International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

A Review of AI-Assisted Academic Submission Management: Whatsapp-Based Deadline Tracking and Context-Aware Assignment Support

Prof. Sushmita Chaure
Shriraj Kamble, Ayush Jadhav, Aadesh Choudhari, Pratham Salunkhe.

[1]Department of Artificial Intelligence and Machine Learning, ZES Zeal Polytechnic, Pune, Maharashtra, India

Abstract—Academic submission work is distributed across classroom announcements, student group conversations, assignment instructions and individual study tasks. Important information such as due dates, deadline changes and required submission components can be difficult to track when embedded in long chat histories. This paper reviews AI and document-oriented information processing for academic submission workflows and presents an implementation-grounded case study of an AI Submission Assistant. The inspected Streamlit application imports exported WhatsApp conversations in text or ZIP form, parses message metadata, identifies task-related messages, resolves selected explicit and relative date expressions, categorizes messages by subject and stores messages and deadlines in SQLite. It also implements duplicate-message filtering, deadline urgency calculation, TF–IDF retrieval of relevant class messages and an optional Anthropic language-model interface for drafting assignment-oriented responses using retrieved context. The review relates these functions to research on large language models in academia, document-level information extraction, document understanding and hallucination in natural language generation. The analysis identifies limitations including rule-based date interpretation, imperfect event-to-assignment matching, dependence on announcement quality, the need for human verification and the absence of a demonstrated document-file export function in the inspected code. No quantitative accuracy or usability claims are made because a controlled evaluation dataset and measured results were not supplied. The paper concludes with a validation framework and development directions for improving reliability, transparency and usefulness.Academic submission work is distributed across classroom announcements, student group conversations, assignment instructions and individual study tasks. Important information such as due dates, deadline changes and required submission components can be difficult to track when embedded in long chat histories. This paper reviews AI and document-oriented information processing for academic submission workflows and presents an implementation-grounded case study of an AI Submission Assistant. The inspected Streamlit application imports exported WhatsApp conversations in text or ZIP form, parses message metadata, identifies task-related messages, resolves selected explicit and relative date expressions, categorizes messages by subject and stores messages and deadlines in SQLite. It also implements duplicate-message filtering, deadline urgency calculation, TF–IDF retrieval of relevant class messages and an optional Anthropic language-model interface for drafting assignment-oriented responses using retrieved context. The review relates these functions to research on large language models in academia, document-level information extraction, document understanding and hallucination in natural language generation. The analysis identifies limitations including rule-based date interpretation, imperfect event-to-assignment matching, dependence on announcement quality, the need for human verification and the absence of a demonstrated document-file export function in the inspected code. No quantitative accuracy or usability claims are made because a controlled evaluation dataset and measured results were not supplied. The paper concludes with a validation framework and development directions for improving reliability, transparency and usefulness.

Index Terms—Academic workflow, assignment management, WhatsApp chat parsing, deadline extraction, information retrieval, TF–IDF, large language models, AI-assisted content generation.

I. INTRODUCTION

Students commonly receive academic instructions through classroom announcements, learning platforms and informal messaging groups. In a class messaging group, an assignment may be announced, clarified, repeated and later rescheduled. A student who reads only the latest message may miss an earlier instruction, while a student relying on an old announcement may follow an outdated deadline. The problem is therefore not limited to storing dates; it involves extracting task information from conversational messages, retaining context and presenting current status in an actionable form.

Large language models (LLMs) have expanded the possibilities for conversational assistance and academic content drafting. However, their outputs can be inaccurate, biased or misleading, and fluent text should not be treated as verified information [1]. A useful academic assistant must combine language generation with access to relevant source messages and explicit task records. This paper examines that combination through the AI Submission Assistant, a student-oriented application designed around exported WhatsApp class conversations.

The paper has two objectives: (i) review research areas that support academic workflow assistance, including document information extraction and AI-generated content; and (ii) describe the functions, architecture, limitations and evaluation needs of the implemented application. The contribution is an implementation-grounded analysis, not a claim that a new general-purpose AI model or a fully automated document-production engine has been developed.

II. REVIEW SCOPE AND APPROACH

This paper adopts a focused narrative review. The selected literature covers four connected themes: LLM use in academic work, extraction of structured information from documents, deep-learning approaches to document understanding, and hallucination in natural language generation. These themes were selected because the application interprets semi-structured conversational text, identifies actionable information, retrieves context and generates student-facing responses. This is not a systematic

review: no exhaustive database search, formal screening protocol or statistical synthesis was performed.

The implementation analysis is based on the supplied Python source files (app.py, engine.py and db.py), dependency list and sample WhatsApp export. The sample conversation is treated only as an illustrative input. It is not a representative dataset and cannot establish extraction accuracy, generalization across institutions or user satisfaction. The code is examined for implemented behavior, not assumed intended behavior.

III. RELATED WORK

A. Large Language Models in Academic Work

Meyer et al. discuss the opportunities and risks of ChatGPT and other LLMs in academia, including writing assistance, educational support, programming, bias, factual accuracy and inappropriate use [1]. This is directly relevant to an assignment assistant: generated explanations can reduce drafting effort, but students remain responsible for checking facts, following institutional instructions and ensuring that submitted work reflects genuine understanding. An LLM should support a student's work rather than replace source verification or academic responsibility.

B. Document-Level Information Extraction

Zheng, Wang and Huang survey document-level information extraction and identify challenges such as coreference resolution, reasoning limitations, labeling noise and relation transitivity [2]. Although classroom chat exports are not conventional business documents, the underlying problem is related: a system must connect information distributed across multiple messages. An initial announcement may contain the task and original date, while a later message changes the date. Correct interpretation requires more than detecting a date string; it requires identifying which task the later statement modifies.

C. Document Understanding and Key Information Extraction

Rombach and Fettke review deep-learning approaches to key information extraction from business documents and analyze methods published between 2017 and 2024 [3]. Their review highlights the importance of converting unstructured or visually complex documents into structured information. The AI Submission Assistant addresses a narrower text-based case: extracting subject, task description, due date and event type from exported chat messages. Its current implementation uses regular expressions and date parsing rather than deep-learning document-understanding models. This comparison places the project within the wider information-extraction landscape without claiming equivalent technical sophistication.

D. Hallucination and Reliability of Generated Text

Ji et al. survey hallucination in natural language generation and discuss its occurrence, evaluation and mitigation across several generation tasks [4]. This concern is important when an assignment assistant creates academic answers. A response may be well structured yet contain incorrect facts or unsupported details. In the reviewed application, retrieved class messages provide task-specific context to the language model, but retrieval does not guarantee that every generated statement is correct. Human review and independent source checking remain necessary.

IV. SYSTEM OVERVIEW AND IMPLEMENTED DESIGN

The application is implemented in Python using Streamlit for the user interface, SQLite for persistence, pandas for dashboard data handling, scikit-learn for TF–IDF retrieval, python-dateutil for date parsing and the Anthropic Python client for optional LLM access. The interface is organized into four tabs: Dashboard, Assistant, Import WhatsApp chat and Deadlines. The sidebar accepts an optional Anthropic API key and a configurable subject list with aliases.

A. WhatsApp Export Import and Parsing

The import interface accepts .txt and .zip files. For a ZIP upload, the engine reads the first text file found in the archive. The parser handles two common WhatsApp export line patterns, normalizes selected Unicode spacing and direction characters, parses timestamps with day-first date interpretation, captures sender and message text, and appends continuation lines to the current message. Common omitted-media and deleted-message placeholders are excluded. The parser is designed for exported text conversations, not direct access to live WhatsApp messages or media content.

B. Task and Deadline Extraction

The engine uses keyword patterns to identify task-related messages containing terms such as assignment, submit, deadline, record, practical, project or report. Separate patterns identify changes (for example, extended, postponed or rescheduled) and cancellations. Subject classification compares configured subject names and aliases with message text. Date extraction recognizes numeric and month-name formats, as well as relative expressions such as today, tomorrow, day after tomorrow, a number of days and weekday names. Relative dates are resolved from the timestamp of the message, rather than the date on which the file is imported.

Detected events are added to the deadline database, marked as changed or marked as cancelled. The code attempts to skip duplicates by comparing subject and due date, with an additional title-similarity check for general-subject entries. This is a lightweight approach, but event association is limited: a change message is matched to the latest active or changed deadline for the detected subject, rather than through a robust assignment-identity model. Multiple assignments in the same subject can therefore create ambiguity.

C. Persistent Storage and Dashboard

The database contains message and deadline tables. Each imported message receives a SHA-1 hash based on timestamp, sender and text; a uniqueness constraint prevents the same message from being inserted again. Deadline records store subject, title, due date and status. The dashboard displays pending items, overdue items and deadlines approaching within seven days, and sorts live

tasks by an urgency score. Users can also add deadlines manually, mark them as done or delete them.

D. Context Retrieval and AI-Assisted Answers

For a user question, the engine builds a searchable set of stored messages. Messages with fewer than four words are omitted from retrieval. A TF–IDF vectorizer with unigrams and bigrams represents the messages, and cosine similarity ranks candidate matches. Up to six messages above a similarity threshold are returned, with a short excerpt of the following message appended as contextual information. The assistant supplies these retrieved excerpts and the current deadline list to an Anthropic language model when an API key is available. A system instruction asks the model to produce organized assignment answers, follow requirements found in the chat, state assumptions when requirements are absent, and include a pre-submission checklist. The interface displays the answer and exposes the class messages used as retrieval context.

The application therefore implements AI-assisted answer drafting grounded in retrieved class messages. The inspected code does not create or download a formatted DOCX or PDF file. Accordingly, "document generation" here refers to generating organized academic answer content in the application interface; automated file creation and export should be treated as future work unless a separate implementation is provided.

V. FUNCTIONAL COMPARISON AND ANALYSIS

Table I summarizes the relationship between the reviewed research themes and the application. It is a qualitative comparison of scope, not a benchmark of performance.

Area	Research perspective	AI Submission Assistant
Academic LLM use [1]	Writing and learning support with accuracy and integrity risks.	Optional LLM drafts assignment-oriented responses; student verification remains necessary.
Document information extraction [2]	Extracting related entities and events across document context.	Rule-based extraction of subjects, task phrases, dates, changes and cancellations from chat messages.
Document understanding [3]	Structured extraction from complex business documents.	Text-only WhatsApp export parsing; no OCR or visual document processing is implemented.
Hallucination [4]	Generated language can be fluent but factually unsupported.	Retrieved messages provide context but do not guarantee factual correctness.

TABLE I. QUALITATIVE COMPARISON OF RESEARCH THEMES AND IMPLEMENTED FUNCTIONS

VI. BENEFITS, LIMITATIONS AND RESEARCH GAPS

A. Potential Benefits

The integrated workflow reduces the need to manually search long class conversations for every assignment announcement. It provides a central view of upcoming tasks, retains imported messages, identifies selected deadline updates and makes relevant messages visible alongside AI-generated responses. Subject aliases allow adaptation to local course terminology. Manual deadline entry provides a fallback when an announcement cannot be parsed automatically.

B. Technical Limitations

The design depends on regular expressions and a fixed keyword set. Messages phrased unexpectedly, dates expressed ambiguously, missing years, multilingual text or conflicting announcements may not be interpreted correctly. Subject-level event matching can associate a changed date with the wrong assignment when several tasks belong to one subject. The weekday rule selects the next occurrence of a named weekday; it does not interpret every possible use of phrases such as "next Friday." The parser ignores media content and does not inspect images or attached documents.

TF–IDF retrieval is lexical. It may miss semantically related messages that use different vocabulary, and the similarity threshold may exclude relevant context. Appending the next message is a simple context heuristic and does not establish that it is a reply to the selected message. The optional LLM call depends on a valid API key and an available model endpoint. The code does not implement a formal citation system for generated academic answers or automatically validate claims against authoritative course material.

C. Privacy, Integrity and Reliability

WhatsApp exports may contain names, personal messages and other sensitive information. Users should import only conversations they are permitted to process, minimize unnecessary data and protect the local database and API credentials. The application should explain when message content is sent to an external model provider. Generated answers should be treated as drafts: students should verify facts, cite original sources, follow institutional AI-use rules and avoid submitting unreviewed model output as independent work.

D. Research Gaps

The main gaps are reliable assignment-level event resolution, multilingual and informal-language understanding, semantic retrieval, evidence-linked answer generation, explainable confidence indicators, privacy-preserving model use and measurable evaluation. A future version could represent each assignment as a persistent entity with an event history, link each deadline change to its source message and require confirmation when the system is uncertain. It could also add document templates and DOCX/PDF export, but those capabilities are not part of the inspected implementation.

VII. PROPOSED EVALUATION FRAMEWORK

The supplied source code and illustrative chat export are insufficient to report measured system performance. A controlled evaluation should be conducted before making claims about accuracy or efficiency. Recommended measures include:

- Deadline extraction precision, recall and F1-score, using manually annotated messages as ground truth.
- Date interpretation accuracy, measured separately for explicit dates, relative dates and ambiguous expressions.

- Change and cancellation association accuracy, checking whether each event updates the correct assignment.
- Duplicate detection rate, including duplicate messages and repeated announcements with different wording.
- Retrieval precision at k and recall at k, based on student questions paired with relevant source messages.
- Answer factuality and instruction compliance, reviewed by subject experts against retrieved messages and course material.
- Usability and task-completion time, measured with student participants using representative class-chat exports.

The evaluation dataset should contain varied announcement styles, multiple subjects, repeated reminders, corrections, cancellations, relative dates, misspellings and ambiguous references. Personal identifiers should be removed or replaced before analysis. Dataset splits and annotation rules should be documented so that results can be reproduced.

VIII. CONCLUSION

This review examined the research foundations and implementation of an AI Submission Assistant for managing academic tasks communicated through WhatsApp class groups. The inspected application combines chat-export parsing, rule-based task and date extraction, subject categorization, SQLite persistence, duplicate-message filtering, urgency-based deadline presentation, TF–IDF retrieval and optional LLM-based assignment-answer drafting. These functions demonstrate a practical integration of information extraction, lightweight information retrieval and generative AI in a student workflow.

The review also shows why such a system must be treated as an assistive tool rather than an autonomous authority. Rule-based extraction can misinterpret ambiguous messages, retrieval may omit relevant context and language models can generate inaccurate content. The current implementation does not yet provide verified assignment-level event linking, measured performance evidence or automatic document-file export. Future development should prioritize source-linked deadline changes, stronger semantic understanding, privacy safeguards, answer verification and transparent evaluation. With these improvements and a documented test study, the system could provide a more reliable foundation for academic submission management.

ACKNOWLEDGMENT

The authors thank the Department of Artificial Intelligence and Machine Learning, ZES Zeal Polytechnic, Pune, for academic guidance and support. This statement should be revised to reflect actual contributions and permissions before submission.

REFERENCES

[1] J. G. Meyer, R. J. Urbanowicz, P. C. N. Martin, K. O'Connor, R. Li, P.-C. Peng, T. J. Bright, N. Tatonetti, K. J. Won, G. Gonzalez-Hernandez, and J. H. Moore, "ChatGPT and large language models in academia: opportunities and challenges," BioData Mining, vol. 16, art. 20, 2023, doi: 10.1186/s13040-023-00339-9.

[2] H. Zheng, S. Wang, and L. Huang, "A comprehensive survey on document-level information extraction," in Proc. Workshop on the Future of Event Detection (FuturED), 2024, pp. 58–72, doi: 10.18653/v1/2024.futured-1.6.

[3] A. M. Rombach and P. Fettke, "Deep learning based key information extraction from business documents: systematic literature review," ACM Computing Surveys, vol. 58, no. 2, pp. 1–37, 2025, doi: 10.1145/3749369.

[4] Z. Ji et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, art. 248, 2023, doi: 10.1145/3571730.