SentimentPulse: A Concurrent Multi-Platform System for Social Media Sentiment Monitoring with LLM-Based Interpretation

Gabriel A. León-Paredes, Erika C. Villa-Quishpi, Jorge E. Márquez-Chávez, Erick F. Zhigue-Granda
Research Group in Cloud Computing, Smart Cities and High Performance Computing
Universidad Politécnica Salesiana, Cuenca, Ecuador

***Abstract*—Monitoring public perception on social media is increasingly important for detecting reputational risks and communication opportunities in rapidly evolving digital environments. However, operational sentiment monitoring remains challenging due to platform fragmentation, heterogeneous data formats, and the need to generate interpretable reports quickly when specialized analysts are not available. This study presents SentimentPulse, a web-based system for multi-platform sentiment monitoring driven by a user-defined query. The system integrates concurrent data extraction from X, Facebook, LinkedIn, and Instagram with large language model (LLM) based sentiment classification and automated executive storytelling generation. The architecture combines parallel scraping processes for data acquisition with multithreaded LLM inference to improve throughput, while structured persistence enables job tracking and cross-platform analysis. The system operates under practical constraints, including platform-specific access limitations, dynamic content availability, and dependency on external LLM services, which may introduce variability in response times and outputs. The evaluation is conducted under controlled experimental conditions using fixed query limits and asynchronous execution settings, and results should be interpreted within these operational boundaries. Experimental results from two anonymized case studies demonstrate the effectiveness and operational performance of the approach. In the first case study, the system processed 1032 social media interactions and produced a sentiment distribution of 49.0% positive, 36.6% negative, and 14.2% neutral, with a manual validation accuracy of 0.88. In the second case study, the pipeline processed 1121 records, with parallel scraping accounting for the majority of the runtime and LLM inference achieving a throughput of 12.08 items per second. These results show that combining concurrent multi-platform extraction with LLM-based interpretation enables practical and interpretable social listening workflows, while highlighting the importance of considering system-level constraints when deploying such solutions in real-world environments.**

***Keywords*—Sentiment analysis; social media monitoring; large language models; social listening systems; multi-platform analytics**

I. Introduction

Sentiment analysis is a central aspect of opinion mining. It seeks to automatically identify the polarity and subjectivity present in text, with applications in perception monitoring, customer service, market research, and social analysis [1], [2], [3], [4]. In recent years, the availability of user-generated content on social media has made this task a practical tool for understanding collective dynamics in (almost) real-time, driving its adoption in scenarios where rapid response is critical. Classic examples show that aggregated sentiment signals on social platforms can correlate with external phenomena such as financial markets or electoral behavior [5], [6], [7].

However, bringing sentiment analysis into an operational environment presents challenges that cannot be solved with a manual approach or isolated tools. First, data are distributed across multiple platforms, each with different formats, access mechanisms, and interaction dynamics, making it difficult to collect content in a consistent and comparable manner across networks [8], [9]. Second, in corporate communication or marketing contexts, interest is not limited to overall percentages: early detection of reputational risks and shifts in tone is required, especially during sensitive events where conversations can escalate quickly [10], [11]. Consequently, it becomes important not only to classify sentiment but also to explain patterns and differentiate results by platform in order to support concrete decisions. Despite the extensive literature on sentiment classification models, comparatively fewer studies focus on the design and evaluation of complete operational systems that integrate multi-platform data acquisition, scalable processing, and decision-oriented interpretation.

This project proposes a unified web application for multi-platform sentiment monitoring around a user-defined topic. The solution integrates (1) concurrent extraction from X, Facebook, LinkedIn, and Instagram, (2) consolidation of content into a common format, (3) global and per-platform sentiment classification, and (4) automatic generation of narrative summaries that interpret quantitative sentiment indicators to support executive-level decision making. To improve robustness in the face of informal language, regional variation, and changing context, a large-scale language model (LLM) is incorporated to assign sentiment labels and produce short explanations for each text. This is relevant because recent evidence suggests that, although LLMs can be competitive in simple tasks and few-shot scenarios, their performance varies when more complex phenomena or additional structure are required, making prompt design and careful evaluation important [12], [13]. Additionally, recent studies have evaluated conversational models (e.g., ChatGPT) as sentiment classifiers, reporting competitive results in certain settings but also variability depending on the prompt, domain, and task formulation, which motivates validation and stability control practices in real-world deployments [14], [15].

The main contributions of this work are: 1) a concurrent hybrid pipeline that combines multiprocessing for extraction

and multi-threading for LLM inference, 2) a persistence and traceability architecture based on MongoDB that enables status and history tracking by topic, and 3) a web interface that integrates search, visualization, data exploration, and executive narrative within a single workflow.

The remainder of this study is organized as follows: Section II reviews related work; Section III presents the methodology and architecture; Section IV reports performance and quality results; and Section V concludes with recommendations.

II. Related Work

Social media sentiment analysis has evolved from lexicon-based and supervised machine learning methods to deep learning and, more recently, large language model (LLM) based approaches. Nip and Berthelier (2024) [16] summarize this evolution and emphasize that sentiment analysis in social platforms extends beyond polarity detection by incorporating multi-modality, temporal dynamics, interaction structures, network effects, and sentiment propagation. They also outline operational challenges that remain prominent in practice, including high-volume data streams, linguistic heterogeneity, fragmented context, and noisy or automated content. Importantly, they discuss limitations that are particularly relevant when LLMs are used in deployed systems, such as output variability, interpretability constraints, bias due to alignment, and reproducibility concerns.

A substantial line of work focuses on improving sentiment classification quality with neural architectures. Katalinić and Dunđer (2025) [17] study sentiment analysis and anomaly detection in crisis-related tweets using 189,626 English-language messages from the 2023 Turkey–Syria earthquake. Their BERT-based sentiment classifier reports 91% precision, 85% recall, and 88% F1, outperforming a logistic regression baseline (54% F1). For anomaly detection, their autoencoder achieves 89% accuracy and 88% precision, exceeding an isolation forest baseline (69% precision, 64% recall), and ablations indicate that attention yields statistically significant gains ($p = 0.011$). While these results demonstrate strong neural performance for crisis monitoring, the scope is limited to a single platform and does not address multi-platform acquisition or end-to-end operational integration.

Complementing individual studies, Sharma et al. (2025) [18] review 178 papers published between 2014 and 2024 and categorize sentiment tasks into document-level, sentence-level, aspect-based, multi-lingual, multi-modal, and emotion-related settings. Their synthesis reports that transformer-based models, particularly BERT variants, typically reach 88–96% accuracy and often exceed 85% F1 depending on dataset and task difficulty, and they note that LLM-based approaches can surpass 96% accuracy in some domains. Despite these advances, the review highlights persistent challenges, such as sarcasm/irony, domain adaptation, multilingual variation, dataset bias, computational cost, and limited transparency, and observes that fewer works study system-level concerns such as real-time pipelines and deployment constraints.

In addition to model-centric approaches, recent work has explored the use of web-based systems for sentiment analysis in social media environments. Parra-Zambrano and León-Paredes (2024) [14] present a system that integrates natural language processing techniques and large language models, including ChatGPT, to analyse public opinion on political figures in Ecuador. Their approach achieves accuracy levels of up to 95% in classifying sentiments into positive, neutral, and negative categories, demonstrating the effectiveness of combining NLP pipelines with LLM-based components in real-world social media scenarios. These results highlight the potential of hybrid architectures that incorporate LLMs within broader analytical pipelines, while also emphasizing the importance of system-level design for handling heterogeneous and large-scale social media data.

Stability and reproducibility are increasingly recognized as deployment-critical properties. Ouyang et al. (2024) [15] analyze ChatGPT-based sentiment analysis through an AI quality management lens, distinguishing operational uncertainty from model robustness. Using three-way sentiment labels on 983 Amazon review samples and 1101 SST samples, they show that timing and model-version changes can lead to different confusion matrices on the same test set across API releases (e.g., 2023-06, 2023-12, 2024-01), complicating reproducibility. They also evaluate adversarial perturbations (typo, synonym, homoglyph, homophone) with accuracy and attack success rate (ASR), reporting that synonym substitution is the strongest perturbation and that shorter texts tend to be more vulnerable than longer reviews. These results motivate stability-aware practices such as prompt standardization, version tracking, and continuous monitoring in LLM-based pipelines.

Overall, prior work either (1) surveys the methodological landscape and challenges [16], [18], (2) improves model-level performance primarily within a single platform or benchmark setting [17], or (3) evaluates LLMs as sentiment analysers with an emphasis on prompting, robustness, and stability [14], [15]. In contrast, our work focuses on an end-to-end, deployable social listening system that integrates concurrent multi-platform acquisition (X, Facebook, LinkedIn, and Instagram), LLM-based sentiment labelling with per-item explanations, persistence for traceability (corpus, job status, and aggregated analytics), and an executive-oriented storytelling layer for cross-platform interpretation. This system-level contribution targets operational constraints—throughput, observability, and comparability across networks—that are only partially addressed in model-centric studies.

III. Methodology

This section describes the implementation of SentimentPulse at a system level to support understanding and reproducibility. The architecture is presented through its layered design, the processing pipeline, the concurrency strategy, and the REST workflow that allows the frontend to trigger jobs, monitor progress, and retrieve results.

A. Layered System Design

SentimentPulse is organized as a layered architecture connected through a REST API. Fig. 1 summarizes the main components and their interactions, including the API endpoints, background execution model, parallel data extraction

modules, large language model interpretation, natural language processing and analysis, and the database persistence layer.

At the presentation layer, a Vue 3 single-page application provides four main views: search, progress tracking, dashboard visualization, and data exploration. Through these interfaces, users can initiate sentiment analysis jobs for a given topic and consume JSON responses generated by the backend to render metrics, charts, and narrative summaries of the results.

The application logic is exposed through a FastAPI backend that implements the REST interface and manages asynchronous execution of the analysis pipeline. Jobs are executed using background tasks, allowing the system to process long-running operations without blocking the API. During execution, job progress and status information are stored in the persistence layer, enabling polling-based monitoring and improving system observability. Table I summarizes the REST endpoints used in the system workflow.

TABLE I. REST ENDPOINTS USED BY SENTIMENTPULSE

Endpoint	Method	Purpose
`/scrape`	POST	Start a job in background execution
`/status/topic`	GET	Retrieve per-platform progress and phase status
`/results/topic`	GET	Retrieve aggregated outputs and storytelling
`/history`	GET	Retrieve previous runs from stored analyses
`/cancel`	POST	Mark a job as cancelled in the database
`/history/topic`	DELETE	Delete analysis and associated posts for a topic

At the orchestration level, the central function `run_pipeline` coordinates the complete workflow of the system. This component manages the transition between processing phases, aggregates outputs obtained from different social media platforms, and transforms them into a unified tabular schema that allows consistent downstream analysis.

The data extraction stage collects posts and comments related to a user-defined topic. To achieve this, the system launches one Playwright-based scraper per social media platform. Each scraper operates with persistent sessions and collects relevant textual content while preserving metadata needed for subsequent analysis.

Once the textual data has been collected, the interpretation stage applies a large language model to each text unit. The model produces both a sentiment label and a short explanation, which improves interpretability and allows the system to provide traceable reasoning for each classification.

Following interpretation, the NLP and analysis stage processes the labeled texts using a traditional natural language processing pipeline. This stage extracts salient terms and generates aggregated indicators, including global sentiment metrics and platform-specific distributions. These results are then used to produce narrative summaries that facilitate high-level interpretation of the analysis outcomes.

Finally, all intermediate and final outputs are stored in MongoDB. The database maintains the structured corpus, job status information, execution times, and aggregated analytical results. This persistence layer enables efficient retrieval of past analyses and supports the reproducibility and traceability of the sentiment monitoring process.

It is important to note that the proposed system should be interpreted as a research-oriented prototype designed to evaluate system-level integration strategies rather than a fully optimized production platform. While it supports real-world data acquisition and analysis workflows, aspects such as large-scale deployment and long-term operational robustness are beyond the scope of the current implementation.

B. Pipeline Phases

The processing pipeline is executed in sequential phases in order to separate extraction, interpretation, and aggregation tasks. These phases are reflected in job status updates to support real-time monitoring of the system. The execution state is tracked in MongoDB through the `job_status` collection, which stores per-platform progress information such as *current*, *total*, and *status*. In addition, the progress of the LLM interpretation stage is tracked through the field `llm_status`.

The first phase corresponds to the extraction stage. The orchestrator launches a multiprocessing pool with one process per platform, where each process runs a Playwright scraper. These scrapers collect posts and comments related to the target topic and return normalized records containing the platform identifier, the post content, and the associated comment content.

Once the content from all platforms has been collected, the pipeline enters the interpretation and sentiment labelling phase. After consolidation into a DataFrame, the system performs concurrent API calls to the large language model using a thread pool in order to maximize throughput during the I/O-bound inference process. Each record is processed by the model and mapped to a fixed set of sentiment labels: *Positive*, *Neutral*, or *Negative*. The model also generates a short natural-language justification, which is stored in the field `explanation_llm`. Outputs that do not conform to the expected format or fail during processing are mapped to the label *Error*.

For each record, the LLM receives an instruction to assign a discrete polarity label and generate a short justification grounded only in the textual evidence present in the message. The explanation, therefore, reflects linguistic signals such as supportive language, expressions of gratitude, celebration, criticism, insults, frustration, or neutral informational tone, while avoiding assumptions beyond the text itself. In practice, the model assigns the label *Positive* when the message conveys approval, encouragement, pride, or favorable outcomes; *Negative* when it expresses disapproval, anger, accusations, mockery, or hostile language; and *Neutral* when the content is primarily descriptive, interrogative, informational, or lacks clear affective cues. These explanations are later aggregated during the storytelling stage to summarize dominant patterns and cross-platform differences.

After sentiment interpretation, the pipeline proceeds to the classic NLP and term extraction phase. During this stage, the system applies text cleaning and tokenization and computes salient terms using frequency-based signals. These terms support quick corpus inspection and reporting and are exported together with the sentiment outputs in the required CSV structure.

The following phase corresponds to aggregation and storytelling. In this stage, the system computes global and per-

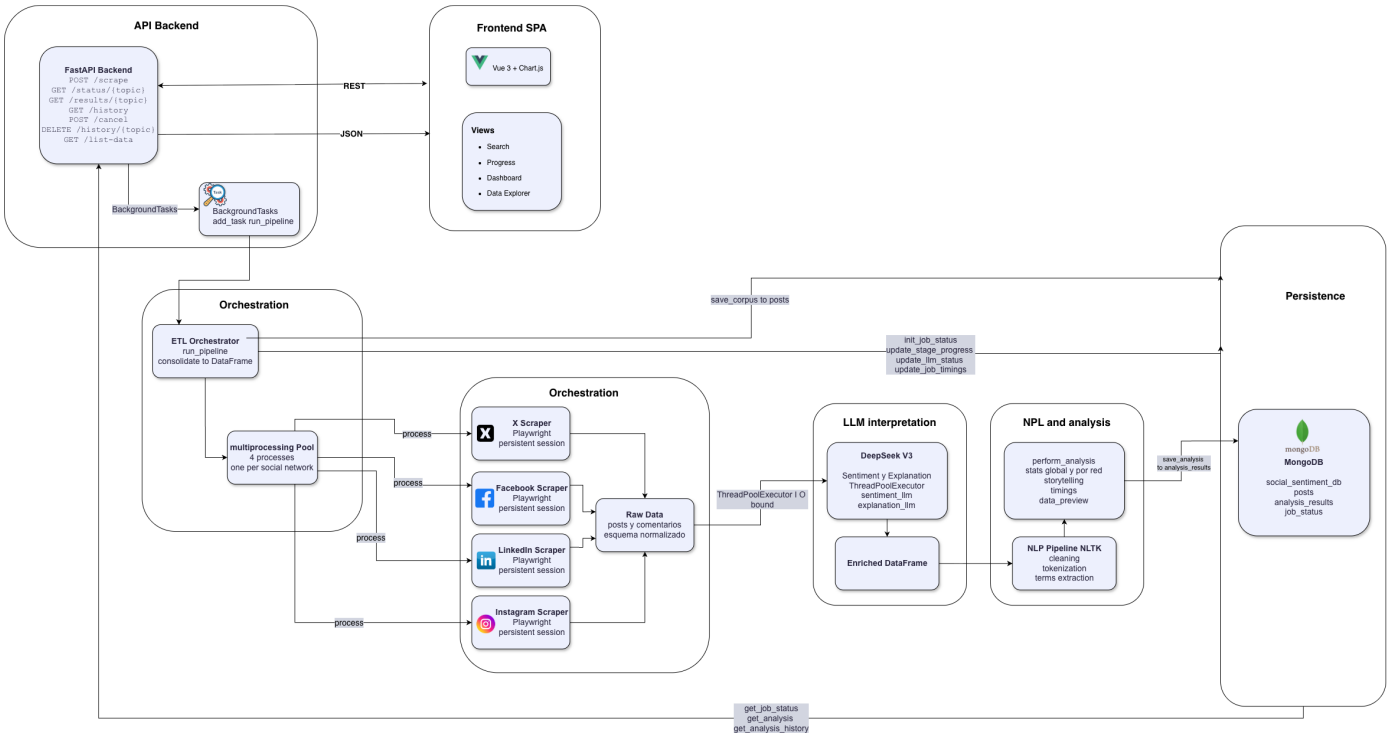

Fig. 1. Layered architecture of SentimentPulse with REST workflow, background execution, parallel scraping, LLM interpretation, NLP and analysis, and database persistence.

platform sentiment counts and percentages, selects representative examples from the dataset, and generates a narrative report intended for executive interpretation. This storytelling component summarizes dominant sentiment patterns and highlights differences across social networks.

Finally, the persistence and export phase stores the results of the analysis. The collected corpus is stored in the `posts` collection, job tracking information is updated in `job_status`, and aggregated outputs are stored in `analysis_results`. Additionally, the system exports a CSV deliverable with the required columns: *Platform*, *Post*, *Comment*, *Sentiment*, *Explication*, and *Terms*.

To mitigate the inherent variability of LLM outputs, the system incorporates several design decisions at the interpretation stage. First, model responses are constrained to a fixed label set (*Positive*, *Neutral*, *Negative*) through prompt standardization and post-processing validation. Second, non-conforming or ambiguous outputs are explicitly mapped to an *Error* category to prevent uncontrolled propagation of inconsistent predictions. Third, the system records the status of the LLM inference phase (`llm_status`) to enable traceability and monitoring of potential inconsistencies during execution.

These mechanisms allow the pipeline to maintain consistent downstream processing despite variability in LLM responses, supporting more stable and auditable sentiment analysis results.

C. Concurrency Strategy and Operational Considerations

The pipeline combines different concurrency mechanisms depending on the characteristics of each stage. This hybrid strategy improves overall system throughput while maintaining isolation between platform-specific operations.

During the extraction phase, scraping tasks are executed using multiprocessing. A separate process is created for each platform worker, allowing the scrapers to run independently and preventing failures or delays in one platform from affecting the others. This approach also improves end-to-end latency because multiple platforms can be processed simultaneously.

In contrast, the interpretation phase relies on concurrent LLM inference implemented with a thread pool. Since the interaction with the language model involves network-bound API calls, a multithreading approach increases throughput by allowing multiple requests to be issued while waiting for responses. This design maximizes resource utilization during the I/O-bound inference stage.

From an operational perspective, social media platforms often rely on dynamic content rendering mechanisms such as infinite scrolling, asynchronous loading, and frequent user interface changes. To maintain robustness under these conditions, the scrapers incorporate controlled scrolling strategies, retry mechanisms, and persistent sessions. Failures or interruptions during extraction are tracked through explicit status fields stored in the system database, allowing post hoc auditing and inspection of scraping runs. Finally, from a data protection perspective, the system is designed to process only publicly available textual content from social media posts and comments. No personally identifiable information (PII) is intentionally extracted, stored, or analyzed by the pipeline. The collected data are limited to the textual content required for sentiment analysis and associated platform metadata necessary for processing.

D. Scalability Considerations

The scalability of the proposed system is primarily determined by the concurrency model adopted for data extraction and LLM-based inference. Scraping tasks are parallelized using multiprocessing, with one independent process per social media platform. As a result, the system scales linearly with the number of platforms, provided that sufficient computational resources (CPU cores and memory) are available.

However, scalability is constrained by several factors. First, the scraping stage is dependent on dynamic web content and platform-specific access mechanisms, which introduce variability in response times and limit the degree of parallelism that can be effectively achieved. Second, LLM-based sentiment classification relies on external API calls, making throughput sensitive to network latency and service rate limits. Third, the overall system performance is bounded by the slowest stage in the pipeline, which in practice corresponds to the data extraction phase.

Empirical results presented in Section IV confirm that scraping constitutes the primary bottleneck, accounting for the majority of the total execution time. These observations indicate that, while the system benefits from concurrent execution, its scalability is inherently constrained by external dependencies and I/O-bound operations rather than computational limitations alone.

E. Algorithmic Summary

Algorithm 1 describes the end-to-end workflow executed after the user submits a topic. The input is a user-defined `topic` and a per-platform `limit`. The pipeline outputs (1) a structured corpus stored in MongoDB (`posts`), (2) an aggregated analysis document with metrics and storytelling (`analysis_results`), and (3) a CSV export with the required columns for the deliverable.

Step 2 to Step 4 perform parallel data acquisition across platforms and consolidate the collected posts and comments into a unified schema. Step 5 to Step 6 execute concurrent LLM inference to assign sentiment labels and explanations. Step 7 to Step 8 extract salient terms and compute global and per-platform statistics, which are then used to generate executive storytelling. Finally, the pipeline persists the corpus and analytics to MongoDB and exports the CSV deliverable.

Algorithm 1 SentimentPulse Multi Platform Sentiment Analysis Pipeline

```
procedure SentimentPulsePipeline(topic, limit)
    init_job_status(topic)
    Phase 1: Parallel scraping
    records ← ∅
    for all platform ∈ {X, Facebook, LinkedIn, Instagram} do in parallel
        platformRecords ← scrape_platform(topic, limit)
        records ← records ∪ platformRecords
    end for
    update_llm_status(topic, running)
    Phase 2: LLM interpretation and sentiment labelling
    results ← ∅
    for all record ∈ records do using thread pool
        label, explanation ← llm_classify(record.text)
        results ← results ∪ (record, label, explanation)
    end for
    update_llm_status(topic, completed)
    Phase 3: NLP term extraction
    terms ← extract_salient_terms(results)
    Phase 4: Aggregation and storytelling
    metrics ← compute_sentiment_metrics(results)
    storytelling ← generate_executive_storytelling(metrics, results)
    Phase 5: Persistence and export
    save_corpus(topic, results)
    save_analysis(topic, metrics, storytelling)
    export_csv(topic, results, terms)
end procedure
```

IV. Results

A. Experimental Setup

The experimental evaluation is based on two anonymized execution runs generated by the SentimentPulse system. The resulting data are stored in CSV files that capture both the sentiment-labeled corpus and the execution-time measurements produced during the analysis pipeline. Specifically, the evaluation considers (1) a sentiment-labeled dataset corresponding to *Case Study A*, and (2) an execution-time log corresponding to *Case Study B*.

Case Study A corresponds to a high-visibility sports topic involving a male professional football player associated with the Ecuadorian national team. Such topics typically generate emotionally charged reactions across social media platforms, including expressions of support, celebration, criticism, and rivalry among users.

Case Study B corresponds to a high-salience political topic involving a prominent political figure in the United States. Political discussions of this type frequently produce polarized discourse and rapid shifts in tone across different platforms.

To comply with ethical and confidentiality considerations, all datasets were anonymised before analysis. Consequently, specific person names, entities, or identifying topic strings are not disclosed in this manuscript.

The collected social media content may include multiple languages as well as informal code-switching. The LLM processes the raw text without language filtering and normalizes all outputs into a fixed sentiment label set consisting of *Positive*, *Neutral*, and *Negative*. Any responses that do not conform to the expected format are mapped to the label *Error*.

B. Case Study A: Dataset Statistics and Sentiment Distribution

Table II summarizes the dataset collected for *Case Study A*. In total, 1,032 records (comments/interactions) were processed across four social media platforms—X, Facebook, LinkedIn, and Instagram—corresponding to 107 unique posts. The largest share of records originated from X (433) and Instagram (321), followed by Facebook (244) and LinkedIn (35). One record contained missing platform metadata and is reported as *Unknown* for completeness.

TABLE II. DATASET SIZE BY PLATFORM FOR *CASE STUDY A*

Platform	Records	Unique posts
X	433	49
Instagram	321	32
Facebook	244	10
LinkedIn	35	15
Unknown	1	1
Total	1032	107

The overall sentiment distribution is reported in Table III. Positive sentiment represents the largest share of the dataset (49.0%), followed by negative sentiment (36.6%) and neutral sentiment (14.2%). A single instance (0.1%) was labelled as *Error*, indicating an unclassified output. Such cases may occur due to transient failures or unexpected responses during automated processing, but their negligible frequency suggests that the pipeline maintained stable inference behaviour.

TABLE III. OVERALL SENTIMENT DISTRIBUTION FOR *CASE STUDY A*

Sentiment	Count	Percentage
Positive	506	49.0%
Negative	378	36.6%
Neutral	147	14.2%
Error	1	0.1%
Total	1032	100%

Table IV provides a platform-level breakdown of sentiment labels. The distribution reveals clear cross-platform differences. X concentrates the largest share of negative reactions (221 instances), reflecting the platform's tendency to host more confrontational or critical discourse in this topic. In contrast, Instagram exhibits a predominantly positive distribution (197 positive versus 89 negative), which is consistent with the more supportive or celebratory tone often observed in visual-centric social media interactions. Facebook presents a more balanced distribution, while LinkedIn contains a relatively small number of records, mostly positive or neutral.

TABLE IV. SENTIMENT BY PLATFORM FOR *CASE STUDY A*

Platform	Positive	Neutral	Negative	Error
X	148	61	221	1
Instagram	197	35	89	0
Facebook	132	44	68	0
LinkedIn	28	7	0	0
Unknown	1	0	0	0

Beyond the numerical distribution, the explanations generated by the LLM provide insight into the linguistic drivers behind the assigned polarity labels. Positive classifications were typically justified by supportive or celebratory expressions such as praise, encouragement, or gratitude. Negative classifications frequently corresponded to critical or hostile language, including insults, accusations, or expressions of frustration. Neutral classifications were generally associated with informational statements, factual descriptions, or questions where explicit affective cues were absent.

These qualitative explanations complement the quantitative analysis by clarifying *why* certain platforms concentrate higher levels of negativity or positivity. In particular, they show that the model's decisions are grounded in explicit textual cues rather than opaque or arbitrary outputs, supporting the interpretability of the system's sentiment classification process.

C. Case Study B: Runtime and Performance

Runtime performance was evaluated using the execution logs generated for *Case Study B* (timestamp: 2026-02-05 23:32:21, limit: 10 posts per platform). In this run, the system processed a total of 1121 records (comments/interactions) collected across the monitored platforms.

Table V summarizes the stage-level execution times recorded in the CSV log. The parallel scraping stage required 678.33 s, while the LLM-based sentiment classification stage required 92.77 s. Considering both stages together, the measured processing time for data acquisition and inference was 771.10 s.

TABLE V. PIPELINE PERFORMANCE FOR *CASE STUDY B* (LIMIT: 10 POSTS PER PLATFORM).

Stage	Time (s)	Proportion
Parallel scraping (4 processes)	678.33	88.0%
LLM classification (ThreadPool)	92.77	12.0%
Measured total (Scraping + LLM)	771.10	100%

Based on these measurements, the LLM inference stage achieved an approximate throughput of 12.08 processed items per second (1121 records divided by 92.77 seconds). The results indicate that the dominant contribution to overall runtime originates from the scraping stage, which represents approximately 88.0% of the measured execution time. In contrast, the LLM inference stage accounts for 12.0% of the total runtime.

This distribution reflects the operational characteristics of social media data acquisition, where dynamic web interfaces, pagination, and interaction-driven loading introduce higher latency compared to API-based model inference. Nevertheless, the concurrent execution strategy—multiprocessing for scraping and multi-threading for LLM inference—allows the system to process more than one thousand social interactions within approximately thirteen minutes in this configuration.

Although the reported execution times correspond to a single experimental run, variability in system performance is expected due to factors such as network latency, platform response times, and external API dependencies. In particular, the scraping stage is subject to dynamic content loading and platform-specific constraints, which may introduce fluctuations in execution time across runs.

D. Storytelling Outputs by Case Study

In addition to quantitative metrics, SentimentPulse generates an executive-oriented *storytelling* report for each run. This report aggregates platform-level sentiment distributions and uses the per-item `explanation_llm` field to interpret *why* polarity patterns emerge. The goal is to transform raw sentiment counts into an interpretable narrative that highlights dominant themes and cross-platform contrasts.

a) Case Study A (sports topic): For *Case Study A*, the storytelling report describes the conversation as predominantly positive, consistent with the distribution presented in Section IV-B. Supportive and celebratory messages constitute the main drivers of positive sentiment, whereas negative sentiment is largely associated with criticism and expressions of frustration. The narrative also emphasizes that polarity in this domain is frequently conveyed through short, emotionally charged statements. Differences across platforms are visible in the tone of engagement: some networks concentrate stronger criticism, while others amplify supportive reactions. These observations are grounded in the LLM explanations, which typically reference praise, encouragement, or gratitude for positive labels and derogatory or accusatory language for negative labels.

b) Case Study B (political topic): For *Case Study B*, the storytelling report characterizes the discussion as highly polarized, with frequent alternation between supportive and hostile language. Compared with the sports topic, the discourse exhibits stronger evaluative framing, including allegations, confrontational phrasing, and explicit ideological positioning. Platform-level patterns are also more pronounced in this case, with certain networks displaying more adversarial exchanges while others contain comparatively neutral or informational contributions. The LLM-generated explanations support these observations by identifying cues such as mockery, accusations, or hostile framing for negative sentiment, and endorsement or approval for positive sentiment.

Overall, the storytelling layer complements the quantitative analysis by linking aggregate sentiment proportions to recurring linguistic cues extracted from the `explanation_llm` field. This additional interpretive layer facilitates faster understanding of the results and provides an auditable explanation of cross-platform sentiment dynamics.

E. Human Validation of Sentiment Classification

To obtain a more reliable estimate of sentiment classification quality, a manual validation was conducted on a random sample of 100 comments drawn from the processed dataset. Each comment was independently labelled by a human evaluator using the same sentiment categories employed by the system (*Positive*, *Neutral*, *Negative*). The predicted labels produced by the LLM were then compared against the human annotations.

The model matched the human label in 88% of the evaluated cases (Accuracy = 0.88), indicating a strong level of agreement under the sampled conditions. Table VI reports per-class precision, recall, and F1-score values. The evaluated sample contains instances from all three sentiment classes, allowing for a more comprehensive assessment of model performance across categories.

The confusion matrix in Table VII shows that most disagreements correspond to polarity shifts between *Positive* and *Negative*, which is consistent with known challenges in sentiment analysis when dealing with ambiguous or mixed expressions. A smaller number of errors involve confusion between *Neutral* and the other classes, suggesting that borderline cases with weak affective signals remain difficult to classify consistently.

TABLE VI. MANUAL VALIDATION METRICS (SAMPLE: 100 COMMENTS)

Class	Precision	Recall	F1-score	Support
Positive	0.97	0.92	0.94	36
Negative	0.79	0.87	0.83	30
Neutral	0.88	0.85	0.87	34
Macro avg	0.88	0.88	0.88	100
Weighted avg	0.88	0.88	0.88	100

TABLE VII. CONFUSION MATRIX (GROUND TRUTH VS. PREDICTED) FOR MANUAL VALIDATION (N=100).

	Predicted (LLM)		
Ground truth (human)	Negative	Neutral	Positive
Negative	26	3	1
Neutral	5	29	0
Positive	2	1	33

Beyond the quantitative agreement, the presence of the `explanation_llm` field enables post-hoc inspection of mismatches. When the predicted label differs from the human label, the associated justification can be examined to determine whether the discrepancy arises from linguistic ambiguity (e.g., sarcasm, mixed sentiment, or implicit tone) or from model misinterpretation. This capability enhances interpretability and provides a practical mechanism for refining prompt design and evaluation strategies.

It is important to note that this validation is based on a single annotated sample and one human evaluator. Therefore, the reported metrics should be interpreted as an indicative assessment of model behaviour rather than a definitive generalization of performance across all possible inputs. Expanding the evaluation to larger and multi-annotator datasets constitutes an important direction for future work.

F. Baseline Comparison

To contextualize the performance of the proposed approach, we consider commonly used sentiment analysis baselines such as VADER and TextBlob, which are widely adopted in social media analysis tasks.

These methods have been widely used in sentiment analysis tasks and have shown competitive performance in detecting explicit sentiment polarity, particularly in short and informal texts [19], [20]. However, prior studies have reported limitations when handling contextual nuances, implicit tone, sarcasm, and mixed sentiment expressions, which are common in social media data [21], [22].

In contrast, the LLM-based approach used in SentimentPulse is designed to capture richer contextual information and provide explanatory outputs, which can improve interpretability and robustness in complex linguistic scenarios. While a direct experimental comparison is left for future work, the results obtained in the manual validation suggest that the proposed approach achieves a high level of agreement with human annotations.

V. CONCLUSION

This work presented SentimentPulse, a unified web application that integrates concurrent multi-platform data extrac-

tion, LLM-based sentiment classification with per-item explanations, and executive-oriented storytelling for operational social listening. The proposed architecture combines parallel scraping, concurrent LLM inference, and structured persistence to support end-to-end analysis of social media conversations across multiple platforms within a single workflow.

Across the reported anonymized case studies, the results show that the system can process large volumes of social media interactions while producing interpretable outputs suitable for cross-platform comparison. In the sports-related topic analyzed in Case Study A, the pipeline processed over one thousand comments across four platforms and revealed a predominantly positive conversation, with substantial variations in sentiment distribution between networks. These differences highlight the importance of analyzing sentiment both globally and at the platform level. The inclusion of the `explanation_llm` field further improves auditability by enabling sentiment labels to be inspected together with short text-grounded justifications.

The runtime analysis in Case Study B shows that the end-to-end pipeline can process more than one thousand interactions in approximately thirteen minutes under the evaluated configuration. The measurements indicate that data extraction remains the dominant cost in the pipeline, accounting for the majority of the execution time, whereas LLM-based inference represents a smaller share of the runtime. A small-scale manual validation achieved an accuracy of 0.85, with strong F1-scores for positive and negative classes, suggesting that the model can reliably detect polarity when sentiment cues are explicit.

Despite these results, several limitations remain. Scraping reliability depends on platform interface changes and access constraints, which may affect both the completeness and latency of data acquisition. Additionally, LLM outputs may vary with prompt formulation and domain shifts, making continuous monitoring and evaluation necessary for operational deployments.

Beyond these limitations, this work highlights the importance of system-level integration in modern sentiment analysis applications. While recent advances in LLMs have significantly improved classification capabilities, their effective use in real-world scenarios depends on how they are embedded within scalable, observable, and interpretable pipelines. In this context, SentimentPulse demonstrates that combining concurrent data acquisition, structured processing, and LLM-based interpretation can support practical social listening tasks under realistic constraints.

Future work should focus on improving extraction efficiency through techniques such as incremental collection and caching, as well as expanding the human validation protocol with larger and more balanced samples to obtain more stable performance estimates across all sentiment classes. These improvements would further strengthen the robustness and scalability of the proposed system for real-world social listening scenarios.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, pp. 1–135, 01 2008.

[2] B. Liu, *Sentiment Analysis and Opinion Mining*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[3] D. A. Andrade-Segarra, G. A. Le *et al.*, "Deep learning-based natural language processing methods comparison for presumptive detection of cyberbullying in social networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[4] G. A. León-Paredes, L. A. Alba-Narváez, S. M. Torres-Cordero, and C. M. Buestan-Villa, "A multilingual sentiment analysis system for tiktok comments in spanish using roberta and lstm," in *International Conference on Information Technology & Systems*. Springer, 2025, pp. 210–219.

[5] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, Mar. 2011.

[6] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010, pp. 178–185, conference held in Washington, DC, United States (May 23–26, 2010).

[7] G. A. León-Paredes, B. L. Padilla-Viñanzaca, S. F. Guamán-Torres, and K. M. Parraga-Riera, "A real-time political sentiment analysis system using deepseek-r1 on multiplatform social media data," in *International Conference on Communication and Applied Technologies*. Springer, 2025, pp. 33–42.

[8] B. Batrinca and P. C. Treleaven, "Social media analytics: a survey of techniques, tools and platforms," *AI & SOCIETY*, vol. 30, pp. 89–116, 2015, published online: 26 July 2014.

[9] A. Diwali, K. Saeedi, K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, "Sentiment analysis meets explainable artificial intelligence: A survey on explainable sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 15, no. 3, pp. 837–846, 2023.

[10] L. Ott and P. Theunissen, "Reputations at risk: Engagement during social media crises," *Public Relations Review*, vol. 41, no. 1, pp. 97–102, Mar. 2015.

[11] R. T. Rust, W. Rand, M.-H. Huang, A. T. Stephen, and T. Chabuk, "Real-time brand reputation tracking using social media," *Journal of Marketing*, vol. 85, no. 4, pp. 21–43, 2021.

[12] W. Zhang, Y. Deng, B. Liu, S. Pan, and L. Bing, "Sentiment analysis in the era of large language models: A reality check," in *Findings of the Association for Computational Linguistics: NAACL 2024*, K. Duh, H. Gomez, and S. Bethard, Eds. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 3881–3906. [Online]. Available: https://aclanthology.org/2024.findings-naacl.246/

[13] X. Gu, X. Chen, P. Lu, and Z. Li, "Agcvt-prompt: An automatic general chain-of-thought verbalizer template prompt learning method for sentiment classification," *Engineering Applications of Artificial Intelligence*, vol. 135, p. 107907, 2024.

[14] B. E. Parra-Zambrano and G. A. León-Paredes, "A Web Approach for the Extraction, Analysis, and Visualization of Sentiments in Social Networks Regarding the Public Opinion on Politicians in Ecuador Using Natural Language Processing and High-Performance Computing Tools," in *Information Technology and Systems*, Á. Rocha, C. Ferrás, J. Hochstetter Diez, and M. Diéguez Rebolledo, Eds. Cham: Springer Nature Switzerland, 2024, pp. 457–466.

[15] T. Ouyang *et al.*, "Stability analysis of chatgpt-based sentiment analysis," *Electronics*, vol. 13, no. 24, p. 5043, 2024.

[16] J. Y. M. Nip and B. Berthelier, "Social media sentiment analysis," *Encyclopedia*, vol. 4, no. 4, pp. 1590–1598, 2024.

[17] J. Katalinić and I. Dunđer, "Neural network-based sentiment analysis and anomaly detection in crisis-related tweets," *Electronics*, vol. 14, no. 11, p. 2273, 2025.

[18] N. A. Sharma *et al.*, "A systematic review of sentiment analysis: Tasks, applications, and deep learning techniques," *International Journal of Data Science and Analytics*, vol. 19, no. 3, pp. 351–388, 2025.

[19] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 2014.

[20] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[21] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, "Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," *Proceedings of AAAI*, 2017.

[22] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, pp. 5731–5780, 2022.