core_Principle": ["Digital Navigation",
  "UI Design"],
  "Key_Properties_Materials": ["Software
  Application", "Interface"],
 "Action_Type": ["Clicking", "Scrolling",
  "Browsing"]
}}

Now, generate the meta-labels for the
following text:

text: {input}

Meta-labels:
```

A.1.2 Cluster Labelling

This is an example of cluster labelling

```
Category: Domain
Cluster labels :
['herbs','dressing','hospital',
'dressing']
```

For these cluster labels printed, user prompts the LLM to suggest some common names. In this case, the user wrote “health_care" but the suggestions were similar like ‘Healing’ and ‘Health’.

A.1.3 Final Closed Taxonomy

```
{
"Domain": [
    "health_care",
 "Industrial Processes & Engineering",
    "Culinary Activities",
    "Animal & Character Management",
    "Lifestyle & Social Systems",
   "Home Gastronomy & Food Science",
    "Digital Media & Interaction",
    "Human Biological Sciences",
    "Office & Productivity Tools",
    "Holistic Personal Development",
    "Home Ambiance & Design",
    "Domestic Upkeep & Maintenance",
    "art and design",
    "Outdoor Life & Safety",
    "Textile activities",
   "Earth & Environmental Sciences",
    "DIY & Home Crafting",
    "Aesthetics & Identity",
    "Food Production & Technology"
],
"Core_Principle": [
 "Somatic Maintenance & Presentation.",
    "Hydration Regulation",
 "Structural Transformation & Design",
    "Targeted State Control",
 "Protective Measures & Hazard Control",
 "Engineering for Performance & Safety",
    "Material Integrity & Surface
    Interaction",
 "Gas Dynamics & Airflow Management",
    "Molecular Composition &
    Transformation",
 "Ecological Stewardship & Restoration",
    "Biochemical Sensory Analysis",
    "Interface-Driven Manipulation",
    "Clarity-Driven Optimization",
 "Foundational Stability & Validation",
    "Strategic State Management",
 "Sanitation & Resource Stewardship",
  "Transactional System Enablement",
  "Sensory Composition & Synthesis",
  "Knowledge Encoding & Application",
  "Systematic Resource Optimization",
    "High-Resolution Material
    Modification",
   "Systematic Navigation & Routing",
    "Human-Centered System Design",
    "Therapeutic Comfort & Symptom
    Alleviation",
    "Structural Assembly & Material
    Transformation",
  "Dynamic Interaction & Kinematics",
 "Sensorimotor Control & Adaptation",
   "Fluid Mechanics & Mass Transfer",
    "Adaptive Reconfiguration",
 "Nutritional Energetics & Metabolism",
  "Thermodynamics & Heat Management",
    "Applied Optics & Illumination"
],
"Key_Properties_Materials": [
    "Adhesive & Sealing Properties",
    "Entity & Defining Attribute",
    "Botanical Products & Sensory
    Attributes",
 "Structural & Surface Characteristics",
   "Textile & Functional Properties",
   "Viscous & Oleaginous Substances",
    "Culinary Components & Sensory
    Profiles",
   "Cereal-Based Products & Textural
    Characteristics",
 "Tools & Defining Functional Attributes",
 "Interactive Surfaces & Environments",
    "Manufactured Goods & Defining
    Properties",
    "Aqueous & Solvent Properties",
   "Anatomical Components & Wearable
    Properties",
  "Information Carriers & Paper-Based
    Goods",
 "Food Products & Compositional States",
    "Metallic Materials & Physical
    Properties",
  "Confectionery & Phase Properties",
    "Gaseous & Reactive Properties",
   "Navigational Cues & Indicators",
   "Biological & Fibrous Structures",
 "Structural Rigidity & Flexibility",
  "Colorants & Pigmented Substances",
 "Animate Beings & Related Consumables",
 "Aromatic & Structural Ingredients",
 "Storage Units & Defining Features",
    "Liquids, Containers, & Chemical
    Properties",
    "Mechanical Systems & Functional
```

```
        Components",
      "Optical & Mechanical Properties",
      "User Interface & Control Points",
      "Engineered Materials & Functional
        Design",
        "Biological Sensors & Perceptual
        States",
      "Particulate & Suspended Matter",
        "Social & Community Entities",
        "Textual Information &
        Representations",
      "Hardware Components & Functional
        Attributes",
        "Digital Records & Systems",
        "Environmental & Foundational
        States",
     "Environmental Alerts & Nuisances",
        "Applied Materials & Physical
        States"
],
"Action_Type": [
        "Joining & Fastening",
        "Data & System Refinement",
     "Direct Manipulation & Extraction",
        "Pathfinding & Boundary
        Manipulation",
        "Resource Management &
        Organization",
        "Physical State Manipulation",
        "Observation & Analysis",
        "Composting & Recycling",
        "Energy Application & Physical
        Transformation",
        "Influence & Counteraction",
        "Acquisition & Processing",
        "Safeguarding & Preservation",
      "Material Handling & Preparation",
        "Excavation & Displacement",
        "Active State Control &
        Manipulation",
    "Systematic Analysis & Organization",
        "Thermal State Transition",
      "Surface Purification & Removal",
     "Scholarly & Formal Communication",
        "Culinary Heat Application",
        "Modification & Adaptation",
        "Logistics & Transfer",
        "Precision Shaping & Grooming",
        "Data Entry & Sensory Testing",
        "Mechanical Compression &
        Deformation",
        "Surface Care & Maintenance",
        "Kinetic Activity & Locomotion",
        "tructural Adjustment &
        Configuration",
        "Tactile Control & Handling",
    "Visual Creation & Representation",
        "Placement & Installation",
        "Aesthetic Design & Fabrication"
    ]
}
```

A.1.4 Classification Prompt

```
Your task is to classify the given text
into a single, most appropriate label
for each of the following categories.
You MUST choose exactly one label from
the provided list for each category.
The output must be a single, valid JSON
object.

Categories and Approved Labels:
{taxonomy_json}

Now, classify the following text:
text: {input}
Classification:
```

A.1.5 Localization Generation Prompt Function

```
def build_prompt(metadata: Dict,
exemplars: List[Dict], locale:
str, num_to_generate: int = 5)
-> str:
"""Constructs the prompt for LLM
generation, requesting multiple
localized examples."""
prompt = (
    "You are generating data for a
  physical reasoning dataset (PIQA).
    \n"
    f"**Crucially, the goal, sol1,
    and sol2 MUST be adapted to the
    context of the following locale:
    {locale}.** "
    "Use regional items, scenarios,
  places, and cultural nuances where
    appropriate.\n\n"
    "The examples must also strictly
    adhere to the following abstract
    classification:\n"
  f"Domain: {metadata.get('Domain')}
    \n"
    f"Core Principle:
        {metadata.get('Core_Principle')}
        \n"
        f"Key Properties/Materials:
        {metadata.get
        ('Key_Properties_Materials')
        }\n"
        f"Action Type:
        {metadata.get('Action_Type')}\n
        \n"
        "Here are some examples of the
        structure, but do NOT copy their
        content. Your goal is to create
        NEW, localized versions:\n"
    )

    for ex in exemplars:
        prompt += json.dumps({k: ex[k]
       for k in ('goal', 'sol1', 'sol2',
        'label')}) + "\n"

    prompt += (
      f"\nGenerate {num_to_generate} new,
      diverse, and localized examples for
        **{locale}**. "
       "Do NOT repeat the examples above.
        \n"
      'Output MUST be a JSON list containing
      objects with "goal", "sol1", "sol2",
        and "label" (0 or 1).\n'

       "Do not output any text or markdown
       formatting before or after the JSON
        list.\n"
      "Example Output Format: [{\"goal\":
       ..., \"sol1\": ..., \"sol2\": ...,
        \"label\": ...}, ...]"
    )
    return prompt
```

A.1.6 Translation Correction Prompt

```
    f"You are a language expert fluent in
    English and {language}. "
    f"The following text was machine-
   translated from English to {language}."
   "Review the translation and rewrite it
    only if it is necessary to be more
    natural, fluent, and grammatically
    correct, "
   "while preserving the original meaning
    in English. Output only the corrected
    text and nothing else.\n\n"
    f"Original English:\n\"
    {original_text}\"\n\n"
   f"Machine Translation to {language}:\n\"
    {translated_text}\"\n\n"
    f"Corrected {language} Version:"
```