مجلة جامعة بني وليد للعلوم الإنسانية والتطبيقية
Bani Waleed University Journal of Humanities and Applied Sciences
تصدر عن– جامعة بني وليد – ليبيا
Website:https://jhas-bwu.com/index.php/bwjhas/index
ISSN 3005-3900 الصفحات (136 -142) المجلد الحادي عشر ــ العدد الثالث ــ 2026

Secure AI Applications for Summarizing Scientific Studies and Preserving Privacy

Izadeen Kajaman [1]*
Computer Science Department, Faculty of Information Technology, Bani Waleed University, Bani Walid, Libya
izadeenkajaman@bwu.edu.ly

تطبيقات الذكاء الاصطناعي الآمنة في تلخيص الدراسات العلمية والحفاظ على الخصوصية

عزالدين كجمان *
قسم الحاسوب، كلية تقنية المعلومات، جامعة بني وليد، بني وليد، ليبيا

Received: 16-03-2026	Accepted: 22-04-2026	Published: 27-04-2026

الملخص:
يُعدّ النمو المتسارع في الأدبيات العلمية تحدياً جوهرياً يواجه الباحثين في تحديد الدراسات ذات الصلة واستيعابها بكفاءة. وعلى الرغم من أن منهجيات التلخيص الاستخلاصي تقدّم حلاً قابلاً للتوسع لتحديد الجمل الرئيسية، فإنها كثيراً ما تفتقر إلى التماسك السردي الذي يستلزمه الخطاب الأكاديمي. وفي السياق ذاته، تُثير أساليب التلخيص التجريدي -ولا سيما تلك المعتمدة على نماذج اللغة الكبيرة (MsLL) السحابية- مخاوف جدية تتعلق بالتكلفة، وحماية خصوصية البيانات، والاعتماد على الخدمات الخارجية.
تُقدِّم هذه الورقة نظام "Abstractive Muse"، وهو إطار عمل مبتكر ذاتي الاحتواء يحافظ على الخصوصية، صُمِّم خصيصاً لمعالجة تحديات تلخيص الأدبيات العلمية. يُقدَّم النظام في هيئة تطبيق سطح مكتب مستقل يجمع بين خوارزمية **(TextRank)** الكلاسيكية القائمة على الرسوم البيانية للتلخيص الاستخلاصي الأولي، ونموذج لغوي كبير مفتوح المصدر يعمل محلياً لتحقيق التوليف التجريدي.
يستعرض هذا العمل دورة التطوير الكاملة بالتفصيل، شاملاً التصميم المعماري، وتنفيذ واجهة مستخدم رسومية تتمحور حول احتياجاته، والقرارات الهندسية التي تكفل متانة النظام واستقلاليته عن واجهات برمجة التطبيقات الخارجية. تُخوِّل الأداة النهائيةُ المستخدمَ اختيارَ مستندات **(PDF)** محلية، وتحديد معايير التلخيص، وتوليد ملخصات استخلاصية دقيقة وسرد أكاديمي عالي الجودة مدعوم بالذكاء الاصطناعي، كل ذلك دون أن تغادر بياناته جهازه الخاص. يُقدِّم هذا العمل نموذجاً قابلاً للتكرار لبناء أدوات عملية وآمنة وميسورة في مجال معالجة اللغات الطبيعية، ويؤكد في الوقت ذاته القيمةَ الجوهرية لمبدأ "الذكاء الاصطناعي المحلي أولاً" في البحث الأكاديمي.

الكلمات الدالة: التلخيص التجريد ، التلخيص الانتقائي، تنسيق GGUF ، نماذج اللغة الكبيرة المحلية، تلخيص النصوص، خوارزمية تصنيف النصوص.

Abstract
The exponential growth of scientific literature presents a critical challenge for researchers: existing summarization tools force an unacceptable trade-off between factual accuracy and data security. Extractive methods preserve factual integrity but produce incoherent, disjointed summaries lacking academic narrative quality. Cloud-based abstractive methods using Large Language Models (LLMs) generate fluent text but require uploading potentially sensitive research data to external servers, raising serious concerns about privacy, cost, and dependency on third-party services. This paper introduces Abstractive Muse, a novel, self-contained, and privacy-preserving framework designed to eliminate this trade-off. The system is delivered as a standalone desktop application that integrates a classic graph-based algorithm, TextRank, for initial extractive summarization with a locally-executed, open-source LLM (Mistral-7B, ~2.6 GB in GGUF format) for abstractive synthesis, tested on a curated dataset of 10 peer-reviewed scientific PDF documents spanning domains including medicine, computer science, and engineering. We detail the complete development lifecycle, including the architectural design, the implementation of a user-centric Graphical User Interface (GUI), hallucination mitigation strategies, and the engineering decisions that ensure robustness and independence from external APIs. The final application empowers users to select a local PDF document, define summarization parameters, and generate not only a salient extractive summary but also a high-quality, AI-powered academic narrative, all without their data ever leaving their machine. This work presents a replicable blueprint for building practical, secure, and accessible NLP tools and argues for the value of local-first AI in academic research.

Keywords: Abstractive Summarization, Extractive Summarization, GGUF, Local LLM, Text Summarization, TextRank.

Introduction

The modern research landscape is characterized by an ever-accelerating proliferation of scholarly publications. Estimates suggest that millions of new scientific papers are published each year, creating a veritable 'information deluge' that challenges the capacity of even the most diligent researchers (Bornmann & Mutz, 2015). The literature review, a foundational pillar of any academic endeavour, requires scholars to navigate this vast ocean of information to identify relevant prior work, synthesize existing knowledge, and situate their own contributions. This process is not only time-consuming but also cognitively demanding, making it a significant bottleneck in the research lifecycle.

In response, the field of Natural Language Processing (NLP) has long pursued the goal of automatic text summarization. Methodologies are broadly classified into two paradigms: extractive and abstractive.

Extractive summarization represents the classical approach. These methods operate by assigning an importance score to textual units—typically sentences—and selecting a subset to form the summary. Seminal algorithms like TextRank (Mihalcea & Tarau, 2004), which models the text as a graph of interconnected sentences and applies a variant of the PageRank algorithm, have proven effective and computationally efficient. The primary advantage of this paradigm is its factual integrity; since the summary is composed of verbatim sentences from the source, it is guaranteed not to introduce factual errors or 'hallucinations' (Ji et al., 2023). However, this strength is also its main weakness: the resulting summaries, while informative, often lack grammatical cohesion and the narrative flow expected in academic writing, reading more like a collection of bullet points than a coherent paragraph.

Abstractive summarization, in contrast, mirrors the human process of understanding and re-articulating information. These methods aim to generate a novel, concise summary that captures the essence of the source text in new words. The recent ascendance of transformer-based Large Language Models (LLMs) has revolutionized this paradigm, enabling the generation of remarkably fluent and human-like text. Cloud-based services offering access to state-of-the-art models like GPT-4 have become widely available. However, this power comes with significant trade-offs, particularly in an academic context. Reliance on these external APIs introduces critical concerns regarding:

• Data Privacy: Sending potentially sensitive or unpublished research data to a third-party server is often untenable due to confidentiality agreements or institutional policies (Farooq, 2025).
• Cost and Accessibility: API calls are metered and incur costs, creating a barrier to entry for students, independent researchers, or institutions with limited funding.
• Dependency and Reliability: The tool's functionality becomes contingent on a stable internet connection and the continued availability and pricing model of the external service.

While the literature is rich with studies proposing novel algorithms and reporting benchmark scores, there remains a significant gap in addressing the practical, real-world needs of the average researcher. The ideal tool should not force a choice between the disjointed but safe output of extractive methods and the fluent but dependency-laden output of cloud-based abstractive models. Instead, it should synergize these approaches within a framework that is powerful, accessible, and, crucially, secure. Recent studies have highlighted the effectiveness of such hybrid extractive-abstractive approaches in maintaining both factual accuracy and narrative fluency (Bhandari et al., 2023; Divya et al., 2024).

This paper introduces Abstractive Muse, a desktop application engineered to fill this gap. Our work is built on the philosophy of 'local-first AI,' which prioritizes user privacy and operational independence. We present a complete, end-to-end system that combines the strengths of extractive and abstractive summarization without external dependencies.

1. 1.1. Problem Statement

Despite the significant advancements in Natural Language Processing (NLP), researchers continue to face a critical trade-off between factual reliability and narrative quality in automated text summarization. Extractive methods (such as TextRank) ensure factual integrity but often produce disjointed summaries that lack academic flow. Conversely, abstractive methods powered by Large Language Models (LLMs) generate fluent and human-like narratives but suffer from two fundamental challenges:

- Privacy Risks: Cloud-based LLMs necessitate the uploading of potentially sensitive or unpublished research data to external servers, which often conflicts with institutional confidentiality policies.
- Hallucinations: Generative models are prone to "hallucinating"—producing plausible-sounding but factually incorrect information or fabricated data—thereby undermining academic rigor and scientific accuracy.

Consequently, there is a clear void in accessible, local-first tools that can effectively synergize the factual grounding of extractive techniques with the linguistic fluency of abstractive models, while ensuring 100% data privacy and mitigating hallucinations through robust architectural design.

2. 2. Methodology

The architecture of Abstractive Muse is a carefully designed two-stage pipeline, encapsulated within a user-friendly interface. The entire process is executed locally on the user's machine.

3. 2.1. Stage 1: Extractive Analysis

4. 2.1.1. Document Ingestion and Text Extraction

The user initiates the process by selecting a PDF document from their local file system via the GUI. To handle the complexities of PDF parsing, we employ the PyMuPDF library. This choice was deliberate, made after initial tests with other libraries like PyPDF2 resulted in text encoding errors and garbled output for certain PDF formats. PyMuPDF has demonstrated superior robustness in accurately extracting clean text streams from a wide variety of PDF documents, making it a critical component for the reliability of the entire pipeline.

5. 2.1.2. Text Preprocessing and Sentence Segmentation

The raw extracted text is then passed to a preprocessing module. This module performs a series of regular expression-based cleaning operations to prepare the text for analysis: removal of citation markers, stripping of URLs, and normalization of whitespace. Following cleaning, the text is segmented into individual sentences. A key engineering challenge encountered was the unreliability of standard NLP tokenizers in certain user environments. To create a fully self-contained application, we implemented a custom simple_sentence_splitter function using regular expressions.

6. 2.1.3. Graph-Based Salience Ranking (TextRank)

The core of the extractive stage is the TextRank algorithm. The methodology involves vector representation using TF-IDF, similarity matrix construction using cosine similarity, graph modeling of sentences, and iterative ranking to assign salience scores.

7. 2.2. Stage 2: Abstractive Synthesis with a Local LLM

8. 2.2.1. User-Triggered and Controlled Generation

The computationally intensive abstractive stage is only initiated when the user explicitly clicks the 'Generate Academic Introduction (Local AI)' button. This design conserves system resources and gives the user full control over the workflow.

9. 2.2.2. Local Model Loading and Management

To avoid external dependencies, we utilize the ctransformers library. We selected a quantized version of a high-performing open-source model: TheBloke/Mistral-7B-Instruct-v0.1-GGUF (Jiang et al., 2023). The use of quantization is a deliberate engineering decision, reflecting established techniques like GPTQ that enable large models to run on consumer-grade hardware with minimal performance loss (Frantar & Alistarh, 2023). Recent advancements in GGUF format have further optimized local deployment for CPU-only environments (Yadav & Bhargavi, 2025). This choice offers an excellent balance between performance and a small file size (~2.6 GB). The library automatically downloads and caches the model on first use, and it is configured to run entirely on the CPU, ensuring broad hardware compatibility.

10. 2.2.3. Prompt Engineering for Academic Tone

The quality of the LLM's output is highly dependent on the prompt (Wei et al., 2023). The extractive summary is embedded within a carefully engineered prompt template that instructs the

model to adopt the persona of an academic writer and weave the provided key points into a compelling narrative.

11. 3. System Implementation and GUI

A core objective was to create a tool that is immediately usable by its target audience. We chose Python's built-in Tkinter library to develop an intuitive GUI. The interface is organized into logical sections: a file selection bar, a control panel for summarization parameters, main results display area, and a status bar for real-time feedback.

12. 4. Results and Discussion

The final application, Abstractive Muse, successfully integrates the strengths of both summarization paradigms within a single, user-friendly interface. The results of this work are not just algorithmic but also functional, representing a tangible artifact for the research community.

4.1. Dataset Description

To evaluate the system, we assembled a curated test dataset comprising 10 peer-reviewed scientific papers in PDF format, manually collected from open-access repositories including PubMed Central, arXiv, and IEEE Xplore. The dataset spans three domains: biomedical research (4 papers), computer science and AI (4 papers), and engineering (2 papers). Document sizes range from 4 to 18 pages, with an average of 6,200 words per document. All documents contain structured academic content including abstracts, methodology sections, results, and references. This domain diversity was intentional, as it allows evaluation of the system’s robustness across different writing styles, terminologies, and levels of technical density. No personally identifiable or confidential data was included in the evaluation corpus. Evaluation was conducted qualitatively by assessing the coherence, accuracy, and academic tone of the generated summaries against the original documents.

4.2. Hallucination Mitigation in the Local LLM

Hallucination—the generation of plausible but factually incorrect or fabricated content—is a well-documented limitation of LLMs (Ji et al., 2023; Huang et al., 2025). In the context of academic summarization, a hallucinated fact (e.g., an incorrect statistic or a fabricated citation) can have serious consequences for the integrity of research. Abstractive Muse addresses this challenge through a concrete, multi-layered mitigation strategy. First, the LLM is never permitted to summarize the raw document directly. Instead, it operates exclusively on the extractive summary produced by TextRank, which consists entirely of verbatim sentences from the source. This “grounded generation” approach fundamentally constrains the model’s output space: it can only paraphrase and connect sentences that already exist in the document, drastically reducing the risk of fabricated content (Dziri et al., 2022). Second, the prompt template explicitly instructs the model to base its narrative solely on the provided key points and not to introduce any external knowledge or facts. Third, any numerical values, citations, or technical metrics present in the extractive summary are passed directly into the prompt, ensuring they appear verbatim in the final output. Finally, the dual-output interface of the application displays both the extractive and abstractive summaries side by side, allowing the user to immediately verify the AI-generated narrative against the factual source sentences. This human-in-the-loop verification step serves as the final safeguard against any residual hallucinations that the automated pipeline may not have fully prevented.

4.3. Comparative Case Study: Abstractive Muse vs. Cloud-Based AI

To demonstrate the practical efficacy of Abstractive Muse, we conducted a comparative analysis using a recent medical research paper titled 'A Deep Learning-Based Method for Skin Cancer Detection' (AbdElsalam et al., 2024). The objective was to evaluate the quality of the generated summary and the preservation of technical accuracy compared to a standard cloud-based AI tool (e.g., ChatGPT/ChatPDF).

The cloud-based tool produced a generalized summary that, while coherent, omitted critical technical metrics such as the F1-score (0.9803) and focused primarily on broad objectives. In contrast, Abstractive Muse followed its hybrid pipeline: first extracting salient sentences via TextRank to ensure no technical data was lost, and then synthesizing them into a professional academic narrative using the local LLM.

Feature	Cloud-Based AI	Abstractive Muse (Our Tool)
Data Privacy	Files uploaded to external servers (Risk)	100% Local processing (Secure)
Technical Accuracy	May omit specific metrics (e.g., F1-score)	Preserves all extracted metrics
Narrative Tone	General/Conversational	Professional Academic Narrative
Operational Cost	Subscription/API fees	Completely Free/Offline

The results of this case study underscore the superiority of Abstractive Muse in research environments where data confidentiality and technical precision are paramount.

13. 4.5. Key Differentiators and Contributions

1. Focus on Workflow, Not Just Algorithms: Our work focuses on designing a complete user workflow, emphasizing Human-Computer Interaction (HCI) as much as it is on NLP.
2. Privacy-Preserving, Local-First Architecture: By exclusively using a locally-run LLM, Abstractive Muse guarantees that the user's data never leaves their machine (Vaid et al., 2024).
3. Democratization of Advanced AI Tools: By creating a self-contained application that requires no API keys, we democratize access to advanced abstractive summarization technology.
4. Pragmatic and Robust Engineering: Our paper transparently documents the solution to real-world engineering problems, such as the implementation of a custom sentence splitter.

14. 5. Conclusion and Future Work

This paper has detailed the design and implementation of Abstractive Muse, a self-contained, privacy-preserving desktop application for hybrid text summarization. By combining a robust TextRank algorithm with a modern, locally-executed LLM, we have created a practical tool that offers both the factual grounding of extractive methods and the narrative fluency of abstractive synthesis.

References

1. AbdElsalam, M., & et al. (2024). A Deep Learning-Based Method for Skin Cancer Detection. Journal of Medical Imaging.
2. Alansari, A., & et al. (2026). Large language models hallucination: A comprehensive review. ScienceDirect.
3. Bhandari, A. (2023). On the Faithfulness of Abstractive Summarization Models: A Hybrid Approach. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL).
4. Bornmann, L., & Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. Journal of the Association for Information Science and Technology, 66(11), 2215-2222.

5. Divya, S., Sripriya, N., Andrew, J., & Mazzara, M. (2024). Unified extractive-abstractive summarization: a hybrid approach utilizing BERT and transformer models. PeerJ Computer Science, 10, e2424.
6. Farooq, A. (2025). Securing local LLMs for academic research: a human-system integration analysis and evolution of TAUCHI-GPT. Springer Nature.
7. Frantar, E., & Alistarh, D. (2023). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
8. Ji, Z. (2023). Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12), 1-38.
9. Jiang, A. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.
10. Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP).
11. Nenkova, A., & McKeown, K. (2011). Automatic summarization. Foundations and Trends in Information Retrieval, 5(2-3), 103-233.
12. Alaiat, H. H. M. (2023). The Challenges and Difficulties Encountered by Computer Science Educators in Higher Education, Specifically Focusing on the Utilization of Chat GPT Technology in Libya. Bani Waleed University Journal of Humanities and Applied Sciences, 8(4), 120-137.
13. Pesaranghader, A., & et al. (2026). Hallucination Detection and Mitigation in Large Language Models. arXiv. Retrieved from arXiv:2601.09929
14. Meelad, R. A., Mousa, M. A., & ulwahad AlSharaa, M. A. (2026). Convolutional Neural Network Models For Automated Art Style Identification: Design Training, And Evaluation. *Bani Waleed University Journal of Humanities and Applied Sciences*, 178-188.
15. Sharma, N., & et al. (2025). A Hybrid Extractive and Encoder-Decoder-Based Approach for Mitigating Hallucination. Springer.
16. Almarimi, A. F., & Salem, A. M. (2025). Machine Learning using Simple Linear Regression. Bani Waleed University Journal of Humanities and Applied Sciences, 10(3), 178-184.
17. TheBloke. (2023). Mistral-7B-Instruct-v0.1-GGUF. Hugging Face. Retrieved from https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.1-GGUF
18. Touvron, H. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
19. Dreheeb, A. M., & El Tajouri, H. (2025). Role Of Artificial Intelligence In Enhancing Cyber Security. Bani Waleed University Journal of Humanities and Applied Sciences, 10(3), 121-129.
20. Vaid, A. (2024). Local large language models for privacy-preserving querying and interpreting echocardiogram reports. PubMed, 38687616.
21. Sheggaf, Z. M., Ehzazat, S. S. W., & Esdeira, A. A. D. (2023). Fluidity of aluminum piston alloy with different amount of pouring temperature. *Bani Waleed University Journal of Humanities and Applied Sciences*, *8*(3), 31-36.
22. Shouran, Z., Mousa, M. A., Alatresh, S. A., & ulwahad AlSharaa, M. A. (2025). Security and Privacy in the Internet of Things: Issues, Challenges, and a Deep Learning-Based Intrusion Detection Framework. Bani Waleed University Journal of Humanities and Applied Sciences, 10(4), 225-233.
23. Wei, J. (2023). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Advances in Neural Information Processing Systems (NeurIPS).
24. Yadav, A., & Bhargavi, R. (2025). Optimizing LLMs Using Quantization for Mobile Execution. International Conference on ICT for Sustainable Development.