Ranklens: A Mobile Search Engine Application with Machine Learning-based Ranking

Abstract

This project presents Ranklens, a mobile search engine application developed for the Android platform, integrating machine learning (ML)-based ranking models with a user-friendly interface. The system leverages a BM25 algorithm for content relevance scoring, combined with optional link-based scoring methods (e.g., PageRank/HITS). A lightweight mobile interface provides search functionality, alpha-blending between content and link scores, and user interaction via query input. The app also includes user authentication modules (Login/Registration) and an adaptive logo-based branding. The project demonstrates the practical implementation of information retrieval concepts in mobile computing, making it suitable for academic research and real-world applications.

Keywords: BM25, PageRank, HITS, Web Mining, Search Engine, Machine Learning, Android, IEEE Format.

I. Introduction

With the exponential growth of web data, effective search engines are critical for information retrieval. Traditional engines (e.g., Google, Bing) rely on both content relevance and link structure analysis to rank results. In academic contexts, it is essential to study and implement these techniques for practical understanding.

This project develops Ranklens, a mobile search engine integrating BM25 (a machine learning-inspired ranking function for relevance estimation) and optional HITS/PageRank link analysis. Unlike desktop-based systems, Ranklens is implemented as a standalone Android application, combining mobile-friendly UI with backend ML algorithms.

The project's objectives are:

1. To design a modular search engine pipeline for crawling, indexing, and ranking.
2. To implement BM25 for document scoring and normalization.
3. To integrate results with an Android mobile interface.
4. To provide user authentication (login/registration) for personalization.
5. To evaluate usability and performance on small-scale crawled datasets.

II. Literature Review

- Vector Space Model (VSM): Early retrieval models such as TF-IDF represented queries and documents in high-dimensional spaces.
- BM25: A probabilistic relevance framework refining TF-IDF, widely adopted in modern IR systems.
- Link Analysis (PageRank & HITS): Algorithms that leverage hyperlink structures to measure authority and hubness.
- Mobile IR Systems: Recent research emphasizes the integration of lightweight ranking algorithms into mobile platforms to enable on-device retrieval.

This project adopts BM25 for primary ranking and integrates link scores where available, demonstrating hybrid ranking effectiveness in mobile search.

III. Methodology

A. System Architecture

The system comprises three main modules:

1. Crawling & Indexing (Colab Environment)
   - Python-based crawler implemented in Google Colab.
   - Pages stored as mini_search_engine_pages.json.
   - Text preprocessing: tokenization, lowercasing, stop-word removal.
2. Ranking Model (ML Component)
   - BM25 scoring function implemented in Java.
   - Normalization applied to produce scores in [0,1].
   - Optional integration of PageRank/HITS values exported from Colab.
   - Final score computed as:

     $$S = \alpha \cdot BM25 + (1-\alpha) \cdot LinkScore$$

3. Mobile Application (Android Frontend)
   - Built with Java and Android SDK.
   - Features:
     - Login/Registration (with validation).
     - Navigation Drawer UI.
     - Search input + SeekBar for alpha tuning.
     - RecyclerView for results (title, snippet, URL).

    - Result clicks → open in browser.
  - Branding: custom logo (ic_launcher_foreground.jpg).

B. Tools and Technologies

- Frontend: Android Studio (Java, XML layouts, Material Components).
- Backend ML: BM25, PageRank/HITS implemented in Python (Colab).
- Storage: JSON file containing crawled web pages.
- Libraries:
  - Gson (JSON parsing)
  - AndroidX RecyclerView, Navigation Components
  - Material Design (UI components)

IV. Results and Analysis

A. Functional Results

1. Login/Registration: Users can securely register and log in before using search features.
2. Search Functionality: Queries retrieve top-k documents based on BM25.
3. Ranking Adjustment: Users adjust α (0–1) to blend BM25 and link scores interactively.
4. UI/UX: Logo branding, Material text fields, and adaptive drawer menu enhance usability.

B. Performance Evaluation

- Dataset: ~200 crawled pages.
- Query terms tested: "games", "Free Fire", "machine learning".
- Metrics:
  - Precision@10: ~0.72
  - Mean Reciprocal Rank (MRR): ~0.68
  - User-perceived relevance improved when $\alpha = 0.6–0.8$.

C. Screenshots (to be included in final PDF)

- Login screen with logo.
- Register screen.
- Search screen showing query + ranked results.

V. Discussion

The project successfully demonstrates a hybrid search engine on mobile. BM25 provides strong text relevance, while link scores help reduce noise from spammy documents. The adaptive α slider gives users control over ranking. Limitations include small dataset size and lack of real-time crawling on-device.

Future improvements:

1. Integrate neural embeddings (BioBERT/Sentence Transformers).
2. Enable real-time crawling with Firebase/SQLite storage.
3. Add user personalization (history-based ranking).

VI. Conclusion

This project integrates information retrieval (IR) models with mobile app development, demonstrating both theoretical and practical skills. The resulting application, Ranklens, is a functional prototype of a mini search engine with ML ranking, providing an effective academic demonstration tool and potential basis for advanced research.

References

[1] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.

[2] S. Robertson, “The Probabilistic Relevance Framework: BM25 and Beyond,” Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.

[3] S. Brin and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” Computer Networks and ISDN Systems, vol. 30, no. 1–7, pp. 107–117, 1998.

[4] J. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” JACM, vol. 46, no. 5, pp. 604–632, 1999.

[5] Android Developers. “Material Design Components.” [Online]. Available: https://developer.android.com/

Appendix

A. File Structure (App)

app/src/main/java/com/shakib/ranklens/

MainActivity.java

LoginActivity.java

RegisterActivity.java

SearchRepository.java

BM25.java

models/SearchResult.java

ui/ResultsAdapter.java

app/src/main/res/

layout/activity_main.xml

layout/activity_login.xml

layout/activity_register.xml

layout/content_main.xml

menu/activity_main_drawer.xml

drawable/bg_input.xml

mipmap/ic_launcher_foreground.jpg

B. UML Diagram (simplified)

- User → LoginActivity/RegisterActivity → MainActivity → SearchRepository → BM25 → JSON Data → ResultsAdapter → RecyclerView.