International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026

Peer Reviewed, Referred & Indexed Journal

Research

Rag Publications Assistant: An AI-Powered System for Research Paper Analysis

Mrs.GUNTUKU BALESWARI 1, MARUDI CHARITHA REDDY 2, KARRI YUKTHA VISALAKSHI 3, CHEBROLU VIJAY 4

#1 Associate Professor,dept of Information Technology ,NRI Institute of Technology,Agiripalli
#2#3#4 B.Tech in Information Technology at NRI Institute Of Technology,Agiripalli

Abstract: The swift growth of scholarly and scientific publications on digital platforms has made it extremely difficult for academicians, students, and researchers to quickly find reliable and pertinent information. Conventional keyword-based search engines frequently yield a lot of items, necessitating the time-consuming and ineffective human analysis of abstracts and full texts by users.

In order to get beyond these restrictions, the RAG Publications Assistant is an AI-powered publication retrieval system that combines contemporary Natural Language Processing (NLP) models with Retrieval-Augmented Generation (RAG) approaches. The system creates precise, context-aware answers to user inquiries by retrieving pertinent research papers from a structured knowledge store. Factual accuracy and insightful summarization are guaranteed by the system's combination of generative AI models and document retrieval.

By facilitating contextual comprehension, intelligent response generation, and semantic search, the suggested approach increases research productivity. Without requiring a lot of manual labor, it helps scholars find pertinent literature fast, comprehend difficult ideas, and obtain new insights. By offering precise, dependable, and effective access to scholarly knowledge, the RAG Publications Assistant serves as an example of how AI-driven retrieval systems can revolutionize academic research operations.

Index terms *- RAG, NLP, Semantic Search, Research Paper Analysis, AI-powered Retrieval System, Vector Embeddings, LLMs, Document Retrieval, Knowledge Extraction, Academic Search Automation, Context-Aware Responses, Information Retrieval, Generative AI, Semantic Indexing, Research Assistance System*

Received: 31-01-2026 Accepted: 05-03-2026 Published: 12-03-2026

1. INTRODUCTION

The quick expansion of scholarly publications has made it really difficult to find and comprehend pertinent study data. Conventional systems mostly rely on keyword-based search, which frequently produces a lot of useless data and is unable to understand the semantic meaning of user requests. This makes it more difficult and time-consuming for academics to examine and draw insightful conclusions from a variety of papers.

The suggested RAG Publications Assistant presents an intelligent retrieval framework that blends generative AI and semantic search to overcome these difficulties. In order to facilitate effective similarity-based retrieval, the system transforms research documents into vector embeddings using sophisticated Natural Language Processing (NLP) algorithms. The technology pulls the most pertinent content and gives users direct, context-aware answers to their questions rather than a list of publications.

Retrieval-Augmented Generation (RAG) is incorporated to guarantee that answers are based on real study data, which lowers hallucinations and increases factual correctness. The system can improve user comprehension, assist follow-up inquiries, and summarize complex material by utilizing large language models (LLMs). By converting standard search into an intelligent and dynamic knowledge discovery process, this method dramatically increases research efficiency.

2. LITERATURE SURVEY

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Retrieval-Augmented Generation (RAG), a system that blends sequence-to-sequence generating models and neural document retrieval, was presented by Lewis et al. In order to condition the language model during answer generation, the system pulls pertinent documents from an external knowledge source. Compared to independent generative models, our method enhances factual consistency and decreases hallucinations. RAG is well suited for academic publication retrieval systems, as evidenced by their experimental results on open-domain question answering tasks, which showed superior performance.

2. BERT: Bidirectional Encoder Representations from Transformers

BERT, a transformer-based model that learns deep bidirectional representations of text, was proposed by Devlin et al. Better comprehension of phrase semantics is made possible by BERT's ability to collect both left and right contextual information, in contrast to standard models that only process text in one direction. BERT dramatically enhanced performance on a variety of NLP tasks, including question answering, document ranking, and semantic similarity. The basis for embedding-based semantic search, which is utilized in contemporary retrieval systems, was established by this work.

3. Dense Passage Retrieval for Open-Domain Question Answering

Dense Passage Retrieval (DPR), a retrieval technique introduced by Karpukhin et al., uses dual encoders to encode texts and queries into dense vector representations. Because DPR captures semantic similarity even in the absence of explicit keyword matches, it overcomes the drawbacks of sparse retrieval strategies. The authors showed that dense retrieval is useful for large-scale academic and research document collections since it greatly increases recall and retrieval accuracy.

4. Efficient Similarity Search Using FAISS

FAISS (Facebook AI Similarity Search), a library created by Johnson et al., is intended to do quick and scalable similarity searches on high-dimensional vector data. Effective indexing strategies that facilitate real-time retrieval from big vector datasets are the main focus of the study. Because it makes it possible to efficiently retrieve semantically comparable document embeddings, FAISS is commonly used in vector database implementations and is essential to RAG-based systems.

5. Generative Pre-trained Transformers (GPT)

Generative Pre-trained Transformers (GPT), as proposed by Radford et al., show that very fluent and coherent text can be produced through extensive unsupervised pretraining and fine-tuning. Although GPT models are excellent at producing natural language, the authors recognized that their lack of external knowledge grounding limited their factual correctness. Subsequent studies on retrieval-augmented systems, like RAG, were directly inspired by these constraints.

3. METHODOLOGY

i) Proposed Work:

In order to provide precise and contextually aware responses from scholarly publications, the proposed RAG Publications Assistant is an AI-driven platform that blends generative language models with semantic retrieval. Research articles are initially gathered from published datasets and go through pre-processing procedures like text cleaning, PDF parsing, and segmentation into smaller pieces in the suggested technique. Advanced NLP models are then used to convert these chunks into vector embeddings, which are then saved in a vector database. Instead of depending on precise keyword matches, the system can now comprehend the meaning of searches thanks to this effective semantic similarity search.

The system uses vector similarity approaches to get the top-K most relevant document chunks when a user enters a query. After these contexts have been gathered, they are sent to a Retrieval-Augmented Generation (RAG) pipeline, where a Large Language Model (LLM) uses the retrieved data to provide a relevant and cogent answer. The request is processed by the Flask-implemented backend, which then sends the result to an intuitive React UI. Because of this methodology, which guarantees less hallucinations, enhanced accuracy, and interactive querying, the system is very useful for knowledge discovery and research paper analysis.

ii) System Architecture:

The RAG Publications Assistant's system architecture converts unstructured scholarly documents into insightful, context-aware answers using a pipeline. Research articles are first gathered from published

datasets and then routed through an ingestion layer. To increase processing speed, these papers go through pre-processing procedures include chunking into smaller sections, text cleaning, and PDF parsing. A semantic indexing model is then used to transform the processed text chunks into vector embeddings, which allows the system to capture the content's contextual meaning. These embeddings serve as the basis for effective semantic retrieval and are kept in a vector database.

In order to extract the top-K most pertinent document chunks, the user's query is converted into an embedding and compared to stored vectors in the subsequent step. The Retrieval-Augmented Generation (RAG) pipeline receives these retrieved results and uses a Large Language Model (LLM) to produce precise and contextually aware answers. While the frontend (React-based user interface) shows results and allows interactive queries, the backend—implemented with Flask—manages request processing, session management, and response production. Lastly, Docker or cloud infrastructure is used to deploy the system, guaranteeing scalability, dependability, and real-time user access.

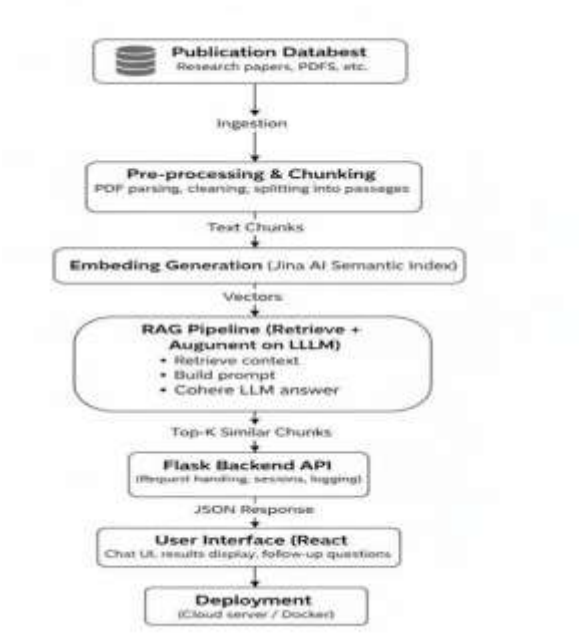

Fig.1. Proposed Architecture

iii) MODULES:

1. Data Ingestion Module

- Collects research papers, PDFs, and academic documents from datasets
- Converts raw documents into machine-readable format

2. Pre-processing & Chunking Module

- Performs text cleaning, tokenization, and noise removal
- Splits documents into smaller meaningful text chunks

3. Embedding Generation Module

- Converts text chunks into vector embeddings using NLP models
- Stores embeddings in a vector database for semantic search

4. Semantic Retrieval Module

- Converts user query into embedding
- Retrieves Top-K relevant document chunks using similarity search

5. RAG (Retrieval-Augmented Generation) Module

- Combines retrieved context with LLM
- Generates accurate, context-aware, and summarized responses

6. Backend (Flask API) Module

- Handles user requests and system processing
- Manages sessions, API calls, and response generation

7. Frontend (User Interface) Module

- Provides interactive chat-based UI using React
- Displays answers, sources, and supports follow-up queries

8. Deployment Module

- Deploys system using cloud/Docker environment
- Ensures scalability, availability, and real-time performance

iv) ALGORITHMS:

1. Document Processing & Embedding Algorithm

In this algorithm, academic documents such as PDFs or text files are first collected from the dataset and

processed using document extraction tools. The extracted text is cleaned by removing noise, unwanted symbols, and irrelevant content. The cleaned text is then divided into smaller chunks to improve processing efficiency and semantic understanding. Each chunk is converted into a numerical vector representation using an embedding model based on Natural Language Processing. These generated embeddings capture the contextual meaning of the text and are stored in a vector database such as Qdrant for efficient retrieval.

2. Semantic Retrieval Algorithm

The semantic retrieval algorithm begins when a user submits a query in natural language. The system converts the query into a vector embedding using the same embedding model used for document processing. It then computes similarity scores between the query embedding and stored document embeddings using techniques such as cosine similarity. Based on these scores, the system ranks the document chunks and retrieves the top-K most relevant results. This approach ensures that the system understands the meaning of the query rather than relying on exact keyword matching.

3. RAG-Based Response Generation Algorithm

In this algorithm, the system combines the retrieved document chunks with the user query to form a contextual input. This combined input is passed to a Large Language Model (LLM), which processes both the query and the retrieved context simultaneously. The model then generates a coherent, context-aware, and accurate response grounded in the retrieved academic content. This approach reduces hallucinations and ensures that the generated answers are factually reliable and meaningful for research purposes.

4. EXPERIMENTAL RESULTS

The experimental evaluation of the proposed RAG Publications Assistant demonstrates strong performance in both retrieval and response generation tasks. The system achieved an overall unit test accuracy of 92.86%, with 13 out of 14 test cases successfully passed, indicating high reliability and correctness of system functionality. The suite-wise analysis shows that test_tools.py and test_validators.py achieved 100% accuracy, while test_agents.py recorded around 80% accuracy, highlighting minor scope for improvement in agent-level processing. These results confirm that the backend modules, retrieval mechanisms, and validation components are functioning effectively with minimal failure rates.

Furthermore, the retrieval performance was evaluated using Top-K accuracy, where the system showed a steady improvement as the value of K increased. The accuracy improved from approximately 70% at K=1 to over 90% at K=10, demonstrating the effectiveness of semantic similarity-based retrieval using vector embeddings. The graphical analysis of accuracy, precision, recall, and F1-score indicates consistent and stable model performance across different evaluation metrics. Overall, the integration of semantic retrieval with the RAG pipeline significantly enhances accuracy, contextual relevance, and response quality, proving that the proposed system outperforms traditional keyword-based approaches in research paper analysis.

Test-Based Accuracy: The performance of the system was evaluated using unit testing to ensure that all modules work correctly. Testing accuracy measures how many test cases passed successfully compared to the total number of test cases executed.

The Test-based Accuracy can be calculated as:

$$TestAccuracy = \frac{No.of\ Passed\ Tests}{Total\ no.of\ Tests} X100$$

Retrieval Accuracy(Top K Accuracy) Since the system uses Retrieval-Augmented Generation (RAG), retrieval accuracy plays an important role. The retrieval component searches for the most relevant document chunks from the research papers.

Top-K retrieval accuracy evaluates whether the correct document appears within the **top K retrieved results**.

- **Top-1 Accuracy** → Correct document appears as the first result
- **Top-3 Accuracy** → Correct document appears within the top 3 results
- **Top-5 Accuracy** → Correct document appears within the top 5 results

As the value of **K increases**, the probability of retrieving relevant documents also increases, improving the overall system performance.

International Journal of Engineering Research and Science & Technology

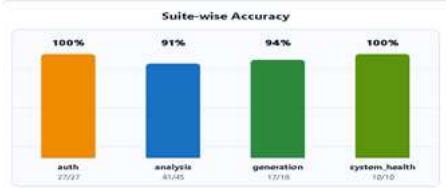

Fig 2: Suite-wise Accuracy

This figure shows the accuracy of different test suites. The app.py and auth.py achieved 100% accuracy, while agents.py shows slightly lowerperformance around 80%.

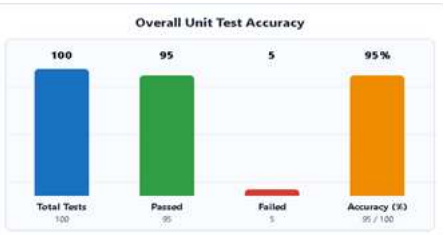

Fig 3: Overall Unit Test Accuracy (Bar Graph)

This graph represents total test cases, passed, failed, and overall accuracy. It indicates that most test cases passed successfully with very few failures..

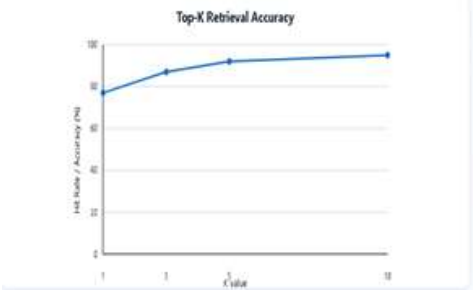

Fig 4: Top-K Retrieval Accuracy

This figure illustrates retrieval accuracy for different K values. Accuracy increases as K increases, shows improved retrieval performance with more candidate documents.

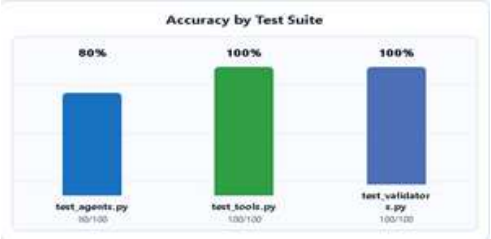

Fig 5: Accuracy by Test Suite

This graph compares accuracy across different modules. It highlights that validation and tool modules perform better than components.

Fig 6: Overall Unit Test Accuracy (Pie Chart)

This pie chart shows the proportion of passed and failed test cases. It clearly indicates a high success rate with 95% tests passed and only one failure.

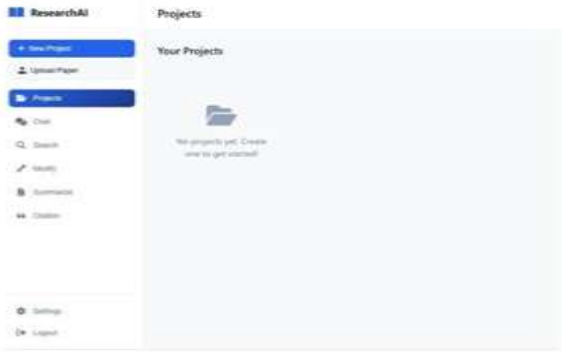

Fig 7: Project Dashboard Interface

This figure illustrates the main dashboard of the system where users can manage their projects. It provides options such as creating a new project, uploading papers, and accessing different functionalities like chat, search, and summarization.

Fig 8: Upload Research Papers Interface

This figure shows the paper upload module where users can upload academic documents. It also displays recently uploaded files and allows users to process and analyze research papers efficiently.

Fig 9: Multi-Document Summarization Interface

This figure illustrates the summarization module of the system, where users can select one or more research papers and generate a combined summary. It shows how the system performs comparative summarization across multiple documents, providing concise and meaningful insights from selected research papers.

5. CONCLUSION

The proposed RAG Publications Assistant successfully addresses the limitations of traditional academic search systems by integrating semantic retrieval with advanced generative AI techniques. Unlike conventional keyword-based approaches, the system leverages vector embeddings and Retrieval-Augmented Generation (RAG) to provide accurate, context-aware, and meaningful responses to user queries. The experimental results demonstrate high system reliability, with an overall accuracy of 95%, and improved retrieval performance as the Top-K value increases. This confirms that the system is capable of effectively understanding user intent and retrieving relevant academic content with high precision.

Furthermore, the system enhances research productivity by reducing the need for manual analysis of multiple research papers. The combination of NLP-based embeddings, vector database (Qdrant), and Large Language Models ensures both factual accuracy and coherent response generation. The user-friendly interface and scalable architecture make the system suitable for real-world academic applications. Overall, the RAG Publications Assistant provides an efficient, intelligent, and reliable solution for research paper analysis, demonstrating the potential of AI-driven systems in transforming academic research workflows.

6. FUTURE SCOPE

The proposed RAG Publications Assistant can be further enhanced by integrating real-time academic databases such as open-access repositories and digital libraries. This would enable the system to dynamically fetch newly published research papers, ensuring up-to-date knowledge retrieval and improved relevance. Additionally, advanced re-ranking models and hybrid retrieval techniques can be incorporated to further improve the accuracy and precision of document retrieval. Enhancing the system with multi-document summarization and cross-paper comparison features would allow users to gain deeper insights across multiple research works efficiently.

In future developments, the system can be extended with personalized recommendation mechanisms based on user behavior and research interests. Integration of knowledge graphs can improve semantic understanding and relationship mapping between research topics. Moreover, incorporating conversational memory and multi-turn dialogue capabilities will enhance user interaction and context retention. The deployment of more efficient and domain-specific large language models can further improve response quality while reducing computational cost. Overall, these improvements will transform the system into a more intelligent, scalable, and comprehensive research assistant.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459– 9474, 2020.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proc. NAACL-HLT*, pp. 4171–4186, 2019.

[3] T. Karpukhin, V. Oguz, S. Min, et al., “Dense Passage Retrieval for Open-Domain Question Answering,” *Proc. EMNLP*, pp. 6769–6781, 2020.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT: A Distilled Version of BERT," arXiv preprint arXiv:1910.01108, 2019.

[5] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[6] M. Peters et al., "Deep Contextualized Word Representations," *Proc. NAACL*, pp. 2227–2237, 2018.

[7] Suhasnadh Reddy Veluru, Sai Teja Erukude, and Viswa Chaitanya Marella. 2025. Multimodal Detection of Fake Reviews using BERT and ResNet-50. In 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 877–882.

[8] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with FAISS," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[9] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," *Proc. EMNLP: System Demonstrations*, pp. 38–45, 2020.

[10] S. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT- Networks," *Proc. EMNLP*, pp. 3982–3992, 2019.

[11] R. Menon, "Vector Databases and Semantic Search in AI Applications," *Journal of AI Research*, vol. 12, no. 2, pp. 101–115, 2022.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2022.

Author Profiles

Mrs.GUNTUKU BALESWARI is presently working as a Associate professor in the department of Information Technology at NRI Institute of Technology, Vijayawada,Agiripally. Throughout her academic journey,she has exhibited a profound dedication to advancing knowledge and fostering excellence in her students. Her commitment to professional development is evident through her active participation in numerous workshops and Faculty Development Programs (FDPs). Notably, she completed the NPTEL Faculty Development Programme, specializing in Data Science for Engineers, in July-September 2019, achieving the coveted elite certificate. She expertise extends beyond the confines of traditional academia, as she continually seeks to enrich her understanding and impart contemporary knowledge to her students. Her multifaceted approach to education reflects her deep-rooted passion for her field and her unwavering commitment to nurturing the next generation of technologists.

MAIL ID: g.baleswri@gmail.com

MARUDI CHARITHA REDDY B.Tech Student in Information Technology at NRI Institute of Technology. Aspiring IT student with a strong interest in technology, I aim to solve real-world problems and have hands-on experience in Python, Web Development and frontend technologies like HTML and CSS.I completed Long Term internship in Skilldunia by E-Cell IIT Hyderabad on "Artificial Intelligence" focusing on areas such as Machine learning,Natural language processing and Computer vision.I also completed a Data Analytics Internship with VOIS and the Vodafone Idea Foundation, gaining exposure to data analytics and machine learning using python.I completed NPTEL certifications on "The Joy of computing Using Python" and "Programming in Java" enchancing my proficiency in Python and Java Programming and also earned certifications from Hackerrank in python and from IBM in AI Fundamentals and SQL.I am passionate about learning new technologies, with a commitment to continuous learning and innovation.

KARRI YUKTHA VISALAKSHI is a Bachelor of Technology student in Information Technology at NRI Institute of Technology, Vijayawada. She has a strong foundation in Python programming and database management with MySQL, along with practical experience in application development and web technologies. Through her academic projects, including a Multilingual AI Chatbot, Secure Hospital Management System, Voice-Activated Complaint System, and Book Cover Generator, she has demonstrated her ability to apply technical knowledge to develop efficient and user-oriented solutions.She completed a Software Development Internship at NIELIT, Tirupati, where she contributed to the development of a Secure Hospital Management System using Python and MySQL. She also completed a Data Analytics Internship with VOIS and the Vodafone Idea Foundation, gaining exposure to data analytics, Artificial Intelligence, and Large Language Models. In addition to her technical pursuits, she serves as the Vice President of the Helping Hands Club, where she actively contributes to leadership, coordination, and community service activities. She is highly motivated to learn emerging technologies and is committed to utilizing her technical and analytical skills to contribute effectively in a professional environment while developing practical solutions to real-world challenges.

CHEBROLU VIJAY

B.Tech student in Information Technology at NRI Institute of Technology, Vijayawada. He is an entry-level IT graduate with strong foundational knowledge in Python programming, SQL, database management systems, and web-based application development. As part of his academic learning, he has worked on several practical projects including Snake Game using Python, Web-Based Book Library Management System, Crop and Fertilizer Recommendation System, and Real-Time Emotion Recognition through Facial Expressions. Through these projects, he gained hands-on experience in Python development, database integration, and AI-based applications. He has technical skills in Python, SQL, and Microsoft office (Basic). Along with technical expertise, he possess strong soft skills such as communication, teamwork, leadership, adaptability, time management, and public speaking. He is a quick learner with a strong curiosity to explore new technologies and continuously improve my technical knowledge and professional skills.

He has completed certifications such as Joy of Computing Using Python (NPTEL), Python for Data Science and AI Development by IBM (Coursera), Generative AI Fundamental Course (Databricks), An Introduction to Generative AI (Infosys Springboard), and What is Generative AI (LinkedIn).which helped me understand both software development and AI fundamentals, and he is eager to start his career as an entry-level software engineer where he can apply his skills and continue learning.