

WRAP: An AI-Enhanced Visual Task Management Application with On-Device Automatic Tag Generation

Kaushiik, Suhail, Kavin Mohan Kumar, Nikitha Magesh

¹B.E. Computer Science and Engineering, Kumaraguru College of Technology, Tamil Nadu, India

Senthil Kumar V

Assistant Professor, Department of Computer Science and Engineering, Kumaraguru College of Technology, Tamil Nadu, India

Abstract - Traditional task management applications primarily rely on text-based task entries and manual categorization, which often leads to inconsistent organization and limited search efficiency. This paper presents WRAP, an AI-enhanced visual task management application that integrates document attachment support and intelligent on-device automatic tagging. Unlike conventional to-do list applications, WRAP enables users to create tasks with images and documents while automatically generating relevant tags using Natural Language Processing techniques. The system employs Google ML Kit Entity Extraction combined with heuristic keyword analysis to identify meaningful contextual tags directly on the device without requiring cloud-based services. The application follows a layered architecture consisting of presentation, application, data, and file storage layers. Generated tags enhance task discoverability, improve search precision, and reduce manual categorization effort. Experimental evaluation demonstrates improved contextual organization and retrieval efficiency compared to manual tagging approaches. WRAP highlights the practical integration of on-device Artificial Intelligence techniques in modern productivity applications.

Key Words: WRAP, Task Management System, Natural Language Processing, ML Kit, Automatic Tagging, Android Application, Semantic Analysis.

1. INTRODUCTION

Task management applications play a crucial role in organizing academic, professional, and personal activities. Most traditional systems rely on manual categorization and text-based entries, which often result in inconsistent tagging and inefficient search capabilities. As modern workflows increasingly involve visual references and document-based information, there is a need for intelligent task management systems that support contextual organization.

WRAP is designed as a visual task management application that integrates image attachments, document support, and intelligent automatic tagging. The system enhances traditional task organization by incorporating on-device Natural Language Processing techniques. Instead of requiring users to manually assign tags, WRAP analyzes task descriptions and generates relevant tags automatically.

The integration of automated tagging improves task discoverability, enhances contextual recall, and reduces user effort. This paper presents the architecture, implementation,

and evaluation of WRAP as a lightweight, efficient, and offline-capable productivity solution.

1.1 Problem Statement

Conventional task management applications suffer from several limitations. Manual tagging requires additional user effort and often results in inconsistent categorization. Many systems do not support contextual grouping of tasks based on semantic meaning. Furthermore, traditional keyword-based search relies on exact string matching, which reduces retrieval accuracy.

There is a need for a structured task management system that integrates visual references and intelligent tag generation while maintaining efficient on-device processing without dependency on external cloud services.

1.2 Objectives of the Study

The objectives of this study are to design and implement a visual task management application with automatic tag generation using on-device Natural Language Processing techniques. The system aims to improve searchability, reduce manual categorization effort, enhance contextual organization, and ensure efficient offline functionality.

2. SYSTEM ARCHITECTURE

WRAP follows a four-layer architecture consisting of the Presentation Layer, Application Layer, Data Layer, and File Storage Layer. This modular design ensures separation of concerns, maintainability, and scalability.

The Presentation Layer represents the Android user interface, including task creation screens, a gallery-style

dashboard, search functionality, and task detail views. The ViewModel component manages state and mediates

communication between the user interface and application logic.

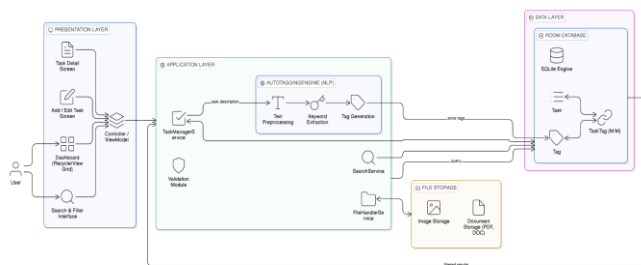
The Application Layer contains core services including TaskManagerService, FileHandlerService, SearchService, and the AutoTaggingEngine. When a user creates a task, the task

description is passed to the AutoTaggingEngine for semantic processing.

The Data Layer is implemented using the Room Persistence Library built on SQLite. It stores Task entities, Tag entities, and their many-to-many relationships to enable efficient querying and filtering.

The File Storage Layer manages image and document storage locally within the device. File paths or URIs are linked to task records without duplicating file data in the database.

Fig. 1: Layered System Architecture of WRAP Application



3. IMPLEMENTATION

WRAP was developed using Java in Android Studio. The application integrates UI components, local database storage, file management, and an on-device auto-tagging module.

3.1 Auto-Tagging Engine Design

The Auto-Tagging Engine is implemented using Google ML Kit Entity Extraction combined with heuristic keyword processing. The tagging mechanism operates entirely on-device, ensuring offline functionality and low latency.

When a user enters a task description, the system invokes the ML Kit EntityExtractor to identify structured entities such as dates and contextual references. Extracted entities are added to a tag collection.

The text is then cleaned by removing special characters and converting words to lowercase. A predefined stopword list filters out common non-informative words. Words exceeding a specified length threshold are considered valid keywords.

To improve contextual representation, a bigram generation mechanism combines adjacent valid words into compound tags. All tags are stored in a LinkedHashSet to eliminate duplicates while maintaining insertion order. The final tag list is limited to a maximum predefined count to ensure concise categorization.

This hybrid approach provides lightweight semantic tagging without requiring cloud-based AI services.

3.2 Database Design and Data Modeling

The Room database manages task storage through normalized relational modeling. The Task entity stores task details, completion status, timestamps, and file references. The Tag entity stores unique generated tags.

A many-to-many relationship is implemented through a junction table to associate tasks with multiple tags efficiently. This structure improves query performance and reduces redundancy during search operations.

3.3 File Handling and Storage Mechanism

WRAP supports attachment of images and documents including PDF and DOC files. The FileHandlerService manages file selection, storage, and retrieval. Files are stored locally on the device, and their URIs are linked to corresponding tasks in the database.

This approach ensures efficient file access while maintaining structured metadata storage.

3.4 Search and Filtering Mechanism

The SearchService enables intelligent retrieval of tasks by matching search queries against both task descriptions and generated tags. Because tags are derived using semantic filtering rather than manual assignment, search results demonstrate improved contextual relevance compared to exact keyword matching systems.

4. PERFORMANCE EVALUATION AND ANALYSIS

The WRAP application was evaluated using diverse task scenarios including academic projects, professional workflows, and personal planning tasks. The evaluation focused on tag relevance, search accuracy, and processing efficiency.

For example, when the input task was “Prepare final presentation for software engineering review,” the system generated tags such as “presentation,” “software engineering,”

and “review.” These tags accurately represented the task context without manual input.

Compared to manual tagging, the automated tagging mechanism reduced categorization effort and improved consistency. Search experiments demonstrated improved retrieval precision because relevant tasks were identified through generated tags rather than relying solely on exact string matching.

The average tag generation time was observed to be negligible under normal execution conditions, demonstrating the efficiency of on-device NLP processing.

5. CONCLUSION

This paper presented WRAP, a visual task management application enhanced with on-device automatic tagging using Natural Language Processing techniques. The system integrates

image and document support with intelligent semantic tagging to improve task discoverability and organization.

The implementation demonstrates that combining ML Kit entity extraction with heuristic keyword processing provides efficient and lightweight automatic tagging. The architecture ensures modularity, offline capability, and scalable design.

WRAP highlights the effective use of on-device AI techniques in productivity systems. Future work may include enhanced contextual ranking, advanced phrase extraction, and adaptive tag prioritization mechanisms.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to the faculty members of the Department of Computer Science and Engineering at Kumaraguru College of Technology for their guidance and support throughout this project.

REFERENCES

- [1] Google Developers, "ML Kit Entity Extraction," Android Developer Documentation, 2023.
- [2] Android Developers, "Guide to App Architecture," 2023.
- [3] SQLite Consortium, "SQLite Documentation," 2023.
- [4] I. H. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2011.