This is a good indicator of which datasets to look for (specifically for the NLP functionality of the model - no TTS or sound datasets included) and which to eliminate from our already existing ones - along with a good training workflow explanation for how everything will come together.

**Phase 1: Core Knowledge & Pre-training**

🔹 **Goal:** Build foundational **STEM understanding** with **clean, structured text** before fine-tuning.  
🔹 **Challenges:**

* Research papers are **noisy** (citations, funding notes, redundant explanations).
* Need **structured, high-quality data** without excessive experimental comparisons.

**✅ Datasets & Sources for Pre-training:**

| **Category** | **Sources** | **Notes** |
| --- | --- | --- |
| **General STEM Knowledge** | Wikipedia (STEM subset), OpenAI’s The Pile (STEM sections), Project Gutenberg (classic science books) | Clean but broad—ensures general domain knowledge |
| **Mathematical & Scientific Reasoning** | GSM8K, MATH, DeepMind Math, SciQA | Focused problem-solving datasets |
| **Computer Science & Coding** | Stack Overflow, GitHub (curated repos), HumanEval, LeetCode | Code comprehension & generation |
| **Research Papers (Filtered for Noise)** | ✅ **S2ORC** (Semantic Scholar Corpus) ✅ **ArXiv CS/Physics subset** ✅ **The Pile (ArXiv section)** | Prefiltered & structured research content |
| **Semantic Knowledge Base Integration** | Wikidata, ConceptNet, SciBERT embeddings | Helps with common sense & logical reasoning |

🔹 **Cleaning & Filtering Strategy for Papers:**

* **Extract abstracts + conclusions** (ignore redundant comparisons).
* **Filter by discipline:** Math, CS, Physics, Engineering (skip Bio-heavy content).
* **Use NLP-based summarization** (e.g., BART, GPT-4) to strip unnecessary citations & experiment details.

**Phase 2: Explainability & Refinement**

🔹 **Goal:** Improve clarity, explanation depth, and adaptability to different users.  
🔹 **Challenges:**

* Need to balance **technical accuracy vs. explainability**.
* Must avoid **over-simplifications** while keeping things engaging.

**✅ Datasets & Sources for Explainability:**

| **Category** | **Sources** | **Notes** |
| --- | --- | --- |
| **Human-like Explanation** | ELI5 (Explain Like I’m 5), Natural Instructions v2 | Helps model adapt explanations to different audiences |
| **Answer Refinement** | Anthropic HH-RLHF, OpenAI InstructGPT datasets | Helps improve structured, coherent responses |
| **Logical & Common Sense Reasoning** | ConceptNet, SocialIQa, Metamath Proofs | Ensures model follows logical reasoning & avoids contradictions |

🔹 **Filtering Strategy:**

* **Skip overly trivialized answers** (e.g., avoid dumbed-down content).
* **Balance simple & technical explanations** by weighting different sources in training.

**Phase 3: Humor & Geek Culture Integration**

🔹 **Goal:** Make responses **witty, engaging, and relatable** to STEM users.  
🔹 **Challenges:**

* Most humor datasets focus on **general jokes**, not **STEM-specific jokes**.
* Some humor sources are **too crude** (need moderation).

**✅ Humor Datasets & Sources:**

| **Category** | **Sources** | **Notes** |
| --- | --- | --- |
| **STEM-Oriented Humor** | ✅ XKCD dataset ✅ StackExchange humor threads ✅ MIT OpenCourseWare jokes | Science/math/programming humor |
| **Geeky Pop Culture References** | ✅ TV show transcripts (Big Bang Theory, Futurama) ✅ Classic comedy books (Monty Python, George Carlin) | Ensures nerdy personality quirks |
| **Reddit Humor (Filtered)** | OrionW humor dataset, r/ProgrammerHumor, r/PhysicsMemes | Must apply NLP-based **joke filtering** to avoid low-quality humor |

🔹 **Filtering Strategy:**

* **Use NLP-based joke structure analysis** (detect setup–punchline formats).
* **Keyword filtering:** Include terms like “quantum,” “integral,” “algorithm,” etc.
* **Profanity moderation:** Allow in context (not excessive swearing).