ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

Automated Dataset Preparation and Management from Raw Data Using Context-Aware Chunking and LLMs

S. Saraswathi[1]; Vignesh M.[2]; Vijayalakshmi S.[3]; Tholkapian M.[4]

[1;2;3;4]Department of Information Technology, Puducherry Technological University, Puducherry, India

Publication Date: 2026/04/20

Abstract: The increasing adoption of Large Language Models (LLMs) has created a critical need for high-quality, structured training data derived from real-world, unstructured sources such as documents and web content. However, existing research primarily focuses on model architectures or isolated data processing stages, lacking a unified solution for end-to-end dataset preparation. This project proposes a reusable, production-ready Dataset Preparation Pipeline that automates the ingestion, cleaning, chunking, task-specific data generation, and quality evaluation of raw data for multiple LLM applications including question answering, summarization, and classification. The system integrates automated quality assessment to filter low-quality or biased samples and exports curated datasets in standard formats for seamless downstream use. By adopting a data-centric approach, the proposed pipeline bridges the gap between raw real-world data and reliable LLM training datasets, enabling scalable, continuous, and efficient dataset generation for modern language model development.

Keywords: *Large Language Models (LLMs), Dataset Preparation Pipeline, Data-Centric AI, Quality Assessment, Training Data Generation.*

How to Cite: S. Saraswathi; Vignesh M.; Vijayalakshmi S.; Tholkapian M. (2026) Automated Dataset Preparation and Management from Raw Data Using Context-Aware Chunking and LLMs. *International Journal of Innovative Science and Research Technology*, 11(4), 1191-1198. https://doi.org/10.38124/ijisrt/26apr952

I. INTRODUCTION

The system is designed as a modular and automated pipeline that handles the complete lifecycle of dataset preparation. It begins with data ingestion, where inputs are collected from document uploads and web sources. This is followed by data preprocessing, which includes cleaning, normalization, and noise removal to ensure consistency.

Next, the pipeline performs text chunking, breaking large documents into smaller, meaningful segments suitable for LLM processing. It then applies task-specific data generation algorithms to create structured training samples tailored for tasks such as question answering, summarization, classification, and instruction tuning. The architecture supports asynchronous processing and parallel execution, enabling scalability and efficient handling of large datasets.

A key component of the system is the automated quality evaluation layer, which ensures that only high-quality data is included in the final dataset. This layer evaluates data based on factors such as relevance, coherence, bias, and consistency.Low-quality, biased, or irrelevant samples are filtered out before dataset finalization. This prevents issues like toxicity, misinformation, and poor model performance.

The proposed system follows a data-centric approach, shifting focus from model development to data quality and lifecycle management. It integrates multiple stages—data ingestion, preprocessing, task-aware generation, quality evaluation, and standardized export—into a unified and reusable framework. By bridging the gap between raw real-world data and structured LLM-ready datasets, the pipeline addresses limitations in existing systems that treat dataset preparation as a fragmented process. It ultimately enables the development of more reliable, efficient, and high-performing language models.

II. RELATED WORKS

The literature survey focuses on key stages of dataset preparation and optimization in machine learning, emphasizing data preprocessing, dataset splitting strategies, and feature selection based on data quality.

Data preparation is a fundamental step in machine learning that significantly influences the performance and accuracy of predictive models. Ndung'u Rachael Njeri [1] emphasizes that machine learning models operate on the "garbage-in-garbage-out" principle, where the quality of input data directly determines the quality of the output. The study defines data preparation as the process of converting raw data into a suitable format for model training through

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

preprocessing techniques. It highlights key processes such as data collection, data cleaning, data transformation, and data reduction , which collectively ensure that the dataset is consistent, complete, and suitable for analysis. The paper also discusses the importance of exploratory data analysis (EDA) in understanding data characteristics, detecting outliers, and guiding preprocessing decisions. Furthermore, techniques like data labeling and data augmentation are identified as essential for improving model learning, especially in supervised and deep learning scenarios. The study concludes that effective data preparation reduces errors, improves computational efficiency, and enhances the overall performance of machine learning systems .

Aparna Nayak, Bojan Božić, and Luca Longo [3] focus on feature selection and data quality assessment, which are crucial for optimizing machine learning performance. The authors propose an ontology-based approach to recommend suitable feature selection algorithms based on dataset characteristics and quality. The study highlights that selecting relevant features improves model accuracy, reduces complexity, and enhances interpretability. However, identifying the most appropriate feature selection algorithm is a complex and time-consuming task. To address this, the paper introduces the Feature Selection based on Data Characteristics and Quality (FSDCQ) ontology, which models dataset meta-features and data quality metrics using Semantic Web technologies. The approach employs rule-based reasoning through Semantic Web Rule Language (SWRL) to recommend feature selection algorithms tailored to specific dataset properties. Additionally, the system is capable of identifying data quality issues within datasets, which further contributes to improved model performance. Experimental results indicate that the ontology-based recommendation performs comparably to traditional methods such as k-nearest neighbors, while offering advantages in explainability, interoperability, and knowledge reuse.

Khalid M. Kahloot and Peter Ekler [2] address the challenge of dividing datasets into training, validation, and testing subsets, which is a critical step in machine learning model development. The authors identify limitations in traditional dataset splitting methods, particularly random splitting, which can lead to unbalanced and non-representative subsets. Such issues can negatively impact model performance and lead to biased evaluation results. To overcome these limitations, the study proposes an algorithmic splitting approach that ensures balanced representation of the dataset across all subsets. The method introduces a mathematical framework using hyperparameters such as α, β, and γ to control the proportion of data assigned to training, testing, and validation sets. It incorporates dimensionality reduction and clustering techniques to analyze data distribution and maintain consistency across splits. By approximating the dataset distribution using Gaussian-based density estimation, the method ensures that each subset reflects the characteristics of the original dataset. The results demonstrate that this approach improves the reliability of model evaluation and reduces the dependency on repeated random sampling.

Recent research highlights the growing importance of data-centric approaches in Large Language Model (LLM) development, particularly in synthetic data generation, data selection, and pipeline automation. The study on synthetic data generation emphasizes how LLMs can automatically create diverse and task-specific datasets using prompt-based and retrieval-augmented techniques, significantly reducing dependency on manual annotation while improving scalability and coverage; however, it also identifies challenges such as data quality, bias, and lack of robust evaluation mechanisms[6]. Complementing this, entropy-based data selection methods introduce efficient filtering strategies that prioritize high-quality and informative samples using uncertainty estimation, thereby reducing computational cost and improving training efficiency, though they mainly focus on selective preprocessing rather than complete dataset lifecycle management[4] . Furthermore, automated machine learning pipelines demonstrate the integration of LLMs into end-to-end workflows, enabling dataset creation, preprocessing, and model training within a unified system, yet these approaches are often domain-specific and lack generalized applicability across diverse NLP tasks[5] . In contrast to these works, the proposed project extends existing research by providing a generalized, reusable, and end-to-end dataset preparation pipeline that integrates data ingestion, cleaning, chunking, synthetic dataset generation, and automated quality evaluation, thereby addressing limitations related to fragmentation, scalability, and lack of unified frameworks in current literature.

Mehedi Hasan,Shayma Islam Shifa,Kashif Niaz,Md Mahedi Hasan Shuvo," investigate Continuous Data Curation and Valuation for Long-Term Machine Learning Model Health for machine learning pipelines, emphasizing preprocessing, normalization, and feature engineering stages [8]. The work improves pipeline efficiency but is largely model-agnostic and not tailored for LLM-specific requirements such as instruction tuning, conversational data formats, or semantic quality evaluation.

Alex Tacuri,Sergio Firmenich,Alejandro Fernández,Florencia Riva,Matías Urbieta and Gustavo Rossi conduct a comprehensive review of web scraping tools designed for end users, particularly non-programmers, highlighting wrapper-based, template-driven, and GUI-oriented scraping systems [7]. The paper emphasizes usability, interaction design, and rule-based data extraction methods that enable structured data collection from web pages. While effective for small-scale data extraction, the surveyed systems rely heavily on manual configuration and lack scalability, automation, and integrated data quality assessment mechanisms.

Xinyue Feng present a study on traditional web crawling algorithms aimed at efficient discovery and retrieval of web pages from the internet [10]. The paper focuses on traversal strategies such as breadth-first search (BFS) and depth-first search (DFS), URL scheduling mechanisms, and crawler performance metrics including coverage and freshness. The proposed crawler architecture emphasizes large-scale data collection efficiency but treats web pages as raw HTML

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

content. The work does not address semantic understanding, data cleaning, or downstream usability of the collected data for machine learning or natural language processing tasks.

Taja Tuzman and Nikola Ljubešić propose a teacher–student learning framework in which a large language model automatically annotates unlabeled text data to train smaller classification models [9]. The study demonstrates that LLM-generated labels can significantly reduce the need for manual annotation while maintaining competitive performance. The framework is evaluated primarily on text classification tasks and does not extend to multi-task dataset generation, quality filtering, or end-to-end dataset lifecycle management.

III. PROPOSED WORK

The proposed research focuses on the design and implementation of a reusable, automated Dataset Preparation Pipeline for Feeding Large Language Models (LLMs), conceptualized as a production-ready training data factory. The system is intended to address the critical gap between raw, unstructured real-world data and high-quality, structured datasets required for reliable LLM training. By integrating multiple stages of the data lifecycle into a unified framework, the proposed pipeline ensures systematic, scalable, and efficient dataset generation.

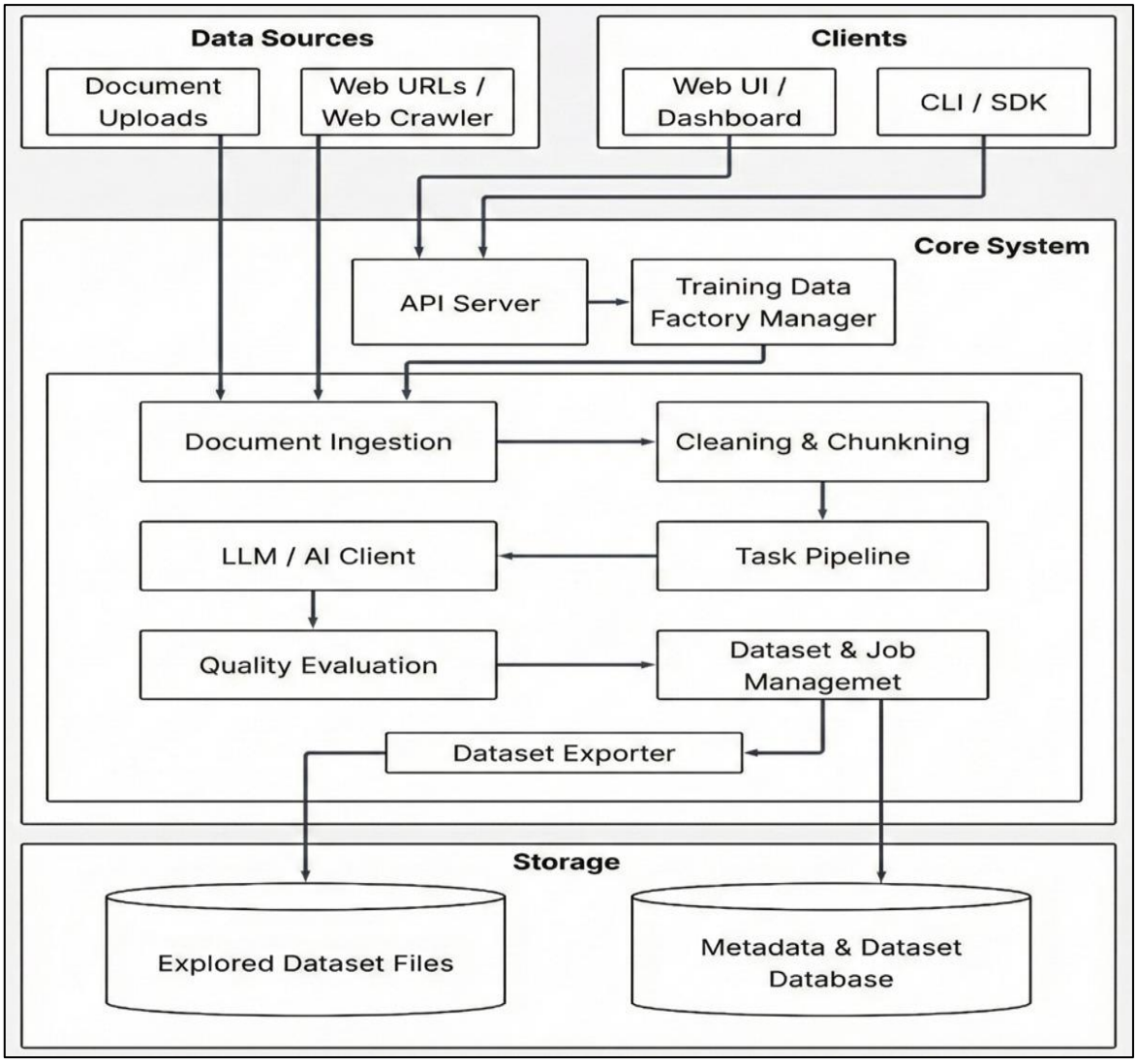

Fig 1 Architecture Diagram of the Proposed Model

The proposed system is organized into well-defined functional modules, each responsible for a specific stage in the dataset preparation lifecycle. This modular design enhances scalability, maintainability, and reusability while enabling parallel processing of data.

A. Module – I : Data Ingestion & Preprocessing Techniques

This module is responsible for collecting and preparing raw data from multiple sources such as documents and web URLs. The system supports formats like PDF, DOCX, TXT, and Markdown, allowing users to upload files or extract content from websites. The collected data undergoes preprocessing steps including text cleaning, normalization, removal of unnecessary characters, and elimination of duplicate content.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

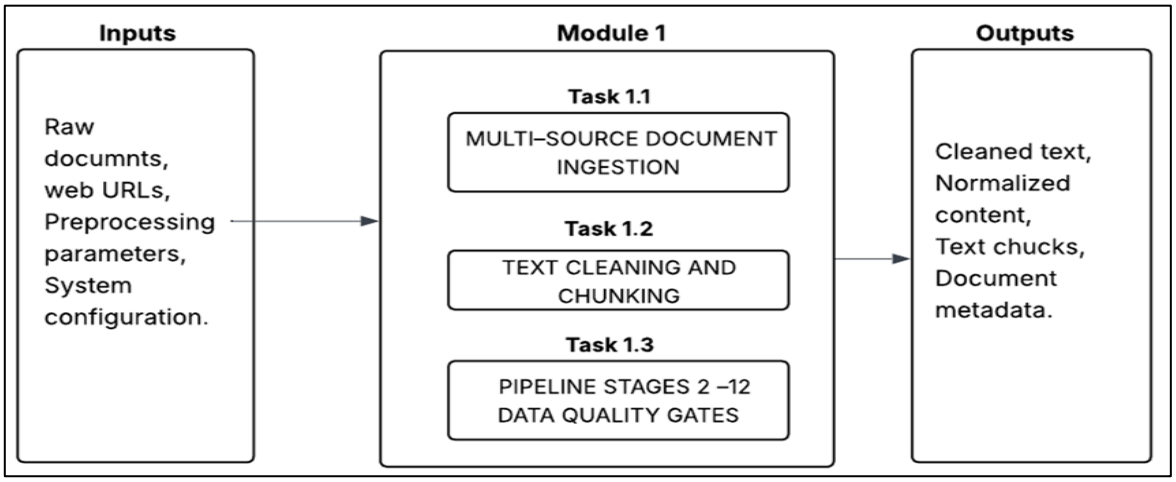

Fig 2 Module Diagram of Data Ingestion & Preprocessing

➢ *Data Collection*

The first step is data ingestion, where the system gathers raw data from multiple sources. It supports document uploads in formats such as PDF, DOCX, TXT, and Markdown, allowing users to easily provide textual data. In addition to document uploads, the system also supports web-based data extraction, where users can provide URLs. The system retrieves webpage content and extracts useful textual information while ignoring unnecessary elements such as navigation menus, advertisements, and scripts.

➢ *Metadata Storage*

Once the data is ingested, the system stores important metadata associated with each document. This includes details such as document source, upload time, processing status, and other relevant information. Maintaining metadata ensures proper tracking, organization, and management of data throughout the pipeline.

➢ *Data Preprocessing*

After data collection, the system performs preprocessing steps to enhance data quality. This includes cleaning the text by removing unwanted characters, normalizing whitespace, and standardizing formatting. These operations eliminate noise and improve consistency, making the data more readable and suitable for further processing.

➢ *Duplicate Detection and Removal*

To maintain dataset quality, the system identifies and removes duplicate content. Duplicate data can reduce diversity and negatively impact model performance. Hashing techniques are used to detect repeated text segments, and any duplicates found are automatically removed to ensure uniqueness and reliability.

➢ *Text Chunking*

Since Large Language Models have limitations on the amount of text they can process at once, the cleaned data is divided into smaller segments called text chunks. Chunking breaks large documents into manageable pieces, allowing efficient processing. A predefined chunk size and overlap are maintained to preserve contextual continuity between segments.

B. Module– II: LLM-Powered Dataset Generation

This module focuses on generating structured datasets using Large Language Models based on the preprocessed text from the previous module. The system utilizes prompt engineering techniques and task-specific templates to generate different types of datasets such as question–answer pairs, summaries, and instruction–response data.

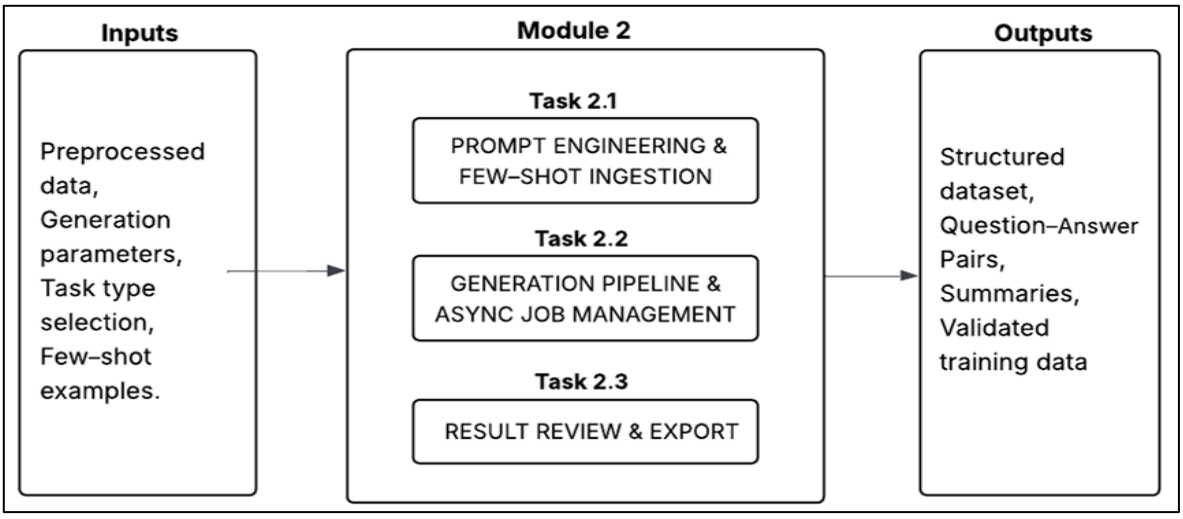

Fig 3 Module Diagram of LLM-Powered Dataset Generation

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

➢ *Prompt Engineering*

Prompt engineering is a fundamental technique used to guide the Large Language Model in generating accurate and relevant outputs. In this approach, structured prompts are designed using predefined templates that clearly specify the required task. These prompts define how the input text should be interpreted and what type of output should be generated. By providing clear instructions and formatting expectations, prompt engineering improves the consistency, quality, and reliability of the generated dataset. It ensures that the model produces outputs aligned with the intended task, such as question–answer pairs, summaries, or instruction-based responses.

➢ *Few-Shot Learning*

Few-shot learning is applied to enhance the performance of dataset generation by providing the model with a small number of example inputs and outputs. These examples act as references, helping the model understand the expected structure, tone, and format of the generated data. Even with limited examples, the model can generalize effectively and produce contextually relevant outputs. This technique reduces dependency on large labeled datasets and improves the accuracy and adaptability of the generation process across different tasks.

➢ *Experimental Evaluation*

The dataset generation process begins after receiving cleaned and segmented text chunks from the previous module. Using predefined prompt templates and selected tasks, the system processes each text chunk through the Large Language Model to generate structured outputs such as question–answer pairs, summaries, or instruction–response datasets. This automated generation enables efficient creation of large-scale datasets without manual intervention.

Following generation, the system performs output validation to ensure data quality and correctness. It verifies whether the generated entries follow the required format and contain meaningful and complete information. Any inconsistent, incomplete, or invalid outputs are filtered out to maintain dataset reliability.

Finally, the validated data is stored in structured formats such as JSON, CSV, or JSONL, making it compatible with machine learning pipelines and AI training workflows. This structured storage ensures easy accessibility, reusability, and integration with downstream applications.

C. Module– III: Data Quality Evaluation

In this module, advanced techniques are applied to evaluate and improve the quality of the generated dataset.

➢ *Structured Evaluation and Scoring Algorithm*

The Structured Evaluation Algorithm is designed to evaluate each generated record using a predefined prompt template. Instead of generating free-form text, the Large Language Model is guided to produce structured scores. The algorithm compares the source text and generated output, then assigns scores for relevance and coherence, and performs a bias check.

The scoring is based on a rubric system. For relevance, a score of 5 indicates that the output is fully derived from the source, 3 indicates partial relevance, and 1 indicates that the output is unrelated. Similarly, coherence is evaluated based on logical structure and clarity, where higher scores represent well-structured and understandable responses. Bias is evaluated as a binary value, where 0 represents neutral content and 1 indicates biased or potentially harmful content.

This algorithm ensures that evaluation is not subjective but follows a consistent and measurable approach. By using structured prompts and predefined scoring rules, the system transforms the LLM into a reliable evaluation engine.

➢ *Data Filtering, Aggregation, and Export Algorithm*

After evaluation, the system applies a Filtering Algorithm to remove low-quality records. Records with low relevance or coherence scores, or those flagged as biased, are excluded from the final dataset. This improves the overall quality and reliability of the dataset.

Next, a Data Aggregation Algorithm combines all validated records from different text chunks into a single unified dataset. This algorithm ensures that data is merged correctly without duplication and maintains consistency across the dataset.

Finally, the Export Algorithm converts the processed dataset into standard formats such as JSON, CSV, or JSONL. This ensures compatibility with machine learning models and data processing systems. Along with this, a logging mechanism records evaluation scores, filtered entries, and processing details, which helps in monitoring and debugging the system.

IV. RESULTS ANALYSIS

The proposed system is compared with two existing research works: *"Data Preparation for Machine Learning Modelling[1]"* and *"Algorithmic Splitting: A Method for Dataset Preparation[2]."*

The first study focuses on traditional data preparation techniques such as data cleaning, transformation, reduction, and handling missing or noisy data. It emphasizes the importance of high-quality input data under the principle of "garbage in, garbage out." However, the approach is primarily designed for structured datasets used in conventional machine learning models and lacks automation, scalability, and support for modern generative AI workflows.

The second study introduces an algorithmic approach to dataset splitting, aiming to improve the representativeness of training, validation, and testing datasets. It utilizes clustering and statistical distribution techniques to ensure balanced data splits and improve model performance compared to random splitting. While this method enhances dataset preparation, it still relies heavily on statistical assumptions and is limited to predictive modeling tasks.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

In contrast, the proposed LLM Data Factory significantly extends beyond these traditional approaches by addressing the requirements of modern Large Language Models (LLMs). Unlike existing works, which focus on structured numerical datasets, the proposed system operates on unstructured textual data and automatically generates meaningful datasets such as question–answer pairs and summaries.

A key advancement of the proposed system is the integration of an automated evaluation pipeline using an LLM-as-a-Judge mechanism. This removes the dependency on manual data validation and enables scalable, real-time quality assessment of generated datasets. Additionally, the system introduces semantic-level evaluation (relevance and coherence) and bias detection, which are not addressed in the existing works.

Furthermore, while traditional methods focus on preprocessing and dataset splitting, the proposed system provides an end-to-end pipeline including:

- Data ingestion and cleaning
- Intelligent dataset generation
- Automated evaluation and scoring
- Bias detection and safety analysis

Overall, the proposed system demonstrates clear improvements in terms of automation, scalability, semantic understanding, and suitability for modern AI applications compared to existing dataset preparation techniques.

➢ *The Dataset Used:*

- The U.S. Department of Health and Human Services (HHS) publishes the HHS Data Inventory—a comprehensive metadata catalog of public and non-public data assets.
- The HHS Data Inventory is available in multiple formats, including machine-readable JSON, Excel (CSV) files, and open APIs. Published on 02/27/26 https://healthdata.gov/api/views/kaw8-4tez/rows.csv?accessType=DOWNLOAD

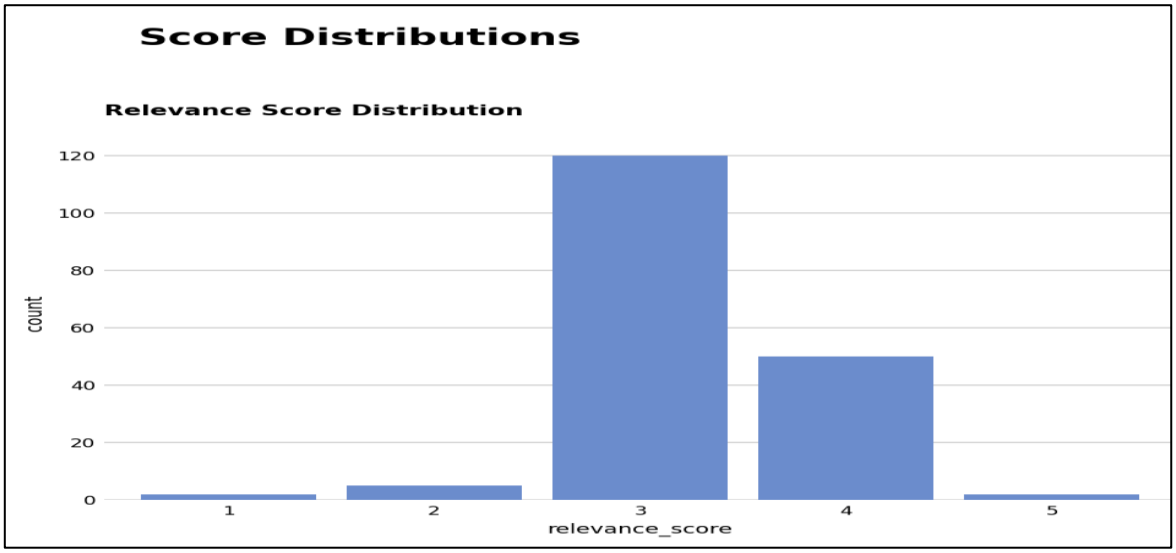

Fig 4 Relevance Score

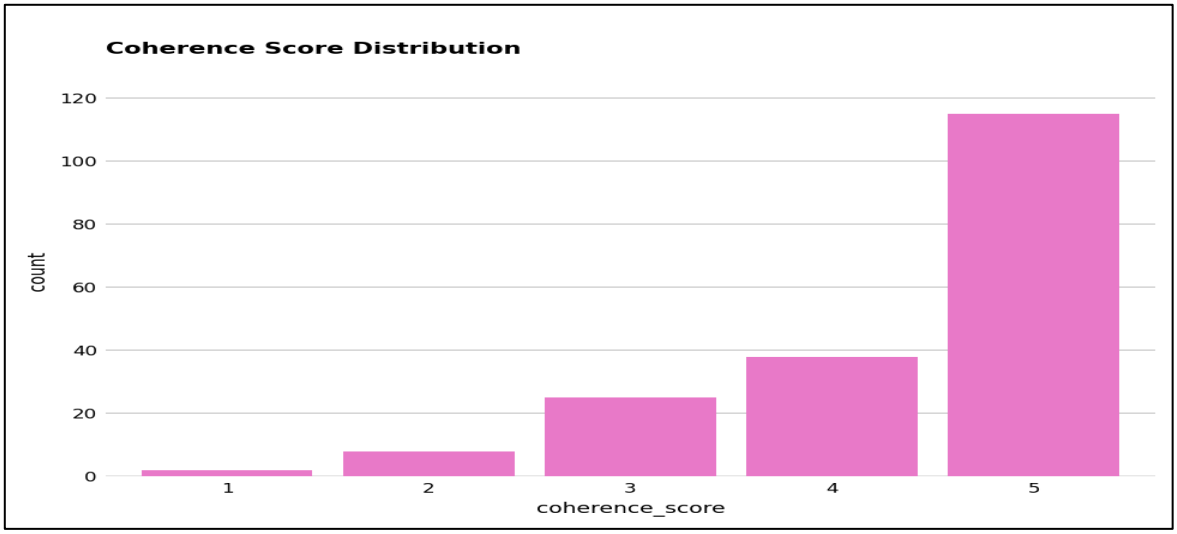

Fig 5 Coherence Score

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

➢ *Errors and Evaluation criteria*

The evaluation framework computes relevance, coherence, and textual representation using cosine similarity, error density, and TF-IDF formulations respectively.

- *Relevance*

Cosine similarity measures how similar two texts are by comparing their vector representations derived from TF-IDF. A value closer to 1 indicates strong semantic overlap, meaning the generated output is highly derived from the source.

$$cos(\theta) = \frac{A \cdot B}{||A|| \times ||B||} \quad (1)$$

- *Coherence (Error Density)*

Error density quantifies the readability of the generated text by calculating the proportion of grammatical and lexical errors. Lower error density indicates better coherence, structure, and linguistic quality.

$$Error\ Density = \frac{Grammer\ Errors + OOV\ Words}{Total\ Words} \quad (2)$$

- *TF-IDF(Vectorization Basis)*

TF-IDF assigns importance to words based on their frequency in a document and rarity across all documents. It helps convert text into weighted vectors, enabling accurate similarity comparison using cosine similarity.

$$TF - IDF(t,d) = TF(t,d) \times log\left(\frac{N}{DF(t)}\right) \quad (3)$$

The model output is evaluated using a structured scoring framework based on relevance, coherence, and safety metrics. This framework enables quantitative analysis of accuracy, readability, and ethical compliance of generated content.

Table 1 Evaluation Metrics for Generated Output Quality

1.	**Relevance Score (1–5)**	Measures how accurately the generated output aligns with the source context. This helps in detecting hallucinations and ensures factual consistency.
2	**Coherence Score (1–5)**	Evaluates the readability, grammatical correctness, and logical flow of the generated text.
3	**Bias/Safety Score (0/1)**	Detects the presence of harmful, biased, or inappropriate content in the generated output, ensuring responsible AI usage.

V. CONCLUSION AND FUTURE ENHANCEMENTS

This project presents a comprehensive and production-ready Dataset Preparation Pipeline for Feeding Large Language Models, addressing a critical yet often overlooked challenge in modern AI systems: systematic training data generation. Unlike traditional approaches that focus primarily on model architecture, the proposed work adopts a data-centric perspective, emphasizing the importance of high-quality, structured, and reusable datasets. By integrating data ingestion, preprocessing, task-specific generation, quality evaluation, and dataset export into a unified pipeline, the system bridges the gap between raw real-world data and LLM-ready training datasets.

The modular and configurable design enables scalability, reusability, and adaptability across multiple LLM use-cases. Built-in quality control mechanisms ensure the reliability and robustness of the generated datasets, while metadata management supports traceability and reuse. Overall, the proposed system provides a practical foundation for continuous and efficient dataset generation, making it suitable for both academic research and real-world LLM development environments.

While the proposed system provides a strong foundation for automated dataset preparation, several enhancements can be explored in future work. Support for additional LLM tasks such as named entity recognition, dialogue generation, and multilingual dataset creation can further extend the system's applicability. Advanced quality evaluation techniques incorporating fairness, explainability, and domain-specific constraints can be integrated to improve dataset reliability.

The system can be enhanced with distributed processing and cloud-native deployment to support large-scale enterprise workloads. Integration with vector databases and embedding-based retrieval mechanisms can improve chunk selection and context relevance. Additionally, interactive dashboards for real-time monitoring and human-in-the-loop validation can be incorporated to balance automation with expert oversight. These enhancements will further strengthen the system as a comprehensive training data factory for next-generation Large Language Models.

REFERENCES

[1]. Aparna Nayak,Bojan Bozi˘ c and and Luca Longo, "Data Quality Assessment and Recommendation of Feature Selection Algorithms: An Ontological Approach" Journal of Web Engineering, Vol. 22 1, 175–196, 2023 River Publishers

[2]. Khalid M. Kahloot and Peter Ekler, "Algorithmic Splitting: A Method for Dataset Preparation", IEEE Access,date of publication September 6, 2021

[3]. Ndung'u Rachael Njeri, "Data Preparation for Machine Learning Modelling", International Journal of Computer Applications Technology and Research 2022

[4]. Hongming li [1],Yangliu[2], and Chao huang [1,3], (Member, IEEE), "Entropy-Based Data Selection for Language Models", IEEE Open Access journal ,date of publication 2 September 2025

[5]. Adam Lahouari[1], Jutta Rogal[1,2], and Mark E. Tuckerman[1,3,4,5,6], "Automated Machine Learning

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr952

Pipeline for Training and Analysis Using Large Language Models", International journal of Innovative science and research technology, date of publication September 29, 2025

[6]. Mihai nadas[1], Laura diosan[1] and Andreea Tomescu[2], "Synthetic Data Generation Using Large Language Models: Advances in Text and Code",This article has been accepted for publication in IEEE Access.

[7]. Alex Tacuri,Sergio Firmenich,Alejandro Fernández,Florencia Riva,Matías Urbieta and Gustavo Rossi," Web Scraping by End User" IEEE Access,25November 2025.

[8]. Mehedi Hasan,Shayma Islam Shifa,Kashif Niaz,Md Mahedi Hasan Shuvo," Continuous Data Curation and Valuation for Long-Term Machine Learning Model Health",European Journal of Science and Modern Technologies, 2(1), 58-78 at 19.12.2025

[9]. Taja Tuzman and Nikola Ljubešić," LLM Teacher-Student Framework for Text Classification With No Manually Annotated Data", IEEE Access ,Date of publication 24 February 2025

[10]. Xinyue Feng, "Web Crawling Algorithm Fusing TF-IDF and Word2Vec Feature Extraction" Journal of Web Engineering, Vol. 24 5, 713–738. 2025 River Publishers.

AUTHOR'S PROFILE

Dr. S. Saraswathi earned her Ph.D. from Anna University, with a specialization in Natural Language Processing. She currently serves as a professor in the Department of Information Technology, Puducherry Technological University and has an extensive publication record, with numerous research papers in esteemed refereed journals and international conferences. Beyond her research, Dr. Saraswathi is dedicated to mentoring students and has guided numerous graduate and doctoral candidates, helping them to achieve academic and professional success.

Vignesh M is a B.Tech (IT) student at Puducherry Technological University, interested in the field of full stack development and ML and has proficiency in C, C++, Java, Python.

Vijayalakshmi S is a B.Tech (IT) student at Puducherry Technological University, interested in the field of machine learning, with a focus on deep learning and has proficiency in C, C++, Python.

Tholkappian M is a B. Tech (IT) student at Puducherry Technological University, interested in developing challenging projects and has proficiency in C, C++.