RAG – BASED WEB INTELLIGENCE SYSTEM FOR DOMAIN SPECIFIC SENTIMENT ANALYSIS

Samydurai A
Prof., Dept. CSE
SRM Valliammai Engineering College
Kattankulathur, India

Sivachandiran K
Dept. CSE
SRM Valliammai Engineering College
Kattankulathur, India

Sathish Kumar D
Dept. CSE
SRM Valliammai Engineering College
Kattankulathur, India

Sanjeev Kumar A
Dept. CSE
SRM Valliammai Engineering College
Kattankulathur, India

Abstract: The recent surge of online content including news articles, blogs, and social media posts has generated an overwhelming amount of information, reflecting public opinion in many different fields. However, mining valuable insights from this data is no doubt challenge since web content is generally highly unstructured and standard sentiment analysis methods are quite limited. In this paper, we present a Retrieval-Augmented Generation (RAG) based Web Intelligence System for Domain-Specific Sentiment Analysis which combines web data collection, smart retrieval, and large language models to give highly accurate and context-aware sentiment suggestions. Our system first collects information specific to your domain from various web sources via a news scraping module and then clean the content so that it can be analyzed later on. The data is broken down into workable pieces and called embeddings which are then stored in a vector database that allows fast semantic searching. Next, we employ a Retrieval-Augmented Generation (RAG)method which queries the vector database for contextual information, and then uses a large language model to generate the sentiment analysis and explanation. We went for a web app that runs on Streamlit, a Python library geared for data-science and machine-learning apps. At the same time, it also runs in the background, performing the whole pipeline integral with LangChain, as well as the vector search capabilities from Databricks. Experimental result also shows that method we have devised remarkably outperforms the sentiment detection accuracy and understanding of context done by means of a mere keyword-based (conventional) sentiment analysis techniques. The tool furnishes a powerful answer for the examination of domain-specific public sentiment, and the same could be extended to such areas as market analysis, brand monitoring, and decision support systems.

Keywords: Retrieval-Augmented Generation (RAG), Sentiment Analysis, Web Intelligence, Vector Database LangChain Natural Language Processing, Domain-Specific Analysis.

1. INTRODUCTION

The internet and digital communication platforms exponential growth have led to the generation of huge online contents every day. Apart from news blogs forums, and social media platforms continuously produce large volumes of textual data that show opinions, discussions, and sentiments of people on different topics. Organizations, researchers, and decision-makers are increasingly using this information to understand how people perceive different issues, keep track of changing trends, and make well-informed decisions. Consequently, sentiment analysis has emerged as a key natural language processing technique for detecting and explaining emotions, opinions, and attitudes in textual data.

Conventional sentiment analysis methods mainly use keyword-based strategies or machine learning models fitted to annotated data. These approaches do well for broad sentiment categorization but can hardly understand the sentiment properly in specialized contexts. Depending on the domain, words and expressions might have totally different meanings. For instance, language related to technology, healthcare finance, or politics might need a deeper level of understanding that goes beyond mere keyword matching. Furthermore, standard sentiment analysis tools usually do not have the capability to draw on contextual information from external knowledge sources, which is one of the reasons why they are not very efficient in handling complex or changing web content. There is also a concern about the ever-expanding amount of information on the web. To derive valuable findings from large chunks of highly unstructured data requires collection, processing, and analysis. It is simply not feasible for humans to perform the analysis however conventional automated systems might not be able to identify the semantic links between the same pieces of information scattered through several sources. So, smart machines that can find the right contextual information and then combine it with state-of-the-art language models are needed to enhance both the accuracy and trustworthiness of sentiment analysis.

Thanks to the recent progress in artificial intelligence, especially to the development of large language models and retrieval-based architectures, the door has been opened wide for the improvement of text analysis systems. Retrieval-Augmented Generation (RAG) is a promising concept that merges the two components: information retrieval systems and generative language models. In the RAG system first the most pertinent information is obtained from a knowledge base or a vector database, and then this piece of information is used as context by the language model when creating a response. Thus, RAG systems that jointly perform retrieval and generation can go beyond the shortcomings of language models working alone and deliver analysis results that are more trustworthy.

Inspired by these situations, the current document conveys a RAG-Based Web Intelligence System for Domain-Specific Sentiment Analysis. The designed system gathers domain-related data from the web via computer-aided web scraping methods and then runs the obtained information through the properly arranged analysis stages. The raw text data is firstly purified, fragmented into smaller pieces, and finally turned into vector embeddings which are kept in a vector database for quick semantic search and content retrieval.

This system is a web-based platform that integrates a Streamlit interface as a user front-end and a back-end processing layer combining LangChain and Databricks vector search capabilities. Whenever users put forward a request or ask for sentiment analysis, the system will fetch relevant contextual information from the vector database through the Retrieval-Augmented Generation pipeline. Then this collected knowledge will be used along with the language model for domain-specific sentiment classification and explanation. This solution significantly improves the level of accuracy, the ability to understand context, and the capability to scale of sentiment analysis systems in general. It would be applicable not only in the fields of public attitude monitoring, product/service market research and brand image analysis, but also in decision support systems.

2. LITERATURE SURVEY

The rapid growth of the internet and social media has led to the creation of a large amount of text data daily. News articles blogs product reviews, and social media posts have information in them that can be used to understand public opinions, attitudes, and sentiments. Sentiment analysis has thus evolved into a significant focus area in natural language processing (NLP) as it helps organizations and researchers to derive valuable insights from text data. Conventional methods for sentiment analysis either depend on machine learning algorithms or on lexicon-based methods to categorize text as expressing positive, negative, or neutral sentiments. Though these approaches have demonstrated effectiveness, they are frequently unable to comprehend sentiments accurately in complex or highly specialized contexts where language usage and terminology can differ greatly [1].

A widely adopted method in sentiment analysis is lexicon-based sentiment classification. This involves using pre-compiled lists of positive and negative words to judge the sentiment of a document. Such methods mainly work by looking at the number and strength of the sentiment-indicative words in a text. In this way, they calculate its overall polarity. Even though these lexicon-based approaches are straightforward and can be implemented with minimal effort, they do not usually acknowledge contextual subtleties such as sarcasm, domain-specific vocabulary, and implicit sentiment expressions. Consequently, their ability to analyze specialized domains such as healthcare, finance, or technology may be quite limited [2].

Another direction for sentiment analysis is the use of machine learning, which has become a popular means of making classification more accurate. Supervised learning methods like Support Vector Machines (SVM), Nave Bayes, and Random Forest models have been utilized for sentiment classification work with annotated datasets. The advantage of these methods is that they identify the characteristic features of sentiment in training data and thus can outperform rule-based systems in terms of accuracy. On the other hand, conventional machine learning models depend heavily on the availability of a large amount of annotated data during the training phase and they may not generalize well to different domains without being retrained. Besides, these models are often incapable of integrating external contextual information when dealing with newly or evolving web contents [3].

Sentiment analysis using deep learning and transformer-based language models is a field that keeps seeing advancements thanks to the development of more complex techniques. For instance, models like BERT, GPT, and other transformer variants have made great strides in deepening the understanding of textual contexts. Such models didn't just stop at getting the meaning right; they even managed to identify subtle semantic patterns and deliver superior outcomes in almost every NLP task, sentiment analysis included. However, a lone language model, no matter how advanced, can sometimes churn out errors especially when it's deprived of necessary external knowledge or domain-specific information [4].

Therefore, to overcome these shortcomings, the latest studies have discovered the possibility of coupling information retrieval mechanisms and language models via Retrieval-Augmented Generation (RAG). RAG is a method that utilizes a retrieval unit and a generative language model to enhance the precision and alignment of the results with the context. Essentially, firstly a relevant fact or information is pulled out from an external knowledge source like a set of documents or a vector database. Secondly, the fetched material is then used as a supplement to the language model, thereby equipping it with the capability to create closer, more knowledgeable responses. This combined, or in other words, hybrid method has yielded leading-edge outcomes particularly in question answering, knowledge retrieval as well as text analysis [5].

In addition to the above, different works have also looked into how vector databases and semantic search engines could be employed to boost the retrieval of information in NLP systems. Text representations can be converted to numerical vectors in a high-dimensional space using deep learning models that produce vector embeddings. This way, the documents or text pieces returned by the systems would be the ones related to the context of the query rather than the ones containing the query keywords only. This is possible through embeddings or semantic similarity searches. To store and retrieve embedding vectors for large-scale applications in an efficient way, vector databases such as FAISS, Pinecone, and Databricks Vector Search have been extensively used.[18]

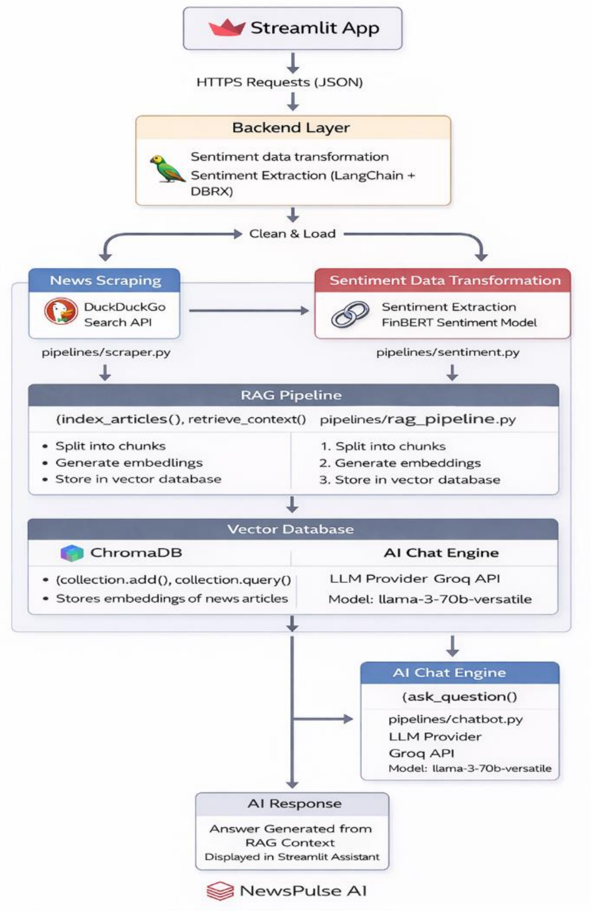
The above-advanced methods of sentiment analysis and retrieval-based NLP systems have not yet fully solved the problem as several issues still exist. For instance, several existing sentiment analysis platforms are dependent on static datasets and are unable to collect and analyze real-time web information dynamically. Moreover, some systems attempt solely at sentiment classification, which is neglecting contextual knowledge retrieval and can therefore compromise their accuracy in specialized domain applications. Besides, the combining of web data collection, semantic retrieval, and sentiment analysis in a single framework is still an area of research.

The paper aims to address a number of issues through a system that is called RAG-based Web Intelligence System for Domain-Specific Sentiment Analysis which combines automated web data collection, vector-based information retrieval, and large language model techniques. Firstly, it scrapes domain-specific web contents automatically secondly it processes the data and stores it in a vector database finally it uses a Retrieval-Augmented Generation pipeline for doing context-aware sentiment analysis. Using web intelligence combined with RAG-based contextual retrieval, this paper's solution is geared towards enhancing sentiment analysis accuracy, making it scalable and supporting reliability in domain-specific scenarios.[17]

3. METHODOLOGY

The RAG-based Web Intelligence System for Domain-Specific Sentiment Analysis is a proposal that aims to create an intelligent framework for gathering web data, processing textual information, and conducting context-aware sentiment analysis with the help of Retrieval-Augmented Generation (RAG). This system combines web data collection, natural language processing, vector databases, and large language models to derive insights from domain-specific content on the web. The goal of the system is to enhance the specificity and frame of reference understanding in sentiment analysis by joining the information retrieval techniques with the proficient language models. The approach includes the following key components:

(A) User Interface and Query Input, (B) Web Data Collection and News Scraping, (C) Data Cleaning and Sentiment Data Transformation, (D) RAG Pipeline and Context Retrieval, (E) Vector Database Storage, and (F) System Workflow.



The complete workflow of the proposed system is illustrated in **Fig. 1 – System Architecture**.

A.User Interface and Query Input

This system includes a web interface for users, which is simple and designed with the help of the Streamlit framework. It gives users the ability to interact with the system by just using a web browser. Through this UI, a user can write down a question or choose a domain topic for a sentiment analysis request. The system is designed in such a way that a user can simply express a desire for a sentiment insight without having any technical knowledge of the system behind.

Primary functionalities of this part are:

- The User Query Submission: Users are allowed to submit actually one or more domain topics or keywords, e. g. product names, technologies, or organizations to get an idea of the public sentiments towards that domain.
- Web-Based Interaction: The Streamlit interface sends and receives data to and from the backend service layer over secure HTTPS requests that are formatted in JSON.
- Result Visualization: The platform renders the sentiment analysis outcomes in a clear and structured manner so that users can grasp public attitudes towards the chosen domain.

B.Web Data Collection and News Scraping

The tools getting right information through sentiment analysis includes a self-operating internet data gathering module that gets hold of domain-specific text content from various online sources. The scraping process is supported by the extractions through APIs and the usage of web intelligence tools to fetch the news articles and other related web data.

- Web crawling automation: The system can access content from DuckDuckGo API which offers the searches that are corresponding with the user's questions.
- Content Extraction: Loading web pages for gathering textual contents such as titles, abstracts, and descriptions.
- Domain-specific filtering: The only contents that are related to the user's query will be selected for the subsequent processing to increase the precision of sentiment analysis.

C.Data Cleaning and Sentiment Data Transformation

The web data collected is mostly noisy or unstructured text that has to be processed for analysis. Hence, the system carries out several preprocessing operations to turn raw data into structured text ready for sentiment analysis. Transformation steps:

- Text Cleaning: Extraction of web-page content in the form of text only, no HTML tags, no special symbols, etc.
- Data Normalization: Usually, it means that text is converted to lowercase and stop words are removed.
- Chunk Segmentation: Textual documents are usually divided into smaller parts or chunks to enable efficient processing and retrieval in the RAG pipeline. Preprocessing of captured data is the first step of the analysis to get it into the right shape.

D. Retrieval-Augmented Generation (RAG) Pipeline

At the heart of the system lies the Retrieval-Augmented Generation (RAG) pipeline, which seamlessly fuses semantic retrieval with language model-based processing. The main idea behind the RAG pipeline is reflected in the below steps:

- Embedding production: Embedding models transform every piece of text into numerical vectors.
- Context retrieval: Upon receiving a query from a user, the system fetches the most semantically matching text segments from the vector database.
- Sentiment mining: The obtained contextual data is fed into a large language model that is interfaced via LangChain and Databricks DBRX which conducts the sentiment analysis and interpretation
- . This method allows the system to offer more precise and context-sensitive sentiment results than traditional sentiment classification methods.

E.Vector Database Storage

Storing the processed text embeddings in a vector database within the Databricks platform, the system can efficiently handle and retrieve a huge amount of textual data. The vector database module is responsible for these actions:

- Vector Index Creation: Generated text embeddings from the processed web data are kept in a vector index.
- Semantic Search: The database facilitates similarity-based retrieval with the assist of Databricks Vector Search Index, thus enabling the system to find text excerpts that are semantically similar to user queries.
- Efficient Data Retrieval: Adding this feature greatly enhances the speed and accuracy of finding information in the RAG pipeline.

F. System Workflow

The overall workflow of the proposed system operates as follows:

1. The user submits a domain-specific query through the Streamlit web interface.
2. The query is sent to the backend processing layer through HTTPS requests.
3. The web scraping module retrieves relevant content using the DuckDuckGo API.
4. The collected data undergoes cleaning and preprocessing.
5. Large textual documents are divided into smaller chunks.
6. The system generates embeddings for each chunk and stores them in the vector database.
7. The RAG pipeline retrieves relevant contextual information from the vector database.
8. The LangChain framework integrates the retrieved data with the DBRX language model to perform sentiment analysis.
9. The analyzed sentiment results are returned to the user through the Streamlit interface.

This integrated workflow allows the system to collect web data efficiently, get the related context information, and produce the accurate domain-specific sentiment insights. Traditional sentiment analysis systems are less effective, hence the new architecture boosting the contextual understanding, scalability, and accuracy of analysis for web intelligence applications.

4. RESULT AND DISCUSSION

The RAG-Based Web Intelligence System for Domain-Specific Sentiment Analysis, which was a proposal, was both developed and tested to find out how effective retrieval-augmented sentiment analysis on domain-specific web data is. The system is a combination of automated web scraping, vector-based information retrieval, and large language models that together produce contextual sentiment insights from online news content.

The platform's front end was built using the Streamlit web framework. The back end was composed out of LangChain for orchestrating the pipeline, DuckDuckGo API for fetching web data, and Databricks Vector Search was used for embedding storage and semantic retrieval. Experimental results show that the system is capable of automatically acquiring domain-specific news articles, conducting sentiment analysis and generating contextual insights via the RAG pipeline.

A. AI Market Assistant Interface

The first element of the system is an interactive AI Market Assistant interface, through which users can communicate with the system and make inquiries. Users can question the system about companies, market trends, or the latest news topics. The interface is realized with the Streamlit framework offering a simple, web-based way to access and utilize the system.

Users can send a request like the name of a company or the topic of the day. After a user submits a query, the system interprets it and, by exploiting information from the web, finds the most relevant news content. The RAG pipeline is used by the system that, after extracting the documents, it understands the information and outputs a response that is coherent with the context. The AI assistant not only explains the topic simply but also gives a glimpse into the mood of the news pieces it has consulted through sentiment analysis.

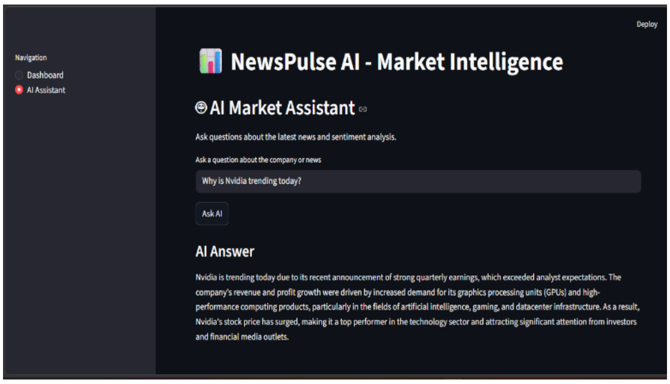


Fig. 2 – AI Market Assistant Interface

B. Market Intelligence Dashboard

Once you provide a company name or a topic, the tool will present a Market Intelligence Dashboard which consists of a summary of recent news articles along with the analysis of market sentiment. The dashboard has:

- Company Query Input: You can enter a company name or a topic like Tesla or Nvidia to get information.
- Latest News Articles: The dashboard updates you with the latest news articles about the selected company that the system finds.
- Sentiment Score Display: For each news article, the sentiment extraction model is used to analyze the text, and the sentiment score is shown along with the title of the article. This dashboard gives a quick snapshot of public sentiment and market developments concerning the selected company.

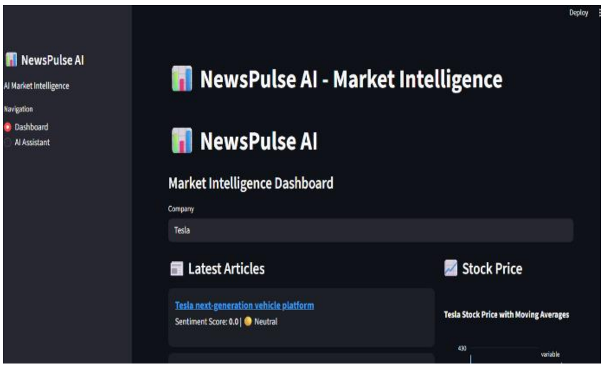


Fig. 3 – Market Intelligence Dashboard

C. Sentiment Timeline Analysis

The system further offers a Sentiment Timeline graph that visually represents the sentiment shifts in various news articles.The timeline chart features sentiment scores on the y-axis and article numbers on the x-axis. This graphic enables users to track changes in sentiment over time or between several news outlets.As an illustration, a steep plunge in sentiment ratings might reflect adverse news or market worries, whereas a rise in positive sentiments could signal good market conditions.Using the sentiment timeline, individuals can monitor rhythmic changes in sentiment and detect emerging attitudes in public opinion.

Fig. 4 – Sentiment Timeline Visualization

D. Sentiment Distribution Analysis

Besides offering a timeline visual, the system also comes with a Sentiment Distribution graph, which reflects the sentiment breakdown in all the news articles analyzed. The sentiment distribution graph divides sentiment scores into three ranges, i. e. positive neutral, and negative, and shows the number of times each has been occurred.This also enables users to spot the overall sentiment pattern of a company or a topic shortly.

For example, if most articles are found in the positive sentiment range, it means that the public perception is favorable. On the other hand, a majority of the articles with a negative sentiment score may suggest a negative publicity or an unfavorable market condition.

Such visualization undoubtedly makes it much easier to understand the results of sentiment analysis and helps users to decide on the right steps based on the changes in public opinion.

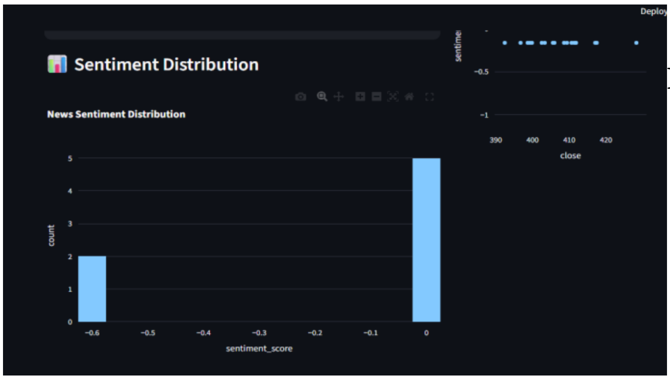


Fig. 5 – Sentiment Distribution Chart

E. RAG-Based Sentiment Analysis Process

The system carries out sentiment analysis by implementing a Retrieval-Augmented Generation (RAG) pipeline. Here is what it does:

- First, through the Streamlit interface, a user either inputs a company name or a domain-specific question for investigation.
- Then, the system looks up related news articles by employing DuckDuckGo search API.
- Next, the contents of the retrieved texts are cleaned and broken down into small chunks.
- Each chunk is transformed into vector embeddings and saved inside the vector database.
- The RAG pipeline identifies relevant contextual information from the vector database.
- LangChain facilitates coordinating the retrieved context and the DBRX language model to realize sentiment analysis.
- The system calculates sentiment scores and presents the results on the dashboard and in visualization graphs.
- This method enables the system to merge information
- retrieval and language model reasoning, and therefore it is capable of producing more accurate sentiment analysis than conventional methods.

F. News Sentiment Classification

The system is doing a sentiment analysis on every news article it grabs. Then it sorts the articles into categories: Positive, Negative, or Neutral. When a user enters a company name on the dashboard, the system goes to work collecting relevant news articles from online sources through the DuckDuckGo API. The contents of the retrieved articles are fed into the sentiment analysis module that operates through the Retrieval-Augmented Generation (RAG) pipeline.

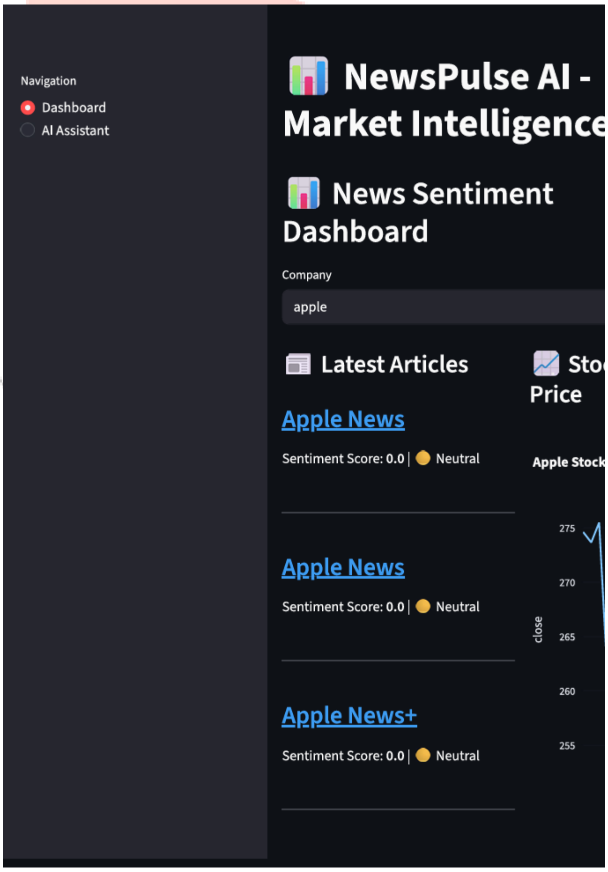


Fig. 6 – News Sentiment Classification in Market Intelligence Dashboard

At that time, the articles' textual contents are analyzed via the combination of the LangChain framework and DBRX language model, which understand the text's tone and context.

According to the semantic interpretation of the article, the system decides a sentiment score that indicates if the news is positive, negative, or neutral about the company.The sentiment class is shown in the dashboard after each article together with the sentiment score level. It makes it easy for users to see at a glance if the news about the company lately has been positive, negative, or neutral in terms of public perception.The sentiment classification for news articles aspect of the system can be a great help to users in getting the overall market sentiment, which in turn might be useful for decision-making by investors, analysts, and researchers.

5. CONCLUSION

This article outlined a RAG-Based Web Intelligence System for Domain-Specific Sentiment Analysis aimed at enhancing accurately and contextually understanding sentiment analysis of the web content. Familiar sentiment analysis systems depend mostly on the application of static datasets or classification methods based on keywords, which can be blind to the intricate contextual relationships of today's texts.

This system overcomes these drawbacks of the conventional methods through automated web data acquisition, vector-based semantic retrieval, and large language models being the components of Retrieval-Augmented Generation architecture. Besides gathering news with domain relevance through web scraping, the system converts the collected information into vectors, stores them in the vector database, and during sentiment analysis, it retrieves the corresponding contextual knowledge. The interface was developed with Streamlit, LangChain was used to orchestrate the whole workflow, DuckDuckGo API fetched the web intelligence, and Databricks Vector Search served embedding storing and retrieval. The raise in experimental outcomes indicates the system is capable of analyzing news content of a particular domain, producing sentiment understanding, and offering visual tools through sentiment timeline and distribution charts.

Integrating web intelligence techniques with retrieval-augmented language models is how the suggested system will boost the dependability, scalability, and contextual precision of sentiment analysis. The system has wide applications including market intelligence, brand reputation monitoring, financial trend analysis, and decision support systems,Future work can be directed at adding more data sources such as social media, enhancing sentiment classification models, and developing the system to facilitate real-time sentiment tracking of extensive web data.

6. REFERENCES

[1] B. Zhang, H. Yang, T. Zhou, A. Babar, and X. Liu, “Enhancing Financial Sentiment Analysis via Retrieval-Augmented Large Language Models,” *arXiv preprint arXiv:2310.04027*, 2023.

[2] Y. Gao et al., “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2312.10997*, 2023.

[3] S. Gupta, R. Ranjan, and S. N. Singh, “A Comprehensive Survey of Retrieval-Augmented Generation: Evolution, Current Landscape and Future Directions,” *arXiv preprint arXiv:2410.12837*, 2024.

[4] M. Arslan et al., “A Survey on Retrieval-Augmented Language Models in Natural Language Processing,” *Procedia Computer Science*, 2024.

[5] A. Brown, M. Roman, and B. Devereux, “A Systematic Literature Review of Retrieval-Augmented Generation: Techniques, Metrics,

and Challenges," 2025.

[6] M. Mathebula, "Fine-Tuning Retrieval-Augmented Generation with an Auto-Regressive Language Model for Sentiment Analysis," *Applied Sciences*, vol. 14, no. 23, 2024.

[7] S. Khaled et al., "Evaluating Large Language Models for Sentiment Analysis Using Retrieval-Augmented Generation," *Procedia Computer Science*, 2024.

[8] G. T. Roy et al., "RAG-Enhanced Aspect-Based Sentiment Analysis Using Large Language Models," 2024.

[9] E. Karakurt, "Retrieval-Augmented Generation and Large Language Models: A Review of Applications and Challenges," 2025.

[10] W. Fan et al., "A Survey on Retrieval-Augmented Generation Meeting Large Language Models," *ACM Computing Surveys*, 2024.

[11] M. Murtiyoso et al., "A Systematic Review of Retrieval-Augmented Generation in Large Language Models," *Sinkron Journal*, 2025.

[12] I. O. William, "Hybrid Search Relation Attribute Graph with LangChain and Pinecone Vector Database," in *Proc. IEEE Int. Conf. Engineering Technologies and Applied Sciences (ICETAS)*, 2024.

[13] S. Sawant et al., "A Retrieval-Augmented Generation Based Large Language Model for Domain-Specific Advisory Systems," 2026.

[14] D. Sun et al., "SCRAG: Social Computing-Based Retrieval Augmented Generation for Community Response Forecasting," *arXiv*, 2025.

[15] Y. Hu and Y. Lu, "RAG and RAU: A Survey on Retrieval-Augmented Language Models in Natural Language Processing," *arXiv*, 2024.

[16] R. Hou et al., "Retrieval-Augmented Multimodal Depression Detection Using Large Language Models," *arXiv*, 2025.

[17] G Kumaresan, K Devi, S Shanthi, B Muthusenthil… - Transactions on Emerging Telecommunications'"Hybrid Fuzzy Archimedes-based Light GBM-XGBoost model for distributed task scheduling in mobile edge computing", 2023

[18] M Mayuranathan, SK Saravanan, B Muthusenthil… - Advances in Engineering Software-"An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique", 2022

[19] L Karthikeyan, C Vijayakumaran, S Chitra… - Wireless Personal Communications" Saldeft: Self-adaptive learning differential evolution based optimal physical machine selection for fault tolerance problem in cloud", 2021

[20] K Revathi, A Samydurai, NS Kumar, K Keerthana - 2019 5th International Conference on" Advanced Health trackers in current market: A systematic review, trends and challenges" 2019