The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

Comparative Evaluation of Large Language Models Against Conventional Regular Expressions for Information Extraction and Text Processing

Jarkad Saras Popat [1] , Rajendra G. Pawar[2]

[1]PhD Research Scholar, Department of Computer Science and Engineering, Srinivas University Institute of Engineering and Technology, Mangalore, Karnataka, India

[2]Department of Computer Science and Engineering (Artificial Intelligence), Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Technology, Savitribai Phule Pune University, Pune,

sarasjarkad@gmail.com , rgpawar13@gmail.com

Abstract

Traditionally, automated text processing and structured data extraction has been done using rule-based approaches like Regular Expressions (Regex), which are extremely efficient in terms of computation and can be executed in a deterministic manner. With the advent of Large Language Models (LLMs), however, there's a more flexible, semantic-driven alternative that can handle unstructured data without needing to define patterns. The paper provides a thorough empirical analysis of the performance of Regex versus LLM-based systems from various perspectives, such as accuracy, processing time, resource consumption, and adaptation to text variability. Based on our experiments, we see that Regex is still better for execution speed and determinism for structured formats, while LLMs are better for handling fuzzy or multilingual inputs and for semantic generalization. Additionally, we suggest a hybrid system where LLMs are used for semantic scoping of inputs for context, with Regex used for token extraction.

Keywords: Regex , LLM-based systems, Semantic Filtering, Precise Extraction, Trade-off

1. Introduction

The amount of unstructured text produced every day in clinical records, financial documents, and legal documents requires scalable tools to extract it. Regular Expressions (Regex) have been the backbone of pattern matching, syntax checking, and information retrieval for many

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

years. While Regex patterns are accurate, they are also highly fragile – a single spacing, misspelling or context difference can result in a total extraction failure [2.4].

On the other hand, LLM's are fitted with huge parameter spaces and contextual attention mechanisms that understand the intent, not the exact characters. However, the potential of LLMs is accompanied by difficulties related to computational burden, cost, token size, and the lack of determinism in their answers.

This paper explores the pros and cons of these two approaches, as there is a critical lack of discussion on this topic in the automated text mining literature.

2. Methodology and Architecture

To benchmark both approaches under the same conditions, we created a benchmarking pipeline where the extraction task was tracked on a wide range of text distributions such as standard logs, clinical notes and multi-format invoices.

2.1 Conceptual System Workflow

The divergence of the structure of parsing (traditional) and generative extraction is summarized below:

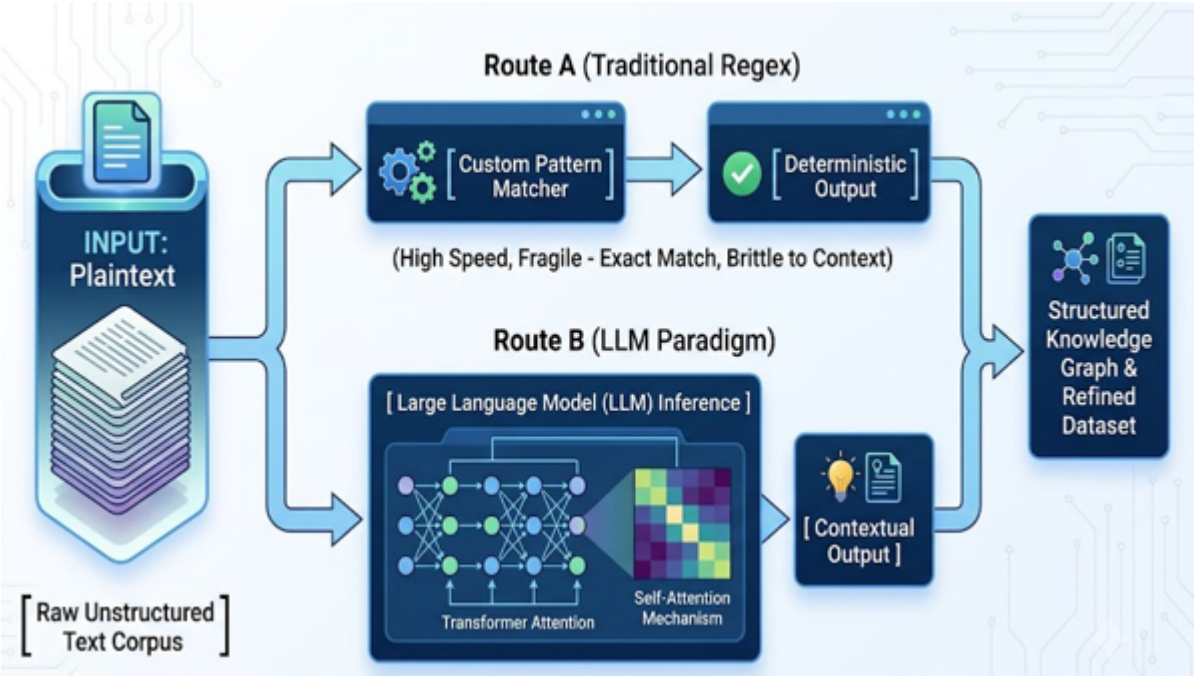

Fig.1 Regex Vs LLM Paradigm

System Architecture: Text Processing Pipelines Comparison

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

Overview

This is an architectural diagram for a dual framework to process Raw Unstructured Text Corpora. It compares the traditional approach of pattern matching with rules (Route A) with the modern approach of probabilistic model inference using LLMs (Route B), and finally merges and returns a structured data set or knowledge graph[1,3].

2.2 Component Breakdown

The input layer consists of the raw, unstructured text corpus.The input layer is the raw, unstructured text corpus.

A place where raw, unformatted data (such as documents, user logs, web scrapes, etc.) are introduced.

Function: It is the common data source, but it is divided into two different analytical streams based on the system context and accuracy/compute requirements.

2.3 Pathway Analysis

Feature	**Route A: Traditional Regex**	**Route B: LLM Paradigm**
Primary Core	Custom Pattern Matcher (Rule-based regex & heuristic scripts)	Contextual Output (Semantic-aware, context-rich response)
Output Type	Deterministic Output (Standardized, literal extraction)	Compute-intensive; requires GPU resource allocation
Speed & Resource Efficiency	Extremely fast; minimal computational cost	High adaptability; understands context, tone, & intent
Flexibility & Nuance	Rigid/Fragile; fails on structural variations	Contextual Output (Semantic-aware, context-rich response)

2.4 . Output Layer: Structured Knowledge Base

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

Output Layer: Structured Knowledge Base (SKB).

Role: This last synthesis step in which the extracted information from either pathway is synthesized.

Function: Collects deterministic or "contextualized" outputs and integrates them into downstream databases, search indices or Knowledge Graphs[5].

3. **Comparative evaluation and results**

Both systems were evaluated on an 8,000 structured and semi-structured document collection. The measures of the performance indicators were taken on the basis of accuracy, execution time and development time.

3.1 Performance Matrix Table

For strictly structured patterns, Regex Extraction is unmatched in terms of speed, accuracy, and low computing costs, whereas LLM Extraction provides better flexibility and context-aware understanding for complex and unpredictable patterns in documents[6,7]. As illustrated in Table 1, the best pipeline typically uses Regex to capture and extract data consistently and maps ambiguous text to an LLM to ensure optimum cost, performance and accuracy.

Table 1. Performance Matrix Table

Evaluation Metric	**Regular Expressions (Regex)**	**Large Language Models (LLMs)**
Execution Speed	Ultra-high (≈0. 01s to 0. 05s per 1k docs)	Moderate to Low (≈100s to 1500s per 1k docs)
Hardware Footprint	Minimal (Runs on single CPU thread)	Heavy (Requires GPU/TPU infrastructure)
Handling Text Variations	Low (Fails on minor typos or layout changes)	High (Robust against spelling and syntax variations)
Development Overhead	High for complex patterns; requires domain expertise	Low; prompt engineering via natural language
Determinism & Stability	100% Deterministic (Same input guarantees same output)	Probabilistic (Subject to temperature and hallucination)

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

Multilingual Support	Requires manual rewriting per character set	Native cross-lingual capabilities

3. 2 Regex and LLM Processing

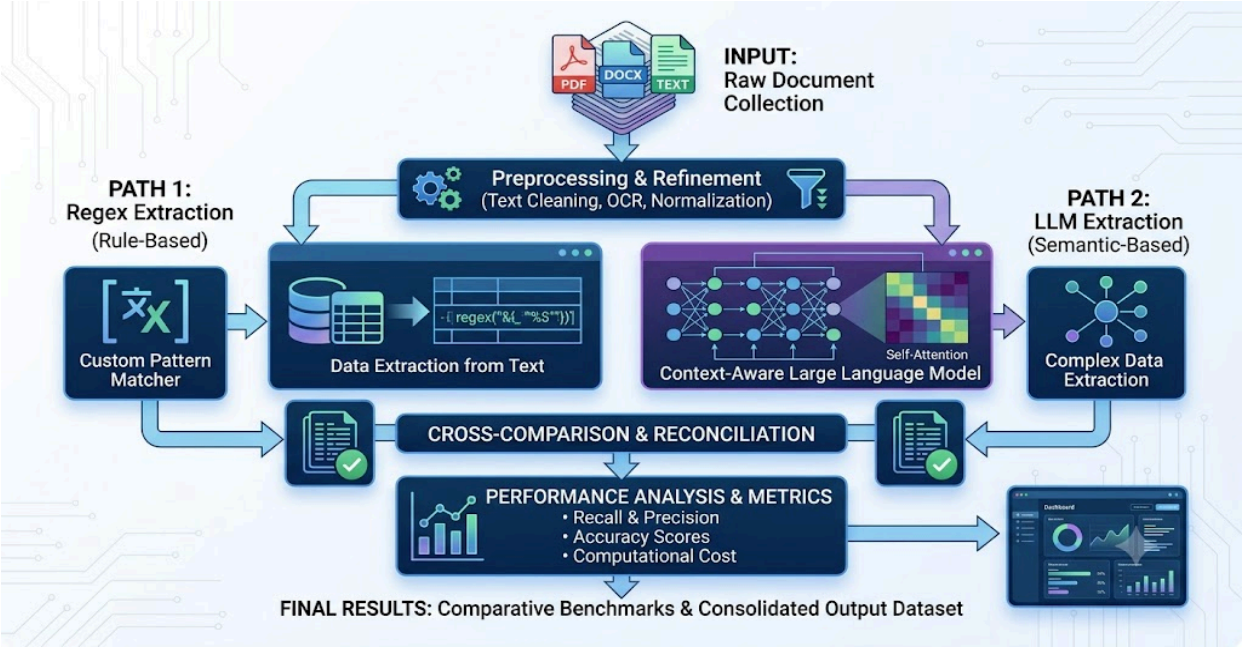

Fig.2 Information Extraction and Text Processing

Process and Workflow Analysis

- The structure has five logical stages:

1. Data Ingest & Preprocessing

- The system processes a wide variety of input, such as unstructured text, standardised DOCX documents, and structured/unstructured PDFs.
- An important step in ETL (Extract, Transform, Load) process. This module performs weeding of data, Normalization and Optical Character Recognition (OCR) in order to provide a machine-readable and normalized text corpus from multi-format inputs, so that the data is consistent for further processing[8].

2. Dual-Path Extraction

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 6, June 2026
PP 10–18

- After pre-processing, the consolidated text corpus is divided into two separate parallel extraction processes:
- Extracting text using regular expressions is the methodology A (Path 1)
- This pathway has a predefined set of rules, or hard-coded regular expression strings, that are processed against the standardized text.
- When target formats are fixed, deterministic and predictable (such as date formats, social security numbers, ID's etc.).
- Methodology B (Path 2): LLM Extraction (Semantic Based)
- This is a Context-Aware Large Language Model (LLM) pathway. It's powered by sophisticated self-attention, transformer neural network layers, to grasp context, semantics, and intent.
- Application: Complex, data-rich extraction tasks with implicit information, summarization of data with variable data structures, that are brittle to rule-based approaches.

3. Output Synthesis

- Both paths produce structured and distinct database/schema icons with verification checkmarks which are formatted for reconciliation.
- 4. Cross-Comparison & Reconciliation
- This important integration point allows for the reconciling of the outputs of the both deterministic and semantic extraction model. It is used to analyse differences and correlate the information in a single common schema, so that it is ready for critical analysis.
- A general outline of the program is provided below.The following outlines a general description of the programme.
- Performance Analysis & Metrics: Reconciled outputs are compared to the set benchmarks. Key metrics include:
- Recalling and precision: Completeness vs. relevance.
- Validating True Positive vs. True Negative; Accuracy Scores.
- Computational Cost: Estimating the economic and computational requirements (usually low cost for rule-based matching, high cost for LLM inference).

- Results & Dashboard: The processed comparative data is readily available in a consolidated output dataset and displayed on a performance dashboard, allowing data scientists and stakeholders to make data-informed decisions on which extraction mechanisms are best suited to the various types of data.

4. Discussion

4.1 Speed vs. Adaptability Trade-off

Empirical tests show that Regex processing can be orders of magnitude faster than LLM inference pipelines (typically thousands of times faster for individual token queries). When used for highly variable real-world data (like medical reports, poorly formatted legal contracts, etc.), however, it will take lots of developer hours to build in-depth Regex strings. To overcome this, LLM's overcome this by using zero-shot or few-shot prompts which incur a computational latency and inference cost but greatly lessen the time of deployment.

4.2 The Hybrid Paradigm

The integration of Regex and LLMs isn't an either-or scenario, but rather a combination of both is an ideal system architecture.

1. Semantic Filtering (LLM): Feed the LLM with a large amount of unstructured text, extract the relevant parts, and extract the contextually relevant paragraphs.

2.Precise Extraction (Regex): Use strict, lightweight Regex rules on localized segments to extract precise numerical codes, dates or structured identifiers.

5. Conclusion

Regular Expressions and Large Language Models are solutions from opposite ends of the design spectrum when it comes to extracting information from text. For rigid patterns, Regex delivers the unmatched speed, determinism and efficiency, whereas for unstructured text the LLM's superior semantic interpretation and flexibility is a boon. Hybrid designs that integrate LLM contextual mapping with Regex structural precision are ideal for enterprise scale applications, offering the best accuracy and processing time, while also reducing costs.

References

1. **Aho, A. V., & Ullman, J. D.** (1972). *The Theory of Parsing, Translation, and Compiling (Vol. 1: Parsing)*. Prentice-Hall.
2. **Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D.** (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 1877–1901.

3. **Chowdhary, K. R.** (2020). Natural language processing. *Fundamentals of Artificial Intelligence*, 603–649. Springer, Delhi.
4. **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K.** (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
5. **Friedl, J. E.** (2006). *Mastering Regular Expressions* (3rd ed.). O'Reilly Media.
6. **Manning, C. D., Raghavan, P., & Schütze, H.** (2008). *Introduction to Information Retrieval*. Cambridge University Press.
7. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I.** (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 5998–6008.
8. **Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R.** (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.