**PROBLEM STATEMENT**

**TABLES DOCUMENT - 2**

**Problem 1: Similarity Detection Gaps**

|  |  |
| --- | --- |
| **The Problem** | Weak detection of phonetic and spelling variants among submitted titles. |
| **Context** | Current title checks often fail to capture variations in spelling, pronunciation, or semantics (e.g., “Namaskar” vs. “Namascar”, “Daily News” vs. “Dainik Samachar”), risking duplication. |
| **Root Cause** | Dependence on basic string-matching algorithms without leveraging phonetic encoding or multilingual semantic understanding causes these misses. |
| **Ideal Solution** | Integrate AI-based semantic models like SBERT or LaBSE, and language detection APIs, to conduct deep multilingual similarity checks across all submissions. |
| **Proposed Solution** | Apply algorithms like Soundex, Metaphone, Levenshtein distance, and use simple token normalization to detect near-matches within and across languages. |

**Problem 2: Inconsistent Rule Enforcement**

|  |  |
| --- | --- |
| **The Problem** | Rule checks like banned terms and format guidelines aren’t always enforced. |
| **Context** | Title applicants sometimes bypass rules on prefixes, suffixes, combinations, or banned words due to manual or inconsistent checks. |
| **Root Cause** | No centralized, automated rule engine; validations are embedded in code and scattered across different points. |
| **Ideal Solution** | Design a dynamic rule engine capable of parsing, updating, and applying contextual rules and exceptions with natural language support. |
| **Proposed Solution** | Define rule sets in external config or tables and enforce them through structured condition checks at submission using modular validation functions. |

**Problem 3: Scalability and Speed Limits**

|  |  |
| --- | --- |
| **The Problem** | System lags or fails under high title load conditions. |
| **Context** | As the title database grows beyond 160,000 entries, slow comparisons delay response time during peak periods. |
| **Root Cause** | Title comparisons are sequential, not indexed or parallelized; caching and queuing strategies are not used. |
| **Ideal Solution** | Use scalable infrastructure with horizontal scaling, load balancers, vector search tools, and Elasticsearch-based comparisons. |
| **Proposed Solution** | Index the title fields, apply batch comparison logic, introduce caching for top rejections, and explore multiprocessing or simple queuing logic. |

**Problem 4: Cross-Language Title Issues**

|  |  |
| --- | --- |
| **The Problem** | Semantically identical titles in different languages can bypass detection. |
| **Context** | Users can submit translated duplicates (e.g., “Daily Evening” vs. “Pratidin Sandhya”), which go undetected due to lack of language mapping. |
| **Root Cause** | No linguistic normalization, synonym matching, or translation stage included in title checks. |
| **Ideal Solution** | Integrate a multilingual semantic matching model using embedding-based NLP with a centralized multilingual dictionary. |
| **Proposed Solution** | Use a static synonym/translation map for commonly repeated terms and expand incrementally; optionally use basic translation APIs for frequent checks. |

**Problem 5: Poor Feedback to Users**

|  |  |
| --- | --- |
| **The Problem** | Users don’t get helpful rejection explanations or tips for rework. |
| **Context** | Many users are unaware of why their titles fail; vague errors lead to confusion and repeat rejections. |
| **Root Cause** | The system lacks integrated logic to map rule violations to meaningful human-readable messages. |
| **Ideal Solution** | Implement an AI assistant that identifies exact issues and provides real-time, interactive suggestions for improvements. |
| **Proposed Solution** | Create a structured feedback system that maps validation failures to feedback categories; highlight problems and offer resubmission help. |

**Problem 6: No Real-Time Acceptance Feedback**

|  |  |
| --- | --- |
| **The Problem** | Users lack insight into the success probability of their title submission. |
| **Context** | With no similarity score or prediction mechanism, users resubmit blindly without knowing how close they are to success. |
| **Root Cause** | No prediction logic or feedback scoring model is integrated into the validation flow. |
| **Ideal Solution** | Train a machine learning model to predict acceptance likelihood using past data and scoring weights on rules/similarity. |
| **Proposed Solution** | Derive a heuristic formula (e.g., probability = 100% - similarity %) along with a consideration of other parameters and display feedback to the user instantly. |

**Problem 7: Repeated Submissions of Pending Titles**

|  |  |
| --- | --- |
| **The Problem** | Identical or similar titles can be submitted by different users if a title is under review. |
| **Context** | Pending applications aren't treated as reserved, so duplicates continue to enter the system until one gets approved. |
| **Root Cause** | No soft-lock mechanism for reserving titles under review; database comparisons exclude active submissions. |
| **Ideal Solution** | Maintain a unified buffer of pending titles and apply the same similarity checks on them as for approved entries. |
| **Proposed Solution** | Introduce a pending title table with soft-lock status and timestamps, and match new submissions against both approved and pending titles. |

**END**