BRAINSTORMING DOCUMENT

CORE PROBLEMS AND SOLUTIONS

PROBLEMS:

❌ Problems That Remain Unsolved or Partially Solved

---

1. Inadequate Similarity Detection Mechanisms

- ✅ **Digitization introduced basic checks**, but:
- ❌ **Advanced phonetic, spelling-variant, or semantic similarity detection** (e.g., using NLP, Soundex, Metaphone) is still **not robust**.
- ❌ **Multilingual equivalence** (e.g., "Daily News" vs. "Dainik Samachar") is **largely ignored**.

---

2. Absence of Automated Guideline Enforcement

- ✅ Guidelines exist (e.g., disallowed words).
- ❌ **Automation is weak or rule-checking is not consistent** — especially for:
  - Prefix/suffix enforcement.
  - Title combination prevention.
  - Periodicity-based variations (e.g., "Weekly XYZ").

---

3. Limited Scalability and Performance

- ❌ While digital portals are in place, they still **struggle under load**, especially during peak periods.
- ❌ The **backend search and comparison algorithms are not optimized** for massive scale (160,000+ titles).

---

4. Challenges in Handling Multilingual Titles

- ❌ No advanced support for **detecting semantic equivalence across languages**.

- ❌ Still possible to submit a title in one language that duplicates the meaning of another existing title in a different language.

---

5. Inadequate User Support and Guidance

- ❌ Many users **do not receive actionable reasons** for rejection.
- ❌ Lack of **interactive suggestions or examples** for improving a title before resubmission.

---

6. Lack of Real-Time Feedback and Probability Scoring

- ❌ The system does **not provide a similarity score** or probability estimate.
- ❌ Users cannot **predict** whether their submission is likely to succeed or fail before submitting.

---

7. Lack of Duplicate Application Tracking

- ❌ Titles under review can **still be submitted again** by another applicant unless already reserved.
- ❌ No robust tracking of **pending applications** for similarity conflict resolution.

✅ Problems That Are Mostly Solved

✅ Manual and Time-Consuming Process

- Replaced with **online PRP portal** and faster digital workflows.

✅ Lack of Transparency

- The system now offers **status tracking** and **application dashboards**.

✅ Delayed Notifications

- Email and SMS alerts are now part of the PRP process.

✅ Integration with Other Authorities

- There’s **some coordination**, especially with the Ministry of I&B, but still not deeply integrated.

SOLUTIONS:

✅ 1. Inadequate Similarity Detection Mechanisms

Objective: Detect visually, phonetically, or semantically similar titles to prevent duplication.

Technical Approach:

- **Phonetic Similarity:**
  - Use algorithms like **Soundex**, **Metaphone**, or **Double Metaphone** to convert words into phonetic codes.
  - Store these codes alongside titles in the database for fast comparison.
  - Compare input title’s phonetic codes with existing ones.
- **Spelling Variation Handling:**
  - Use string similarity algorithms like **Levenshtein Distance**, **Jaro-Winkler**, or **Cosine Similarity** on token sets.
  - Compute a **similarity score (0–100%)** and reject if above threshold.
- **Semantic Similarity:**
  - Use **pretrained language models (e.g., BERT, SBERT)** to convert titles into vector embeddings.
  - Measure **cosine similarity** between vectors of new and existing titles.
- **Title Token Normalization:**
  - Preprocess titles by:
    - Lowercasing
    - Removing stop words
    - Stemming/Lemmatization
- **Store & Index:**
  - Index phonetic codes and embeddings for fast lookup.
  - Use **vector databases** (e.g., FAISS, Pinecone) for semantic search if needed.

✅ 2. Absence of Automated Guideline Enforcement

Objective: Enforce title naming rules consistently and automatically.

Technical Approach:

- **Disallowed Words/Prefixes/Suffixes:**
    - Maintain **configurable rule sets** (in JSON, YAML, or database).
    - On each submission, run checks against:
        - Disallowed keywords
        - Disallowed affixes
        - Sensitive or prohibited terms
- **Combination Detection:**
    - Break all existing titles into token sets.
    - Prevent concatenations of existing tokens from forming a new title.
- **Periodicity Enforcement:**
    - Create rules like:
        - If base title exists: block `"Daily <base>"`, `"Weekly <base>"`, etc.
    - Use **keyword detection** for `"daily"`, `"monthly"`, etc.
- **Rule Engine:**
    - Design a central validation engine with modular rules:
    - `if contains_disallowed_word(title): reject("Disallowed word found")`
    - `if violates_combination_rule(title): reject("Combination of existing titles")`

✅ 3. Limited Scalability and Performance

Objective: Ensure the system performs well under heavy load and large data volumes.

Technical Approach:

- **Efficient Search:**
  - Use **full-text search** capabilities or custom indexing structures (e.g., tries, inverted index).
  - Add indexes on fields like normalized title, phonetic codes.
- **Asynchronous Processing:**
  - Offload expensive similarity comparisons to a background job queue (e.g., Celery, BullMQ, Sidekiq).
  - Use task prioritization to handle peak loads.
- **Pagination and Caching:**
  - Paginate all results.
  - Cache results of high-frequency or recent searches using Redis/Memcached.
- **Distributed Architecture:**
  - Scale the app horizontally (e.g., load-balanced API servers).
  - Use batch processing for periodic checks (e.g., to clean or archive old entries).

🟡 4. Challenges in Handling Multilingual Titles

Objective: Detect duplicate meanings across different languages.

Technical Approach:

- **Translation API or Engine:**
  - Translate all non-English titles to a **common base language** using:
    - Google Translate API
    - DeepL API
    - In-house translation dictionary
- **Cross-Language Embeddings:**
  - Use multilingual models (e.g., **LaBSE**, **mBERT**, **XLM-R**) to encode title semantics.
  - Compare using cosine similarity.
- **Synonym/Thesaurus Mapping:**
  - Build a multilingual synonym dictionary (especially for common media-related terms).
  - Normalize or flag titles containing equivalent terms in different languages.

✅ 5. Inadequate User Support and Guidance

Objective: Help users understand why their title is rejected and how to improve it.

Technical Approach:

- **Explainable Feedback System:**
  - Return structured feedback:
    - “Too similar to: ‘X’”
    - “Contains disallowed word: ‘Y’”
    - “Title pattern violates: combination rule”
- **Interactive Suggestions:**
  - Suggest:
    - Synonym replacements
    - Alternate prefixes/suffixes
    - Minor variations that lower similarity
- **UI Support:**
  - Highlight issues in user input in real-time.
  - Allow one-click title revision or clearing flagged terms.
- **Title History & Resubmission:**
  - Let users view past attempts.
  - Offer smart "revise & resubmit" tools.

✅ 6. Lack of Real-Time Feedback and Probability Scoring

Objective: Inform users early on about likelihood of approval.

Technical Approach:

- **Probability Scoring Model:**
  - Use a formula:
  - `score = 100 - max(similarity_percentage, rule_violation_weight)`
  - Weigh based on:
    - Title closeness
    - Number/type of rules violated
- **Score Interpretation Layer:**
  - Interpret score:
    - 90–100 → Likely approved
    - 60–89 → Risky, show advice
    - 0–59 → Likely rejected
- **Feedback API or Component:**
  - Let users query similarity/score before final submission.

✅ 7. Lack of Duplicate Application Tracking

Objective: Avoid duplicate title submissions while a title is under review.

Technical Approach:

- **Submission Locking:**
    - When a user submits a title:
        - Create a temporary "lock" entry.
        - Prevent submission of titles with similarity > X% until approval decision is made.
- **Application Timeline Tracker:**
    - Track:
        - Submission date
        - Status (pending, approved, rejected)
        - Expiration date of lock (e.g., 7 days)
- **Conflict Resolution Rules:**
    - If two users submit similar titles:
        - First-come-first-reviewed.
        - Notify later submitters to try a different variant.
- **System Feedback:**
    - Alert: “This title is currently under review. Please try again later or choose a different title.”

✅ Recap Table (Generalized)

Problem	Solution Summary	Tools/Concepts
Similarity Detection	Phonetic + semantic checks	Soundex, SBERT, tokenization
Guideline Enforcement	Configurable rule engine	Regex, DB rules, custom logic
Scalability	Indexing, async processing	Job queues, caching, indexing
Multilingual Handling	Translation + semantic mapping	mBERT, LaBSE, Translate APIs
User Support	Clear rejection reasons + suggestions	UI hints, structured responses
Real-Time Scoring	Predictive feedback engine	Scoring model, AJAX/API
Duplicate Tracking	Locking system + timeline	Timestamping, conflict detection

---

END