JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AUTOMATED TITLE VERIFICATION SYSTEM USING SIMILARITY DETECTION

[1]Riya Ameta, [2]Rinku Kunwar Rao, [3]Priyansh Saxena

[1]Student, [2]Student, [3]Student [1]Department of Computer Science and Engineering ,
[1]Geetanjali Institute of Technical Studies, Udaipur,Rajasthan

Abstract : This research paper presents an automated system for validating new title submissions using advanced similarity detection techniques. The system integrates phonetic, string-based, and semantic similarity algorithms to evaluate new titles against a database of existing ones. Additionally, it enforces strict guidelines to reject titles with disallowed prefixes, suffixes, or words, and prevents the creation of titles by combining existing ones or adding periodicity terms. Implemented using Python and Django, the system provides real-time feedback to users, ensuring faster and more accurate title validation. The proposed solution is scalable, efficient, and suitable for industries like publishing, media, and academia

IndexTerms **- Title verification, similarity detection, phonetic matching, fuzzy logic, TF-IDF, Django, multilingual translation.**

I. INTRODUCTION

Effective title management is critical in industries such as publishing, media, and academia, where unique and contextually relevant titles are essential. Manual title verification methods are often inefficient, errorprone, and unable to handle large datasets. This paper introduces an automated title verification system that uses advanced similarity detection techniques to streamline the process. The system leverages multiple algorithms, including fuzzy matching, phonetic similarity, and semantic similarity, to detect duplicate or similar titles. It also enforces specific guidelines to reject titles containing disallowed words, prefixes, or suffixes. By providing real-time feedback to users, the system ensures operational efficiency and reduces the risk of duplication or copyright conflicts.

1.1 MOTIVATION

With the exponential growth of content submissions on digital platforms, maintaining distinct and unique titles has become increasingly challenging. Manual validation methods suffer from human error, inconsistency in detection, inability to process large datasets quickly, and difficulty in identifying semantic and phonetic resemblances. These challenges result in title duplication, degraded user experience, and potential copyright conflicts. A reliable, fast, and intelligent system is required to detect similar titles automatically and provide users with real-time feedback during the submission process.

1.2 KEY FEATURES

The automated title verification system incorporates several key features to ensure efficient and accurate title validation: Comprehensive similarity detection using fuzzy matching, TF-IDF, and phonetic algorithms

- Guideline enforcement for disallowed words, prefixes, and suffixes
- Real-time feedback system with rejection reasons and similarity percentages
- Multilingual support for detecting similar meanings across languages
- Scalable architecture using Django web framework
- Database integration for storing existing titles and pending applications.

1.3 RESEARCH OBJECTIVES

The main goal of the research is to develop an automated system for validating new title submissions. The specific objectives include:

I. Enhancing Title Verification Accuracy:
a. Detect character-based similarities (e.g., minor spelling differences).
b. Identify phonetic and semantic closeness (e.g., "Journey to the Stars" vs. "Starry Journey").

II. Enforcing Submission Guidelines:
a. Reject titles with disallowed prefixes, suffixes, or words.
b. Prevent the creation of titles by combining existing ones or adding periodicity terms.

III. Improving User Experience:
a. Provide real-time feedback to users during the submission process.
b. Display similarity percentages and a probability score for title acceptance.

IV. Developing a Scalable Solution:
a. Create a modular codebase using Django for scalability.
b. Implement efficient algorithms for handling large datasets.

II. LITERATURE SURVEY

The development of automated systems for text similarity detection and verification has gained significant attention in recent years. Various studies highlight the effectiveness of algorithmic approaches, such as fuzzy matching, phonetic encoding, and semantic analysis, in identifying similar content across diverse domains.

2.1 *SURVEY OF EXISTING SYSTEM*

Several systems exist for content verification and similarity detection, but they often have significant limitations when it comes to title validation and comprehensive coverage across different domains. Content management systems typically offer basic duplicate detection but lack advanced phonetic and semantic analysis capabilities. The major drawbacks of existing systems include limited similarity metrics, minimal guideline enforcement, and a lack of real-time feedback mechanisms. This research aims to address these gaps by developing an integrated system that combines multiple similarity detection algorithms and enforces specific guidelines for title verification.

2.2 *IDENTIFIED RESEARCH GAPS*

Despite the presence of various content management systems, significant gaps remain in effectively verifying and validating titles. Most existing systems rely on simple string matching algorithms without incorporating phonetic and semantic analysis, which are essential for comprehensive similarity detection. Additionally, many platforms offer fragmented verification, focusing only on exact matches rather than providing insights into different types of similarities. User feedback is also limited, as current systems do not offer detailed explanations for rejection or provide similarity percentages to guide users in modifying their submissions.

III. PROPOSED SYSTEM

3.1 *PROBLEM STATEMENT AND OBJECTIVES*

With the exponential growth of content submissions on digital platforms, maintaining distinct and unique titles has become increasingly challenging. Manual validation methods suffer from human error, inconsistency in detection, inability to process
large datasets quickly, and difficulty in identifying semantic and phonetic resemblances. This project aims to develop an intelligent, web-based system that helps users identify whether a submitted title is too similar to an existing one. The primary objective of this research is to develop a comprehensive and interactive system that automatically verifies and validates title submissions. The platform aims to provide a centralized solution covering various aspects of similarity detection, including character-based, phonetic, and semantic analysis. Ensuring accuracy and efficiency, the system will support a responsive design and multilingual content to reach diverse industries.

3.2 *SCOPE OF THE WORK*

This project is particularly useful for online publishing platforms (e.g., books, blogs), media production houses (e.g., movies, TV shows), academic journals and conference submission systems, and product and brand naming

services.

The system aims to:

- Detect character-based similarities using fuzzy matching algorithms
- Identify phonetic resemblances through the Double Metaphone algorithm
- Analyze semantic closeness using TF-IDF and cosine similarity
- Enforce guidelines for disallowed prefixes, suffixes, and words
- Provide actionable feedback to users with similarity percentages

IV. SYSTEM FRAMEWORK AND ARCHITECTURE

The system follows a 3-layer architecture designed to provide an efficient, accurate, and user-friendly title verification experience:

4.1 *PRESENTATION LAYER*

Users submit a title using a web form designed with HTML and CSS The form sends the title to the server for processing via Django views

Results are displayed using Django templates with clear feedback and similarity percentages

4.2 *BUSINESS LOGIC LAYER*

1. **Title Processing:**
   - Users submit a title through the web interface
   - The system preprocesses the title by removing extra spaces and converting to lowercase
   - Existing titles are fetched from the database for comparison
2. **Similarity Detection**:

Fuzzy Matching: Uses fuzz.token_sort_ratio() to determine character-level closeness

TF-IDF Matching: Converts titles into vector space models and compares them using cosine similarity Phonetic Matching: Uses the Double Metaphone algorithm to detect similar-sounding titles

3. **Guideline Enforcement:**

Checks for disallowed words, prefixes, and suffixes

Prevents creation of titles by combining existing ones

Rejects titles with periodicity terms added to existing titles

4. **Decision Logic:**

If similarity checks fail, the title is saved as unique

If matches are found, the system calculates a probability score and provides rejection reasons

4.3 *DATA LAYER*

- Django ORM for database operations
- SQLite database (scalable to PostgreSQL)
- Storage of existing titles and pending applications

V.IMPLEMENTATION DETAILS

5.1 *TECHNOLOGY STACK*

Component	Technology Used
Backend	Python (Django)
Algorithms	Fuzzy Matching, TF-IDF, Cosine Similarity, Double Metaphone
Frontend	Django Templates (HTML, CSS)
Database	SQLite
Libraries	scikit-learn, fuzzywuzzy, jellyfish, translate
C C	

5.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

- Processor: Intel Core i5 or higher
- Storage: SSD with 256GB or more for faster read/write operations
- RAM: 8GB or more

Software Requirements:

- OperatingSystem:Windows/Linx
- Framework: Django (Web Framework)
- Programming Language: Python 3.x

Libraries:

- fuzzywuzzy for fuzzy string matching
- scikit-learn for TF-IDF vectorization and cosine similarity
- translate for multilingual checks
- jellyfish for phonetic similarity

VI. EVALUATION AND RESULTS

6.1 *TESTING METHODOLOGY*

The system was tested using a dataset of 1,000 titles across domains like books, movies, and blogs to evaluate its effectiveness in detecting similar titles and enforcing guidelines.

6.2 *PERFORMANCE METRICS*

- Detection Accuracy: > 95% for similar strings
- Semantic Match Reliability: High precision for meaningful phrases
- User Feedback: Positive – clarity, speed, and usability
- Processing Time: < 2 seconds for verification against 10,000 existing titles

6.3 *SAMPLE CASE STUDY*

Input Title: "The Last Voyage"

Detected Matches:"Last Voyage" (Fuzzy Match: 90%)

- "Voyage to the End" (Semantic Match: 75%) **System Feedback:** "Similar titles found. Please consider changing the name." **Probability Score:** 35% chance of acceptance

VII. CONCLUSION AND FUTURE WORK

The automated title verification system effectively replaces manual validation with real-time feedback using string, phonetic, and semantic similarity algorithms. It ensures faster, more accurate title checking and can be adopted across various domains for better content management.

Limitations:

- The system currently supports English and a limited set of Indian languages
- Creative metaphorical titles may not be effectively detected
- Advanced deep learning models (e.g., BERT) are not yet integrated

Future Enhancements:

- Add support for more languages
- Integrate BERT-based NLP models for deeper semantic understanding
- Build REST APIs for cross-platform integration
- Add support for bulk title uploads and processing
- Implement machine learning to improve similarity detection over time

VIII. REFERENCES

- Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing. Pearson.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
- Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
- Scikit-learn Developers. (2023). scikit-learn Documentation.
- Django Software Foundation. (2023). Django Documentation