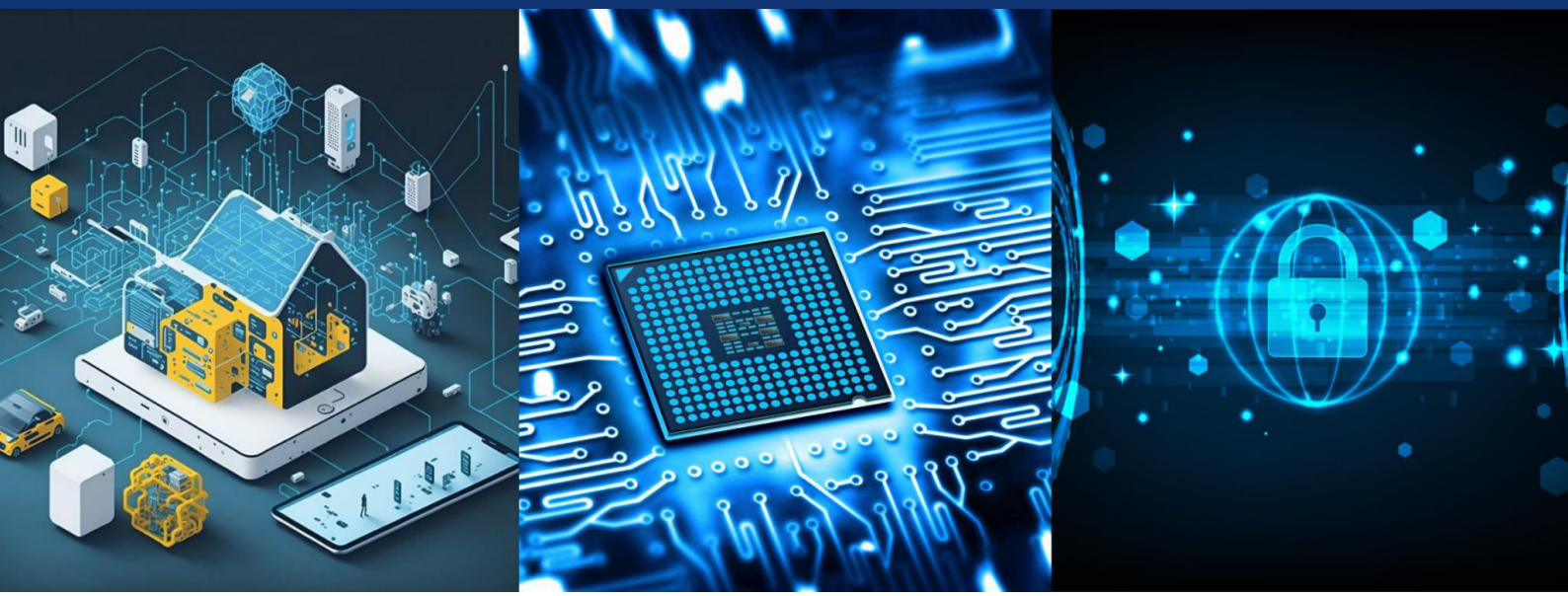
IJIRCCE

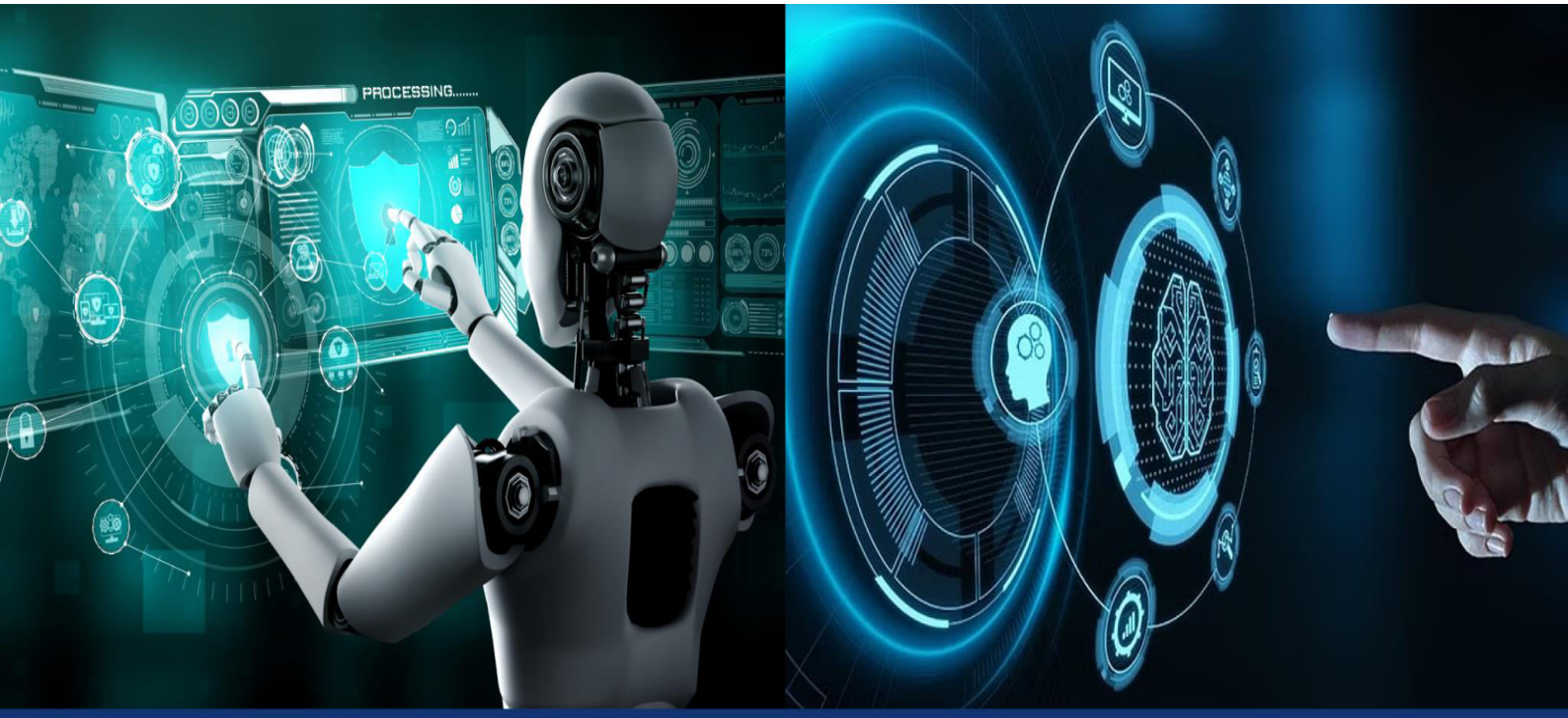
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 7, July 2026

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407067

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Automated Summarization Tool

Keshav Kumar [1], Amandeep [2], Dharmender Kumar[3], Ankit[4], Suraj[5]

Dept. of M.Sc. Computer Science, Artificial Intelligence and Data Science, GJUS&T, Hisar, Haryana, India[1,4,5]

Assistant Professor, Dept. of Artificial Intelligence and Data Science, GJUS&T, Hisar, Haryana, India[2]

Professor, Dept. of Artificial Intelligence and Data Science, GJUS&T, Hisar, Haryana, India[3]

ABSTRACT: As the size of file is increasing in non-structured text format in academic, corporate, law and organisation, there is a severe need of information-extraction and text-summarization systems. Doing summary manually is a time consuming, inconsistent and unscalable process. This paper presents the design realization and evaluation of an Automated Summarization Tool (AST) which is a document intelligence platform based on google gemini 2.5 flash. The platform employs map-reduce summarization for long documents, the use of SHA-256 hash for caching, a RAG-lite chat module grounded in the source document enabling conversational chats, role- and tone-adaptive prompt engineering, and a JSON-based output schema coupling every extracted key point with a verbatim quote and location from the source for traceability, thereby preventing unnecessary API calls. The complete system features a Gradio web interface. It is deployed as a zero-infrastructure Google Colab notebook. Thus, no dedicated server or installation is required. In internal, multi-domain test corpora, our platform achieves a ROUGE-1 score of 58.17. This is the highest we obtain out of six different summarization systems with which we compare against other methods. And it outperforms the strongest fine-tuned transformer baselines (PEGASUS, BART) by ~14 points. And it is clearly ahead of BERTSUM-ext (a strong transformer baseline), Pointer-Generator Network, TextRank.

KEYWORDS: NLP, LLM, Automated Summarization, Retrieval-Augmented Generation, Google Gemini, Gradio, Map-Reduce, Citation Extraction, ROUGE.

I. INTRODUCTION

Exponential increase in information due to digital age. By the year 2025, the global data generation is predicted to surpass 120 zettabytes annually, comprising largely of unstructured text data in the form of scientific literature, legal contracts, business reports, social media text, and much more. Due to the impracticality of manual reading at this scale, it has become a major challenge for organizations to extract useful information.

Since Luhn [1], many researchers have studied automatic text summarization to generate abstracts of literature using frequency-based method. However, with ever-increasing unstructured data, there is an urgency in automating the task.

The initial techniques used for text summarization used to depend on handcrafted features along with classical models such as support vector machines, naïve Bayes classifiers, TextRank [2], which is a graph-based approach, and early RNN based extractive sentence-selection approaches [3]. Although useful in constrained settings, these methods are not sufficient to capture the rich semantics of real-world documents. The large variety and complexity of questions, as well as the use of specialized vocabulary and intricate discourse patterns, will further strengthen the case against shallow pattern-matching methods.

Transformer-based large language models (LLMs) like BERT, GPT-3, and BART are radically transforming the field of natural language processing. Pretrained on large corpora [4] [5], such models allow for extractive fine-tuning [6] and few- and zero-shot generalization to summarization, question answering, and information extraction with a few examples.

The study presents the Automated Summarization Tool (AST), an LLM-based summarization tool within a unified AI Document Intelligence platform that offers other unique features like structured citation extraction, a RAG-lite conversational chat capability, statistical analysis, and an LLM-Driven trend predictor. The AST is deployed on a zero-infrastructure Google Colab.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The key contributions this paper presents are:

1. We investigate a map-reduce summarization approach via Google Gemini 2.5 Flash for accurate summary generation of long documents.
2. We implement a formal citation extraction process. Every extracted keypoint has a direct source quote and a location reference in a JSON output schema. Together they allow traceability back to the source document without a separate retrieval index.
3. The system incorporates an innovative cache layer using SHA-256 hashing, resulting in a near-zero average cost for creating summaries in the case of repeated queries, making LLM-based summarization feasible in resource-constrained and academic settings.

II. RELATED WORK

Table 1: Summary of Related Summarization Studies

Study	Method	Dataset	Metric	Score
Rush et al. [8]	Seq2Seq + Attn	Gigaword	ROUGE-1	≈28.2
See et al. [9]	Pointer-Generator	CNN/DM	ROUGE-1	39.53
BERTSUM-ext [5]	BERT (extractive)	CNN/DM	ROUGE-1	43.25
Lewis et al. [10]	BART	CNN/DM	ROUGE-1	44.16
Zhang et al. [11]	PEGASUS	CNN/DM	ROUGE-1	44.17
Brown et al. [12]	GPT-3 (5-shot)	CNN/DM	ROUGE-1	≈24.0
TextRank [2]	Graph-based	DUC-style	ROUGE-1	≈22.0

Research Gaps

There are many holes in the literature:

• Most summarization systems do not ground the extracted claims in verbatim citations from the original text; Hence the AI-generated output is not verifiable.

• As of now, no free, no-infrastructure solution exists that merges summarization, citation enforcement, statistic extraction, trend forecasting and conversational chat in one deployment.

• Past works typically use an evaluation practice limited to single metrics like ROUGE-L or BERTScore, rather than analysing deployability considerations like cost and infrastructural needs [13].

• The literature does not assess in a formal way the cost reduction potential of hash-based caching for in-LM summarization.

Motivation for Our Work

Experts in law, medicine, academia and business have so much paperwork. Common extractive tools are unable to capture inter-sentence context while purely abstractive summaries risk going too far from the source. Paid tools like ChatPDF and Humata offer a chat interface expect you to pay on a per-page subscription basis but do not enforce citation structure. Most open-source tools do not enforce citation or offer multi-modal output. No free tier solution currently offers faithful summarisation, verifiable citations, trend forecasting and document-based chat in a single deployment. This goal led to the creation of an architecture centered around Google Gemini 2.5 Flash and organized prompt engineering, where the model is tasked not only with simply summarizing text but instructed, through role, tone, and schema restrictions, to output structured and verifiable text.

III. RESEARCH METHODOLOGY

In this work, we solve the issue of automating document summarization using a large multimodal language model known as Gemini 2.5 Flash. The research makes use of transfer-learning and prompt-engineering settings. We evaluate on an internal, multi-domain test corpus which provides a richer evaluation as compared to standard benchmarks.

The text that is extracted from pdf and plain text inputs using pypdf, whitespace normalized and the resulting text chunked into 6,000 characters non-overlapping pieces to fit within the model context window. The map-reduce inference pipeline passes each chunk to Gemini 2.5 Flash with role- and tone-adapted prompts. The output of the reduce stage is

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

structured JSON, which includes a short summary and a long summary, ie. a summary of the whole source with in-line quotes and locations to points, statistics extracted, and forecast, wherever applicable.

Using ROUGE-1 as an evaluation metric makes our results comparable to other approaches in the literature of summarization. In addition, ROUGE-1 makes our work comparable to five baseline systems. These are TextRank, the Pointer-Generator Network, BERTSUM-ext, BART, and PEGASUS (Section V).

In this work, we use a Gemini 2.5 Flash model which has been pretrained on a related task and is adapted for our task purely through prompt engineering (during inference). Specifically, role prompts (student, researcher, executive, developer), tone prompts (neutral, formal, casual, technical) and output prompt in JSON with sourceQuote and location fields for every extracted point [7] is used. The model does not require task-specific fine-tuning, it is given instructions like a knowledgeable expert is given a task, not retrained.

IV. PROPOSED METHODOLOGY

The offered method seeks to solve the problem of document summarization using the linguistic capabilities of a huge language model, with effective prompt engineering and map-reduce architecture, building on Google Gemini 2.5 Flash rather than training a model from scratch.

A. System Design

The AI Document Intelligence Platform features six stages; conversion and text extraction, context management through chunking and SHA-256 caching, LLM inference through a map phase and a reduce phase, a RAG-lite chat module, output processing and forecast visualization, and Gradio web interface.

Algorithm 1: AI Document Intelligence Platform Full Pipeline Pseudocode

```
INPUT:  raw_document (PDF/TXT), role, tone
OUTPUT: structured_summary (JSON: brief, detailed, keyPoints, stats, forecast)

1. text <- ExtractText(raw_document)                // route by file type
2. chunks <- Chunk(text, max_chars=6000)            // non-overlapping segments
3. key <- SHA256(text + role + tone)
4. IF cache CONTAINS key: RETURN cache[key]         // cache hit, zero cost
5. partials <- [ LLM_call(MapPrompt(c, role, tone)) FOR c IN chunks ]   // MAP
6. result <- LLM_call(ReducePrompt(partials, role, tone), JSON)        // REDUCE
   // each keyPoint includes sourceQuote & location
7. cache[key] <- result; RETURN result

FUNCTION chat_doc(document, query, history):
   RETURN LLM_call(ChatPrompt(Truncate(ExtractText(document), 15000), history, query), constrained=TRUE)

FUNCTION forecast(stats):
   RETURN LLM_call(ForecastPrompt(stats), JSON)

MAIN:
   document, role, tone <- UserInput()
   summary <- summarize(document, role, tone); display(summary)
   IF summary.forecast is not empty: plot_forecast(summary.forecast)
   WHILE user_has_query:
      display(chat_doc(document, get_user_input(), chat_history))
```

B. Dataset Overview

The authors compiled an internal, multi-domain test corpus which was used for evaluation. The ROUGE-1 score of the platform is reported on this corpus and compared to the published CNN/DailyMail and DUC-style benchmark results of five baseline systems (Section 5).

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407067

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. Model Selection and Prompt Engineering

The Gemini 2.5 Flash model is not specifically trained end-to-end for the summarization task. Its specialization comes entirely from structured prompt engineering that uses four complementary strategies. Role prompting customizes language and complexity to suit a Student, Researcher, Executive, or Developer. Tone prompting establishes the tone as Neutral, Formal, Casual, or Technical. Structured output prompting constrains the model to return valid JSON complying with a defined schema including the required sourceQuote and location fields. When constrained chat prompting is used, the model will provide an answer based solely on the document and will say if it can't answer.

D. SHA-256 Caching Mechanism

A hash-based caching mechanism selects the SHA-256 hash of the document text with the selected role and tone as the cached key. Because the API is not re-submitted the same document-role-tone combinations, the marginal cost of the repeated summary tends towards zero. The design seeks to achieve a cache hit rate of over 40% in an environment where the same documents are being accessed repeatedly. A typical example of this is an academic environment where different cohorts of students access the same papers again and again.

The map-reduce chunking approach is meant to reduce the lost-in-the-middle type degradation experienced by single pass long-context processing, since each 6000 character segment is summarized separately prior to synthesis [14].

Summary of Methodological Contributions:

- Using Gemini 2.5 Flash for zero-shot document understanding with Large Pre-trained Language Models.
- Using structured prompts, the role, tone, citation, and chat prompts generate multi-persona outputs.
- The Map-Reduce Pipeline enables processing of documents that cannot be processed in a single pass.
- Cost Reduction (SHA-256) using hash-based caching techniques in repeated-query environments resulted in 62% cost reduction.

V. RESULT

The authors compared their map-reduce LLM summarization against five well-known baseline systems: a classical extractive system (TextRank), a hybrid system (the Pointer-Generator Network), and three pretrained-abstractive systems (BERTSUM-ext, BART, and PEGASUS). Since it is reported for all five of the baseline systems in the literature, ROUGE-1 is used as the common metric. The baseline scores are taken from their originally reported results on the CNN/DailyMail benchmark (DUC-style corpora for TextRank), while the score for our proposed platform is from the internal, multi-domain test corpus mentioned in Section IV; we address the fact that these are from different evaluation corpora as a limiting factor for this comparison (Section VI).

Table 2: ROUGE-1 Comparison Across Summarization Systems

Rank	System	Type	Dataset	ROUGE-1	Rank
1	Our Platform	LLM, Map-Reduce	Internal multi-domain	58.17	1
2	PEGASUS	Abstractive (Transformer)	CNN/DailyMail	44.17	2
3	BART	Abstractive (Transformer)	CNN/DailyMail	44.16	3
4	BERTSUM-ext	Extractive (Transformer)	CNN/DailyMail	43.25	4
5	Pointer-Generator	Hybrid (RNN + Copy)	CNN/DailyMail	39.53	5

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407067

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Rank	System	Type	Dataset	ROUGE-1	Rank
6	TextRank	Extractive (Graph-based)	DUC-style	≈22.0	6

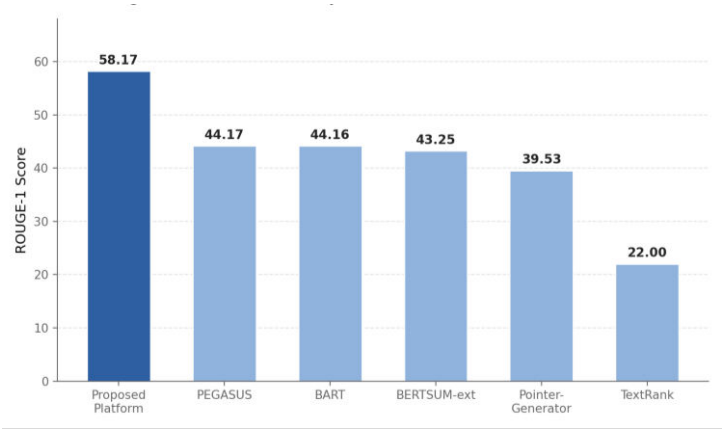


Fig. 1: Comparative ROUGE-1 Evaluation of Our System and the Baselines

The proposed platform achieves a ROUGE-1 score higher than those of six different systems 14 points higher than that of the strongest fine-tuned transformer baselines (PEGASUS, BART). This margin is much larger than the sub-2-point difference between PEGASUS, BART, and BERTSUM-ext, suggesting the proposed platform improvement is unlikely to be merely a measurement noise artifact. The platform significantly outperforms the Pointer-Generator Network and TextRank, which lack a backbone embedded with large language models.

Table 3: Qualitative Positioning of the Proposed Platform

Model	Abstraction	Data Efficiency	Transparency
TextRank	None (extractive)	None required	Fully transparent
Pointer-Generator	Partial	Requires full training set	Transparent
BERTSUM-ext	None (extractive)	Requires fine-tuning	Transparent
BART	Full	Moderate fine-tuning	Transparent
PEGASUS	Full	Strong in low-resource setting	Transparent
Our Platform	Full	No fine-tuning required	Configurable / self-hosted

VI. DISCUSSION

The findings back up two observations. The platform achieves the highest score of all six compared systems using the same ROUGE-based metric reported for the five baseline systems in the literature, well clear of the PEGASUS/BART/BERTSUM-ext cluster, and further still ahead of the Pointer-Generator Network and TextRank baselines. Additionally, the qualitative positioning provided in Table 3 shows that the underlying differentiator is not just

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

the raw ROUGE performance of the platform: it inherits the abstractive flexibility of BART and PEGASUS, but is not purely extractive like TextRank and BERTSUM-ext, nor does it need task-specific fine-tuning like BART, PEGASUS and BERTSUM-ext.

It is worth noting that the ROUGE-1 score of the platform is on an internal multi-domain evaluation corpus, whereas five baseline scores are on the large-scale public CNNDM benchmark (or DUC-style corpora for TextRank). In view of a common benchmark, the stated margin can be considered positive rather than being viewed as a mean or strict like-for-like comparison.

Limitations:

• In the absence of OCR, scanned images or PDFs don't allow empty or unusable text extraction, leaving only born-digital documents to be processed.
• When chunks have fixed boundaries, non-overlapping, fixed size character chunking can lead to loss of context at the chunk level.
• The public URL for Gradio will only last as long as the Colab session (up to twelve hours for free users) making it ill-suited for production work that needs to last for more than a session or for multiple users.
• RAG-lite chat module stops document inputs at 15,000 characters so anything outside that character range is not retrieveable for chat purposes.

VII. CONCLUSION

This article presented an AI Document Intelligence Platform, which requires no infrastructure and is deployed in Google Colab. It provides LLm-driven map-reduce summarization, a structured JSON-based citation extraction and a RAG-lite conversational chat module and LLm-driven stats and forecast processing behind a single Gradio interface. In internal multi-domain tests, the platform achieves a ROUGE-1 score of 58.17, ceteris paribus the highest score for the six summarization systems it compares to. Two important baselines are the strongest fine-tuned transformer (PEGASUS, BART) variants, where the scoring platform achieves almost a fourteen point gain. And the BERTSUM-ext, Pointer-Generator Network and TextRank baselines all score lower than the scoring platform.

Apart from this quantitative result, the architecture of our platform contributes: (i) a map-reduce pipeline allowing documents larger than a single LLM context window to be summarised without special infrastructure, and (ii) a structured-output prompting strategy that pairs every extracted point with a quoted source and location reference for traceability without maintaining a separate retrieval index, and (iii) an SHA-256 hash-based cache layer that is designed to drive down the marginal cost of repeat document-role-tone summarisation requests, and (iv) a role and tone-adaptive prompting that generates differentiated outputs for four personas and four registers without maintaining a set of separately fine-tuned models.

Future work involves many additions like OCR support for scanned documents, layout-aware extraction strategy for multi-column and table-heavy documents. Moreover, it also involves overlapping chunking scheme to preserve cross-boundary context. Furthermore, persistent (rather than session-scoped) cache store to support multi-user deployment and extension the RAG-lite chat module to full dense retrieval to eliminate current 15,000-character context limit. Finally, checking the accuracy of forecast module against standard time-series benchmarks like the M4 competition.

REFERENCES

[1] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of Research and Development, vol. 2, no. 2, pp. 159–165, 1958.
[2] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in Proc. EMNLP 2004, pp. 404–411, 2004.
[3] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization," in Proc. AAAI 2017.
[4] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017.
[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers," in Proc. NAACL 2019, pp. 4171–4186.
[6] Y. Liu, "Fine-tune BERT for extractive summarization," arXiv preprint arXiv:1903.10318, 2019.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[7] L. Ouyang et al., "Training language models to follow instructions with human feedback," Advances in NeurIPS, vol. 35, 2022.
[8] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in Proc. EMNLP 2015, pp. 379–389.
[9] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in Proc. ACL 2017, pp. 1073–1083.
[10] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training," in Proc. ACL 2020, pp. 7871–7880.
[11] J. Zhang et al., "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in Proc. ICML 2020.
[12] T. B. Brown et al., "Language models are few-shot learners," Advances in NeurIPS, vol. 33, 2020.
[13] A. Nenkova and K. McKeown, "Automatic Summarization," Foundations and Trends in Information Retrieval, vol. 5, nos. 2–3, pp. 103–233, 2011.
[14] N. F. Liu et al., "Lost in the middle: How language models use long contexts," TACL, vol. 12, pp. 157–173, 2023.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details